

9.4

Verwaltung von IBM MQ

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 599 gelesen werden.

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

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

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

Inhaltsverzeichnis

Verwalten.....	7
Methoden zur Verwaltung von IBM MQ-Warteschlangenmanagern und zugehörigen Ressourcen.....	8
Verwaltung von IBM MQ for Multiplatforms über Steuerbefehle.....	10
IBM MQ mit MQSC-Befehlen verwalten.....	12
MQSC-Befehlssyntax.....	13
Syntax der MQSC-Eingabedatei.....	16
MQSC-Befehle interaktiv unter runmqsc ausführen.....	18
MQSC-Befehle aus Textdateien unter runmqsc ausführen.....	23
Automatische Konfiguration aus einem MQSC-Script beim Start.....	25
IBM MQ-Verwaltung mit PCF-Befehlen automatisieren.....	26
Einführung in IBM MQ Programmable Command Formats.....	27
Verwendung von MQAI zur Vereinfachung der Verwendung von PCFs.....	39
Verwaltung über die REST API.....	76
Erste Schritte mit der administrative REST API.....	76
Fernverwaltung über die REST API.....	82
REST API-Zeitmarken.....	86
REST API-Fehlerbehandlung.....	86
REST API-Erkennung.....	89
Unterstützung landessprachunterstützter Sprachen in REST API.....	90
Versionen der REST API.....	92
Verwaltung über die IBM MQ Console.....	94
Erste Schritte mit der IBM MQ Console.....	94
Kurzübersicht über die IBM MQ Console.....	95
IBM MQ Console Einstellungen.....	124
Verwaltung über die IBM MQ Explorer.....	125
Was Sie mit dem IBM MQ Explorer tun können.....	125
IBM MQ Explorer einrichten.....	127
Anwendung " IBM MQ Taskbar" verwenden (nurWindows).....	133
Die IBM MQ -Alertmonitoranwendung (nur Windows).....	133
Mit lokalen IBM MQ-Objekten arbeiten.....	134
Mit Warteschlangenmanagern arbeiten.....	134
MQI-Kanäle stoppen.....	145
Mit lokalen Warteschlangen arbeiten.....	146
Mit fernen Warteschlangen arbeiten.....	156
Mit Aliaswarteschlangen arbeiten.....	159
Mit Modellwarteschlangen arbeiten.....	161
Mit dead-letter-Warteschlangen arbeiten.....	163
Mit Verwaltungsthemen arbeiten.....	184
Mit Subskriptionen arbeiten.....	188
Mit Services arbeiten.....	192
Objekte zum Auslösen verwalten.....	200
Dienstprogramm dmpmqmsg zwischen zwei Systemen verwenden.....	202
Mit fernen IBM MQ-Objekten arbeiten.....	206
WS-Manager für die Fernverwaltung konfigurieren.....	207
Den Befehlsserver für die Fernverwaltung verwalten.....	211
MQSC-Befehle auf einem fernen WS-Manager absetzen.....	212
Datenkonvertierung zwischen codierten Zeichensätzen.....	214
Managed File Transfer verwalten.....	218
MFT-Agenten starten.....	220
MFT-Agenten auflisten.....	225
MFT-Agenten stoppen.....	226
Neue Dateiübertragung starten.....	228

Geplante Dateiübertragung erstellen.....	231
Arbeiten mit anstehenden Dateiübertragungen.....	232
Dateiübertragung auslösen.....	233
Überwachen von Dateiübertragungen, die sich in Bearbeitung befinden.....	235
Status von Dateiübertragungen im Übertragungsprotokoll anzeigen.....	237
MFT-Ressourcen überwachen.....	239
Mit Dateiübertragungsvorlagen arbeiten.....	273
Daten von Dateien in Nachrichten übertragen.....	276
Daten von Nachrichten in Dateien übertragen.....	292
Die Protokollbridge.....	303
Connect:Direct-Bridge.....	327
Mit MFT über IBM Integration Bus arbeiten.....	344
Wiederherstellung und Neustart von MFT.....	344
Zeitlimit für die Wiederherstellung von blockierten Übertragungen festlegen.....	345
MQ Telemetry verwalten.....	351
Warteschlangenmanager unter Linux und AIX für Telemetrie konfigurieren.....	352
Warteschlangenmanager unter Windows für Telemetrie konfigurieren.....	354
Verteilte Steuerung von Warteschlangen für Senden von Nachrichten an MQTT-Clients konfigurieren.....	356
MQTT-Client - Identifikation, Berechtigung und Authentifizierung.....	359
Telemetriekanal-Authentifizierung mit TLS.....	365
Datenschutz auf Telemetriekanälen.....	369
TLS-Konfiguration von MQTT-Java-Clients und Telemetriekanälen.....	370
JAAS-Konfiguration für Telemetriekanal.....	375
AMQP-Client verwalten.....	377
AMQP-Service wird beim Start des Warteschlangenmanagers nicht automatisch gestartet.....	377
Von AMQP-Clients verwendete IBM MQ-Objekte anzeigen.....	378
AMQP-Clientidentifikation, -Autorisierung und -Authentifizierung.....	380
Datenschutz bei der Veröffentlichung von Kanälen.....	382
AMQP-Clients mit TLS konfigurieren.....	382
Verbindung zwischen AMQP-Clients und dem Warteschlangenmanager trennen.....	383
Multicast verwalten.....	384
Erste Schritte mit Multicasting.....	384
IBM MQ Multicast-Thementopologie.....	385
Größe von Multicastnachrichten steuern.....	386
Datenkonvertierung für Multicast-Messaging aktivieren.....	389
Multicast-Anwendungsüberwachung.....	389
Multicast-Nachrichtenzuverlässigkeit.....	390
Erweiterte Multicast-Tasks.....	391
IBM MQ for IBM i verwalten.....	394
IBM MQ for IBM i mithilfe von CL-Befehlen verwalten.....	394
Alternative Methoden zur Verwaltung von IBM MQ for IBM i.....	409
Ablaufsteuerung für IBM i.....	415
Verfügbarkeit, Sicherung, Wiederherstellung und Neustart unter IBM i.....	422
IBM MQ for IBM i in den Quiescemodus versetzen.....	468
Administering IBM MQ for z/OS.....	472
Issuing queue manager commands on z/OS.....	473
Using the operations and control panels on z/OS.....	487
Using the IBM MQ for z/OS utilities.....	495
Using the Command Facility on z/OS.....	498
Working with IBM MQ objects on z/OS.....	498
Implementing the system using multiple cluster transmission queues.....	501
Writing programs to administer IBM MQ for z/OS.....	503
Managing IBM MQ resources on z/OS.....	515
Recovery and restart on z/OS.....	553
IBM MQ and IMS.....	574
Operating Advanced Message Security on z/OS.....	586
IBM MQ Internet Pass-Thru verwalten.....	587

MQIPT starten und stoppen.....	587
MQIPT über die Befehlszeile verwalten.....	590
Backups erstellen.....	596
Leistungsoptimierung.....	597
Bemerkungen.....	599
Informationen zu Programmierschnittstellen.....	600
Marken.....	601

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.

Informationen zu diesem Vorgang

Sie können IBM MQ-Objekte lokal oder über Fernzugriff 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.

Weitere Informationen finden Sie unter [„Mit lokalen IBM MQ-Objekten arbeiten“](#) auf Seite 134.

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.




Weitere Informationen finden Sie unter [„Mit fernen IBM MQ-Objekten arbeiten“](#) auf Seite 206.


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 8
- [„IBM MQ-Scriptbefehle \(MQSC\)“](#) auf Seite 8
- [„Programmierbare Befehlsformate \(PCFs\)“](#) auf Seite 8
- [Die administrative REST API](#)
-  [„IBM i-Steuersprache“](#) auf Seite 10

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

-   [„Die IBM MQ Explorer“](#) auf Seite 9
- [„Die IBM MQ Console“](#) auf Seite 9
-  [„Microsoft Cluster Service \(MSCS\)“](#) auf Seite 10

 Informationen zu den Verwaltungsschnittstellen und -optionen unter IBM MQ for z/OS finden Sie unter [„Administering IBM MQ for z/OS“](#) auf Seite 472.

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 26.

Zugehörige Konzepte

[IBM MQ - Technische Übersicht](#)

Zugehörige Tasks

[Planung](#)

[konfigurieren](#)

Zugehörige Verweise

[Befehlsätze, Vergleich](#)

Methoden zur Verwaltung von IBM MQ-Warteschlangenmanagern und zugehörigen Ressourcen

Sie können IBM MQ -Warteschlangenmanager und zugehörige Ressourcen mithilfe von IBM MQ -Steuerbefehlen, IBM MQ Scriptbefehlen (MQSC), PCFs (Programmable Command Formats), administrative REST API, IBM MQ Console und IBM MQ Explorer verwalten. Für IBM i können Sie auch die IBM i -Steuersprache und für Windows auch den Microsoft Cluster Service (MSCS) verwenden.

IBM MQ-Steuerbefehle



Die Steuerbefehle bieten eine Möglichkeit, eine Reihe von IBM MQ -Verwaltungstasks auszuführen. Für AIX, Linux®, and Windows geben Sie diese Befehle in der Systembefehlszeile aus. Für IBM i geben Sie diese Befehle in einer Qshell aus. Siehe „[Verwaltung von IBM MQ for Multiplattformen über Steuerbefehle](#)“ auf Seite 10.

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.



Unter AIX, Linux, and Windows öffnen Sie eine **runmqsc** -Eingabeaufforderung und geben von dieser Eingabeaufforderung MQSC-Befehle an einen lokalen oder fernen Warteschlangenmanager aus. Sie können dies interaktiv tun oder eine Folge von Befehlen aus einer ASCII-Textdatei ausführen. Weitere Informationen hierzu finden Sie in den Abschnitten „[MQSC-Befehle interaktiv unter runmqsc ausführen](#)“ auf Seite 18 und „[MQSC-Befehle aus Textdateien unter runmqsc ausführen](#)“ auf Seite 23.



Unter IBM i erstellen Sie eine Liste mit Befehlen in einer Scriptdatei und führen die Datei anschließend mit dem Befehl **STRMQMMQSC** aus. Weitere Informationen finden Sie unter „[MQSC-Befehle unter IBM i verwalten](#)“ auf Seite 410.



Unter z/OS können MQSC-Befehle abhängig von dem Befehl aus einer Reihe von Quellen ausgegeben werden. Weitere Informationen finden Sie unter „[Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS](#)“ auf Seite 473.

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, Kanallis-

tener, 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 27.

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

Die administrative REST API

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:

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

For more information, see [The administrative REST API](#)

Die IBM MQ Console

Sie können die 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.


Die IBM MQ Explorer



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 for x86-64 -Systemen können Sie IBM MQ Explorer über das Systemmenü oder über die ausführbare Datei MQExplorer starten.

 Unter Linux müssen Sie zum erfolgreichen Starten von IBM MQ Explorer in der Lage sein, eine Datei in Ihr Ausgangsverzeichnis zu schreiben, und das Ausgangsverzeichnis muss vorhanden sein.

Weitere Informationen finden Sie unter „[Verwaltung über die IBM MQ Explorer](#)“ auf Seite 125.

Sie können IBM MQ Explorer verwenden, um ferne Warteschlangenmanager auf anderen Plattformen einschließlich z/OS zu verwalten.

Ab IBM MQ 9.3.0 wurde IBM MQ Explorer aus dem IBM MQ -Installationspaket entfernt. Das Produkt ist als separater Download weiterhin verfügbar und kann mithilfe der Downloaddatei für die eigenständige Version von IBM MQ Explorer, die von Fix Central heruntergeladen werden kann, installiert werden.

Weitere Informationen finden Sie unter [IBM MQ Explorer als eigenständige Anwendung unter Linux und Windows installieren und deinstallieren](#).

IBM i-Steuersprache

IBM i

Dies ist die bevorzugte Methode zum Absetzen von Verwaltungsbefehlen an IBM MQ for IBM i. 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 Programmbenutzer vorgesehen.

Ausführliche Informationen zur IBM i -Steuersprache (CL) finden Sie in den Abschnitten „[IBM MQ for IBM i mithilfe von CL-Befehlen verwalten](#)“ auf Seite 394 und [IBM MQ for IBM i -CL-Befehle](#).

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 ausgefalleneren 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 Tasks

„[IBM MQ mit MQSC-Befehlen verwalten](#)“ auf Seite 12

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. MQSC-Befehle sind auf allen Plattformen verfügbar.

Zugehörige Verweise

[Referenzinformationen zur Verwaltung](#)

Multi

Verwaltung von IBM MQ for Multiplatforms über Steuerbefehle

Die Steuerbefehle bieten eine Möglichkeit, eine Reihe von IBM MQ -Verwaltungstasks auszuführen. Für AIX, Linux und Windows geben Sie diese Befehle in der Systembefehlszeile aus. Für IBM i geben Sie diese Befehle in einer Qshell aus.

Vorbereitende Schritte

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.

Anmerkung: Verschiedene Plattformen können Befehlsargumente akzeptieren, die in einer anderen Reihenfolge eingegeben werden. Dies bedeutet insbesondere, dass Befehle, die unter Linux funktionieren, auf anderen Plattformen möglicherweise nicht funktionieren. Aus diesem Grund sollten Sie immer Argumente eingeben, die in den Syntaxdiagrammen angegeben sind.

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

Prozedur

- 

Verwenden Sie Steuerbefehle auf AIX and Linux -Systemen.

In IBM MQ for AIX or Linux-Systemen geben Sie Steuerbefehle in ein Shell-Fenster ein.

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 AIX, Linux, and Windows](#). Beachten Sie außerdem die umgebungsspezifischen Informationen. für die Plattform oder Plattformen, die Ihr Unternehmen verwendet.

In UNIX and Linux-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 `JUPI - TER . queue . manager` unterscheidet.

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

- 

Verwenden Sie Steuerbefehle auf Windows -Systemen.

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

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 AIX, Linux, and Windows](#). Beachten Sie außerdem die umgebungsspezifischen Informationen. für die Plattform oder Plattformen, die Ihr Unternehmen verwendet.

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

IBM i

Verwenden Sie Steuerbefehle auf IBM i -Systemen.

Unter IBM MQ for IBM i führen Sie die Steuerbefehle aus einer Qshell-Umgebung aus. Zur Verwendung der Qshell geben Sie `STRQSH` in die IBM i -Befehlszeile ein. Sie können jederzeit beenden und zur Befehlszeile zurückkehren, indem Sie die Taste F3 drücken.

Eine kleine Anzahl von Steuerbefehlen wird unter IBM i nicht unterstützt. Beispielsweise werden mehrere Installationsbefehle nicht unterstützt, weil Sie nicht mehr als eine Kopie von IBM MQ auf einem IBM i -System haben können. Befehle, die unter IBM i nicht unterstützt werden, sind **ALW** in der [IBM MQ -Steuerbefehlsreferenz](#) markiert.

Zugehörige Verweise

[Referenz zu IBM MQ-Steuerbefehlen](#)

IBM MQ mit MQSC-Befehlen verwalten

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. MQSC-Befehle sind auf allen Plattformen verfügbar.

Informationen zu diesem Vorgang

Die verfügbaren MQSC-Befehle werden in der [MQSC-Befehlsreferenz](#) beschrieben.

Die Art und Weise, wie Sie MQSC-Befehle absetzen, hängt von Ihrer Plattform ab:

- **ALW** Unter AIX, Linux, and Windows geben Sie MQSC-Befehle an einen Warteschlangenmanager über die `runmqsc` -Eingabeaufforderung aus. Sie können diese Eingabeaufforderung auf verschiedene Arten verwenden:
 - Interaktive Ausgabe von MQSC-Befehlen über eine Tastatur. Weitere Informationen finden Sie unter [„MQSC-Befehle interaktiv unter runmqsc ausführen“](#) auf Seite 18.
 - MQSC-Befehle aus einer ASCII-Textdatei ausgeben. Weitere Informationen finden Sie unter [„MQSC-Befehle aus Textdateien unter runmqsc ausführen“](#) auf Seite 23.
 - MQSC-Befehle auf einem fernen Warteschlangenmanager ausgeben. Weitere Informationen finden Sie unter [„MQSC-Befehle auf einem fernen WS-Manager absetzen“](#) auf Seite 212.
- **IBM i** Unter IBM i erstellen Sie eine Liste mit Befehlen in einer Scriptdatei und führen die Datei anschließend mit dem Befehl `STRMQMMQSC` aus. Weitere Informationen finden Sie unter [„MQSC-Befehle unter IBM i verwalten“](#) auf Seite 410.

- **z/OS** Unter z/OS können MQSC-Befehle abhängig von dem Befehl aus einer Reihe von Quellen ausgegeben werden. Weitere Informationen finden Sie unter [„Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS“](#) auf Seite 473.

Prozedur

- [„MQSC-Befehlssyntax“](#) auf Seite 13
- [„MQSC: Sonderzeichen und generische Werte“](#) auf Seite 15
- [„MQSC-Befehle interaktiv unter runmqsc ausführen“](#) auf Seite 18
- [„MQSC-Befehle aus Textdateien unter runmqsc ausführen“](#) auf Seite 23
- [„Automatische Konfiguration aus einem MQSC-Script beim Start“](#) auf Seite 25

Zugehörige Tasks

[Probleme mit MQSC-Befehlen beheben](#)

Zugehörige Verweise

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

MQSC-Befehlssyntax

Sie können MQSC-Befehle verwenden, um Warteschlangenmanagerobjekte zu verwalten. MQSC-Befehle sind auf allen Plattformen verfügbar. Einige Elemente der Befehlssyntax sind plattformspezifisch.

Reihenfolge der Parameter

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.

Leerzeichen und Kommas

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


Wiederholte Parameter sind nicht zulässig. Die Wiederholung eines Parameters mit seiner "NO"-Version, wie in REPLACE NOREPLACE, ist ebenfalls nicht zulässig.

Zeichenfolgen und einfache Anführungszeichen


Zeichenfolgen, die Leerzeichen, Kleinbuchstaben oder Sonderzeichen enthalten, müssen in einfache Anführungszeichen eingeschlossen werden, sofern keine der folgenden Bedingungen zutrifft:

- Die Sonderzeichen sind ein oder mehrere der folgenden Zeichen:
 - Punkt (.)
 - Schrägstrich (/)
 - Unterstrich (_)
 - Prozentzeichen (%)
-  Der Befehl wird aus den IBM MQ for z/OS-Betriebs- und -Steuerkonsolen ausgegeben.
- Die Zeichenfolge ist ein generischer Wert, der mit einem Stern endet. (unter IBM i müssen diese von Hochkommata umschlossen sein)
- Die Zeichenfolge ist ein einzelner Stern, z. B. TRACE(*) (unter IBM i müssen diese von Hochkommata umschlossen sein)
- Die Zeichenfolge ist 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.

 Unter Multiplatforms wird eine Zeichenfolge, die keine Zeichen enthält (d. h. zwei einfache Anführungszeichen ohne Leerzeichen dazwischen) als Leerzeichen in einfache Anführungszeichen gesetzt, d. h. in der gleichen Weise interpretiert wie ("). Die Ausnahme hiervon ist, wenn das Attribut, das verwendet wird, eines der folgenden Attribute ist, wenn zwei einfache Anführungszeichen ohne Leerzeichen als Zeichenfolge mit Nulllänge interpretiert werden:

- TOPICSTR
- SUB
- USERDATA
- SELECTOR

 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 ().

Alle abschließenden Leerzeichen in Zeichenfolgeattribut auf Basis von MQCHARV-Typen (z. B. SELECTOR und untergeordnete Benutzerdaten) werden berücksichtigt, d. h. 'abc ' entspricht nicht 'abc'.

Leere Klammern

Eine öffnende Klammer gefolgt von einer schließenden Klammer ohne wesentlichen Informationen dazwischen ist nicht zulässig, es sei denn, es wurde ausdrücklich festgelegt. Beispielsweise ist die folgende Zeichenfolge nicht gültig:

```
NAME ( )
```

Kleinschreibung und Großschreibung

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

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.

Sonderzeichen

MQSC-Befehle verwenden bestimmte Sonderzeichen, um bestimmte Bedeutungen zu haben. Weitere Informationen zu diesen Sonderzeichen und zur Verwendung dieser Sonderzeichen finden Sie unter „MQSC: Sonderzeichen und generische Werte“ auf Seite 15.

Zugehörige Tasks

[Probleme mit MQSC-Befehlen beheben](#)

Zugehörige Verweise

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

MQSC: Sonderzeichen und generische Werte

Einige Zeichen, z. B. Backslash (\) und Anführungszeichen (") Zeichen haben bei Verwendung mit MQSC-Befehlen eine besondere Bedeutung. Einige Sonderzeichen, die mit Parametern verwendet werden können, können generische Werte haben, müssen aber korrekt angegeben werden.

Vor Backslash (\) und Anführungszeichen (") Zeichen mit einem \, d. h., geben Sie \\ oder \" ein, wenn der Text \ oder " enthalten soll.


Ist für einen Parameter die Angabe eines generischen Wertes zulässig, wird dieser mit einem Stern (*) am Ende angegeben (z. B. ABC*). Die Angabe eines generischen Werts bedeutet, dass alle Werte mit diesem beginnen; bei der Angabe von ABC* beginnen also alle Werte mit ABC. Wenn der Wert Zeichen enthält, für die Anführungszeichen erforderlich sind, muss der Stern innerhalb der Anführungszeichen stehen, zum Beispiel 'abc*'. Der Stern muss das letzte oder das einzige Zeichen des Werts sein.

Das Fragezeichen (?) und der Doppelpunkt (:) sind in generischen Werten nicht zulässig.

Wenn diese Sonderzeichen in einem Feld (z. B. für eine Beschreibung) angegeben werden sollen, müssen Sie die ganze Zeichenfolge zwischen einfache Anführungszeichen setzen.

Zeichen	Beschreibung
	Leerzeichen werden als Trennzeichen verwendet. Mehrere Leerzeichen sind äquivalent zu einem einzelnen Leerzeichen, mit Ausnahme von Zeichenfolgen, die in Hochkommas (') eingeschlossen sind. Alle abschließenden Leerzeichen in diesen Zeichenfolgeattributen, die auf MQCHARV-Typen basieren, werden als signifikant behandelt.
,	Kommas werden als Trennzeichen verwendet. Mehrere Kommas entsprechen einem einzelnen Komma, außer in Zeichenfolgen, die in Hochkommas (') eingeschlossen sind.
'	Ein Hochkomma gibt den Anfang oder das Ende einer Zeichenfolge an. In IBM MQ werden alle Zeichen, die in Anführungszeichen eingeschlossen sind, exakt so angezeigt, wie sie eingegeben wurden. Bei der Berechnung der Länge einer Zeichenfolge werden die umschließenden Anführungszeichen nicht mitgezählt.
"	Einfache Anführungszeichen innerhalb einer Zeichenfolge werden von IBM MQ bei der Berechnung der Länge der Zeichenfolge als ein einziges Zeichen gezählt. Die Zeichenfolge ist an dieser Stelle nicht beendet.

Tabelle 1. Beschreibungen von Zeichen mit besonderer Bedeutung (Forts.)

Zeichen	Beschreibung
=	 Unter z/OS kennzeichnet ein Gleichheitszeichen den Beginn, ein Komma oder Leerzeichen das Ende eines Parameterwerts.
(Eine linke runde Klammer kennzeichnet den Beginn eines Parameterwertes oder einer Liste mit Werten.
)	Eine rechte runde Klammer kennzeichnet das Ende eines Parameterwertes oder einer Liste mit Werten.
:	Ein Doppelpunkt gibt einen Wertebereich an. So steht z. B. (1:5) für (1,2,3,4,5). Diese Schreibweise kann nur in TRACE -Befehlen verwendet werden.
*	Ein Stern steht für alle Zeichen. DISPLAY TRACE (*) bedeutet zum Beispiel, dass alle Traces angezeigt werden sollen, und DISPLAY QUEUE (PAY*) bedeutet, dass alle Warteschlangen angezeigt werden sollen, deren Namen mit PAY beginnen.




Syntax der MQSC-Eingabedatei

Wenn Sie lange Befehle haben oder eine bestimmte Befehlsfolge wiederholt verwenden, können Sie eine Eingabedatei verwenden, um MQSC-Befehle auszugeben. Der Inhalt der Eingabedatei muss der in diesem Thema beschriebenen Syntax entsprechen.

Übersicht

MQSC-Befehle werden über die *Standardeingabeeinheit* eingegeben, die auch als `stdin` bezeichnet wird. Normalerweise ist dies die Tastatur, aber Sie können angeben, dass die Eingabe aus einer Eingabedatei stammen soll.

Sie können diese Eingabedatei mit einem der folgenden plattformspezifischen Tools verwenden:

-  Bei Verwendung des Befehls **runmqsc** unter AIX, Linux, and Windows. Siehe [„MQSC-Befehle aus Textdateien unter runmqsc ausführen“](#) auf Seite 23
-  Bei Verwendung des Befehls **STRMQM** unter IBM i. Siehe [„MQSC-Befehle unter IBM i verwalten“](#) auf Seite 410
-  Bei Verwendung der Initialisierungsdatengruppen CSQINP1, CSQINP2 und CSQINPX oder des Stapeldienstprogramms CSQUTIL unter z/OS. Siehe [„Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS“](#) auf Seite 473

Syntax

Syntax der MQSC-Eingabedatei:

- Um Portierbarkeit zwischen IBM MQ-Umgebungen zu gewährleisten, müssen Sie die Zeilenlänge in MQSC-Befehlsdateien auf 72 Zeichen begrenzen.
- Jeder Befehl muss in einer neuen Zeile gestartet werden.
- Eine Zeile, die mit einem Stern (*) an erster Position startet, wird ignoriert. Die kann zum Einfügen von Kommentaren in die Datei verwendet werden.
- Leerzeilen werden ignoriert.
- Ein Pluszeichen (+) gibt an, daß der Befehl ab dem ersten nicht leeren Zeichen in der nächsten Zeile fortgesetzt wird. Wenn Sie + zum Fortsetzen eines Befehls verwenden, beachten Sie, dass Sie mindestens ein Leerzeichen vor dem nächsten Parameter setzen müssen (unter z/OS ist dies nicht

erforderlich). Alle Kommentare oder Leerzeilen werden gelöscht, wenn der Befehl in eine einzelne Zeichenfolge erneut assembliert wird.

- Ein Minuszeichen (-) gibt an, dass der Befehl am Anfang der nächsten Zeile fortgesetzt wird. Alle Kommentare oder Leerzeilen werden gelöscht, wenn der Befehl in eine einzelne Zeichenfolge erneut assembliert wird.
- MQSC-Befehle in einem Escape-PCF-Befehl (Programmable Command Format) können nicht mit dem Plus- oder Minuszeichen fortgesetzt werden. Der gesamte Befehl muss in einem einzigen Escape-Befehl enthalten sein. Informationen zu den PCF-Befehlen finden Sie im Abschnitt „Einführung in IBM MQ Programmable Command Formats“ auf Seite 27.
- Auf Multiplatforms und auf z/OS für Befehle, die vom Stapeldienstprogramm CSQUTIL ausgegeben werden, können Sie ein Semikolon (;) verwenden, um einen Befehl abzuschließen, auch wenn Sie am Ende der vorherigen Zeile ein Pluszeichen (+) eingegeben haben.
- Eine Zeile darf nicht mit Tastatursteuerzeichen (z. B. einem Tabulatorzeichen) enden.
- Wenn Sie den Befehl **runmqsc** im Clientmodus ausführen, indem Sie `stdin` aus einer Textdatei umleiten, und das Flag **-u** angeben, um Berechtigungsnachweise bereitzustellen, fordert der Befehl **runmqsc** nicht zur Eingabe eines Kennworts auf. Stattdessen wird das Kennwort aus `stdin` eingelesen. Sie sollten sicherstellen, dass die erste Datenzeile, die über `stdin` bereitgestellt wird, das Kennwort ist. Dazu können Sie Befehlszeilentools wie "echo" oder "cat" verwenden und das Kennwort gefolgt vom MQSC-Script an den **runmqsc** -Befehl `stdin` übergeben.
- **Windows** Wenn unter Windows bestimmte Sonderzeichen wie das Pfundzeichen (£) und das logische Nicht (¬) in einem Befehlsscript verwendet werden (z. B. als Teil einer Objektbeschreibung), werden sie in der Ausgabe von einem Befehl wie beispielsweise **DISPLAY QLOCAL** unterschiedlich angezeigt.

Siehe hierzu auch „MQSC-Befehlssyntax“ auf Seite 13.

Beispiele

Das folgende Beispiel ist ein Auszug aus einer MQSC-Befehlsdatei, die den Befehl **DEFINE QLOCAL** 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 1. Aus einer MQSC-Befehlsdatei extrahieren

Nach Abschluss des Befehls **runmqsc** wird ein Bericht zurückgegeben. Das folgende Beispiel zeigt einen Auszug aus einem Bericht:

```

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

Sie können auch die MQSC-Beispielbefehlsdateien verwenden, um Ihre Textdatei zu erstellen:

amqscos0.tst

Definitionen von Objekten, die von Beispielprogrammen verwendet werden.

amqscic0.tst

Definitionen von Warteschlangen für CICS-Transaktionen

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

Windows Unter 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.

ALW MQSC-Befehle interaktiv unter `runmqsc` ausführen

Unter AIX, Linux, and Windows können Sie über die `runmqsc` -Eingabeaufforderung MQSC-Befehle interaktiv an einen Warteschlangenmanager ausgeben. Interaktives Laufen eignet sich besonders für Schnelltests.

Vorbereitende Schritte

Sie müssen den Befehl `runmqsc` aus der Installation des Warteschlangenmanagers ausführen, mit dem Sie arbeiten. Sie können mit dem Befehl `dspmqr -o installation` herausfinden, welcher Installation ein Warteschlangenmanager zugeordnet ist.

Sie können leichter erkennen, dass Sie sich in einer MQSC-Umgebung befinden, und einige Details der aktuellen Umgebung anzeigen, indem Sie mit der Umgebungsvariablen `MQPROMPT` eine Eingabeaufforderung Ihrer Wahl festlegen. Weitere Informationen finden Sie unter [„Festlegen der MQSC-Eingabeaufforderung“](#) auf Seite 21.

Linux **AIX** Wenn Sie MQSC-Befehle interaktiv auf AIX and Linux -Plattformen ausführen, unterstützt die `runmqsc` -Eingabeaufforderung auch zusätzliche Befehlszeileneditorfunktionen. Siehe [„Befehlsrückruf und -beendigung und Emacs-Befehlstasten für runmqsc aktivieren“](#) auf Seite 22.

Informationen zu diesem Vorgang

Der Befehl **runmqsc** wird zum Öffnen einer Eingabeaufforderung verwendet, über die Sie MQSC-Befehle ausgeben können. Diese Befehle und ihre Syntax werden in der [MQSC-Befehlsreferenz](#) beschrieben.

Wenn Sie die **runmqsc** -Eingabeaufforderung wie in dieser Task beschrieben starten, legen Sie die Eingabeaufforderung so fest, dass sie in einem von drei Modi ausgeführt wird, je nachdem, welche Flags im Befehl festgelegt sind:

- *Verifizierungsmodus* - Die MQSC-Befehle werden in einem lokalen Warteschlangenmanager überprüft, aber nicht ausgeführt.
- *Direktmodus* - Die MQSC-Befehle werden in einem lokalen Warteschlangenmanager ausgeführt.
- *Indirekter Modus* - Die MQSC-Befehle werden in einem fernen Warteschlangenmanager ausgeführt.

Die folgende Prozedur legt die Ausführung der Eingabeaufforderung im Direktmodus fest. Weitere Optionen werden in den Beispielen dargestellt, die den Hauptschritten folgen.

Vorgehensweise

1. Öffnen Sie ein Befehlsfenster oder eine Shell und geben Sie den folgenden Befehl ein:

```
runmqsc QMgrName
```

Dabei gibt *QMgrName* den Namen des Warteschlangenmanagers an, den Sie die MQSC-Befehle verarbeiten wollen. Sie können *QMgrName* leer lassen, um MQSC-Befehle auf dem Standardwarteschlangenmanager zu verarbeiten.

2. Geben Sie bei Bedarf alle MQSC-Befehle ein. Geben Sie zum Beispiel den folgenden Befehl ein, um eine lokale Warteschlange mit dem Namen ORANGE . LOCAL . QUEUE zu erstellen:

```
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 ab dem ersten nicht leeren Zeichen in der folgenden Zeile fortgesetzt werden soll.

Die Befehlseingabe wird mit dem letzten Zeichen einer nicht leeren Zeile beendet, die kein Fortsetzungszeichen ist. Sie können die Befehlseingabe auch explizit beenden, indem Sie ein Semikolon (;) eingeben.

3. Stoppen Sie die Arbeit mit MQSC-Befehlen, indem Sie den folgenden Befehl eingeben:

```
end
```

Alternativ können Sie den Befehl **exit**, den Befehl **quit** oder das EOF-Zeichen für Ihr Betriebssystem verwenden.

Ergebnisse

Wenn Sie MQSC-Befehle absetzen, gibt der WS-Manager Bedienernachrichten zurück, die Ihre Aktionen bestätigen oder Ihnen mitteilen, welche Fehler Sie gemacht haben. Mit der folgenden Nachricht wird beispielsweise bestätigt, dass eine Warteschlange erstellt wird:

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

Die folgende Nachricht weist darauf hin, dass Sie einen Syntaxfehler gemacht haben:

```
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 Nachrichten werden an die Standardausgabeeinheit gesendet, die standardmäßig die Anzeige ist. Wenn Sie den Befehl nicht korrekt eingegeben haben, lesen Sie die Referenzinformationen zu dem Befehl, um die richtige Syntax zu finden. Siehe [MQSC-Befehlsreferenz](#).

Beispiel

Die folgenden Varianten des Befehls `runmqsc QMgrName` werden in den vorherigen Schritten verwendet. Diese Varianten erstellen unterschiedliche Konfigurationen der **runmqsc** -Eingabeaufforderung.

- Der folgende Befehl verwendet die Befehlsfilterung, um einen einzelnen MQSC-Befehl an den MQSC-Interpreter zu übergeben.

Unter Windows:

```
echo display chstatus(*) | runmqsc QMname
```

Unter Linux:

```
echo "display chstatus(*)" | runmqsc QMname
```

- Der folgende Befehl gibt keinen Warteschlangenmanagernamen an, sodass die MQSC-Befehle auf dem Standardwarteschlangenmanager verarbeitet werden.

```
runmqsc
```

- Mit dem folgenden Befehl werden Befehle mithilfe von QMLOCAL an den QMREMOTE-Warteschlangenmanager übergeben.

```
runmqsc -w 30 -m QMLOCAL QMREMOTE
```

- Dieser Befehl prüft, ob die Befehlssyntax auf einem lokalen WS-Manager korrekt ist, ohne die Befehle auszuführen. Beachten Sie, dass die zu überprüfenden Befehle aus einer Eingabedatei `myprog.in` eingelesen werden.

```
runmqsc -f myprog.in -v QmgrName
```

Weitere Informationen zum Arbeiten mit Ein- und Ausgabedateien finden Sie unter [„MQSC-Befehle aus Textdateien unter runmqsc ausführen“](#) auf Seite 23.

Nächste Schritte

Ausführliche Informationen zur **runmqsc** -Befehlssyntax, zu optionalen Parametern sowie zu Rückkehr-codes finden Sie im Abschnitt [runmqsc \(MQSC-Befehle ausführen\)](#).

Zugehörige Tasks

[„MQSC-Befehle aus Textdateien unter runmqsc ausführen“](#) auf Seite 23

Wenn Sie lange Befehle haben oder eine bestimmte Befehlsfolge wiederholt verwenden, können Sie eine Textdatei verwenden, um MQSC-Befehle auszugeben. Sie können `stdin` aus einer Textdatei umleiten. Sie können die Ausgabe auch in eine Datei umleiten.

Zugehörige Verweise

[Referenzinformationen zu MQSC-Befehlen](#)

ALW Festlegen der MQSC-Eingabeaufforderung


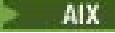




Verwenden Sie unter AIX, Linux, and Windows die Umgebungsvariable **MQPROMPT**, um die Eingabeaufforderung festzulegen, die beim Ausführen des Befehls **runmqsc** angezeigt wird. Dadurch können Sie leichter erkennen, dass Sie sich in einer MQSC-Umgebung befinden, und einige Details der aktuellen Umgebung anzeigen.

Informationen zu diesem Vorgang

Sie können die Eingabeaufforderung festlegen, die beim Ausführen des Befehls **runmqsc** angezeigt wird. Die Eingabeaufforderung wird eingefügt, wenn der Befehl **runmqsc** interaktiv ausgeführt wird und wenn die Eingabe von einer Datei oder von der Standardeingabeeinheit (`stdin`) in **runmqsc** umgeleitet wird.

Sie können einfachen Text in die Eingabeaufforderung einfügen und Sie können auch Umgebungsvariablen einfügen, indem Sie die Notation `+VARNAME+` auf dieselbe Weise wie IBM MQ-Serviceobjektdefinitionen verwenden. Weitere Informationen finden Sie unter „Austauschbare Einfügungen für Servicedefinitionen verwenden“ auf Seite 196.

Es gibt eine Reihe weiterer austauschbarer Einfügungen, die von IBM MQ bereitgestellt und in der folgenden Tabelle beschrieben werden.

Austauschbare Einfügung	Beschreibung
MQ_HOST_NAME	Hostname des Systems
MQ_FILE_SEP	Plattformspezifisches Dateitrennzeichen: <ul style="list-style-type: none">   Auf AIX and Linux -Systemen ist MQ_FILE_SEP /.  Auf Windows -Systemen lautet die Position von MQ_FILE_SEP \
MQ_PATH_SEP	Plattformspezifisches Pfadtrennzeichen: <ul style="list-style-type: none">   Auf AIX and Linux -Systemen ist MQ_PATH_SEP :.  Auf Windows -Systemen lautet die Position von MQ_PATH_SEP ;
MQ_DATE_TIME	Datum und Uhrzeit des lokalen Systems in einem festen YYYY-MM-DD hh:mm:ss.SSS-Format, z. B.: <div style="background-color: #f0f0f0; padding: 5px; margin-top: 5px;"> 2020-12-25 17:41:37.408 </div>

Anmerkungen:

- Die MQ-Werte der austauschbaren Einfügungen beziehen sich auf die IBM MQ-Installation und das Hostsystem, denen der Befehl **runmqsc** zugeordnet ist.
- MQPROMPT** ist auf maximal 256 Zeichen begrenzt, wenn Einfügungen erweitert werden. **MQPROMPT**-Erweiterungen über diesen Wert führen dazu, dass die gesamte Zeichenfolge **MQPROMPT** ohne die Erweiterungen abgeschnitten wird.

Beispiel

Im folgenden Beispiel wird die Eingabeaufforderung auf MQSCgesetzt:

• **Linux** **AIX**

```
export MQPROMPT="MQSC"
```

• **Windows**

```
set "MQPROMPT=MQSC"
```

AIX

Im folgenden Beispiel wird die Variable **MQPROMPT** auf einem AIX -System festgelegt. Die Eingabeaufforderung ist so eingestellt, dass ein Benutzername (aus der zugehörigen Systemumgebungsvariablen übernommen), der Name des Warteschlangenmanagers und der IBM MQ -Hostname (aus den austauschbaren IBM MQ -Einfügungen) angezeigt werden:

```
sh> export MQPROMPT="+USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
sh> runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT+=USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
C:\ > runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

Im folgenden Beispiel wird den obigen **MQPROMPT** -Beispielen eine Zeitmarke hinzugefügt, die aus den ersetzbaren MQ -Einfügungen stammt:

```
sh> export MQPROMPT="+MQ_DATE_TIME+ +USER+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
sh> runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
2020-11-24 18:10:00.404 myuser @ MY.QMGR @ aix1> DISPLAY QMSTATUS
```

```
C:\ > set "MQPROMPT+=MQ_DATE_TIME+ +USERNAME+ @ +QMNAME+ @ +MQ_HOST_NAME+> "  
C:\ > runmqsc MY.QMGR  
5724-H72 (C) Copyright IBM Corp. 1994, 2024.  
Starting MQSC for queue manager MY.QMGR.  
  
2020-11-24 18:10:01.007 myuser @ MY.QMGR @ WIN1> DISPLAY QMSTATUS
```

Linux **AIX** Befehlsrückruf und -beendigung und Emacs-Befehlstasten für runmqsc aktivieren



Verwenden Sie die **runmqsc** -Eingabeaufforderung unter AIX and Linux , um Befehlsrückruf, Befehlsabschluss und Emacs-Befehlstasten zu aktivieren.

Informationen zu diesem Vorgang

Auf AIX and Linux -Systemen können Sie die folgenden zusätzlichen Befehlszeileneditorfunktionen über die **runmqsc** -Eingabeaufforderung verfügbar machen:

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

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

-  Installieren Sie unter AIX curses.
-  Installieren Sie unter Linux ncurses.

Prozedur

- Installieren Sie ncurses oder curses.

Anmerkung: Im folgenden Beispiel werden Anweisungen für Linux verwendet.

Führen Sie den folgenden Befehl aus, um die vorhandenen ncurses -Pakete zu suchen:

```
rpm -qa | grep -i ncurses
```

Folgende ncurses -Pakete sind erforderlich:

```
ncurses-term-6.1-7.20180224.el8.noarch
ncurses-6.1-7.20180224.el8.x86_64
ncurses-base-6.1-7.20180224.el8.noarch
ncurses-c++-libs-6.1-7.20180224.el8.x86_64
ncurses-libs-6.1-7.20180224.el8.x86_64
ncurses-compat-libs-6.1-7.20180224.el8.x86_64
ncurses-devel-6.1-7.20180224.el8.x86_64
```

Sie können alle erforderlichen ncurses -Pakete installieren, die im vorherigen Text aufgelistet sind, indem Sie den folgenden Befehl ausführen:

```
yum install ncurses*
```

- Passen Sie die Emacs-Tastenbelegungen an.

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 durch Bearbeiten der Datei `.editrc` angepasst, die im Ausgangsverzeichnis gespeichert ist. Weitere Informationen finden Sie unter [Editrc](#) auf den Seiten des FreeBSD-Mannes.

- Inaktivieren Sie den Befehlsrückruf, den Befehlsabschluss und die Emacs-Befehlstasten.

Setzen Sie dazu die Umgebungsvariable **MQ_OVERRIDE_LIBEDIT_LOAD** auf TRUE.

Diese Umgebungsvariable kann als Problemumgehung verwendet werden, wenn die **runmqsc** -Eingabeaufforderung die folgende Informationsnachricht anzeigt:

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

MQSC-Befehle aus Textdateien unter runmqsc ausführen

Wenn Sie lange Befehle haben oder eine bestimmte Befehlsfolge wiederholt verwenden, können Sie eine Textdatei verwenden, um MQSC-Befehle auszugeben. Sie können `stdin` aus einer Textdatei umleiten. Sie können die Ausgabe auch in eine Datei umleiten.

Vorbereitende Schritte

Bei dieser Task wird davon ausgegangen, dass Sie eine Textdatei erstellt haben, die die MQSC-Befehle enthält, die Sie ausführen wollen. Detaillierte Informationen zur Syntax und Beispiele für diese Dateien finden Sie in [„Syntax der MQSC-Eingabedatei“](#) auf Seite 16.

Sie können die MQSC-Eingabeaufforderung mit der Umgebungsvariablen **MQPROMPT** auf eine Eingabeaufforderung Ihrer Wahl setzen. Weitere Informationen finden Sie unter [„Festlegen der MQSC-Eingabeaufforderung“](#) auf Seite 21.

Informationen zu diesem Vorgang

Die Eingabe für den Befehl **runmqsc** wird aus der *Standardeingabeeinheit* übernommen, die auch als `stdin` bezeichnet wird. Normalerweise ist dies die Tastatur, aber Sie können angeben, dass die Eingabe von einem seriellen Anschluss oder einer Datei kommen soll.

Die Ausgabe für den Befehl **runmqsc** wird an die *Standardausgabereinheit* ausgegeben, die auch als `stdout` bezeichnet wird. Normalerweise ist dies eine Anzeige, aber Sie können die Ausgabe an einen seriellen Anschluss oder an eine Datei umleiten.

Vorgehensweise

1. Prüfen Sie auf einem lokalen Warteschlangenmanager, ob die Befehlssyntax in der Datei korrekt ist, ohne die Befehle auszuführen.

Verwenden Sie das Flag **-v** im Befehl **runmqsc** zusammen mit einer der folgenden Optionen:

- Verwenden Sie die Option **-f**, um den Dateinamen des Eingabetexts anzugeben. For example:

```
runmqsc -f myprog.in -v localQmgrName
```

Bei der Prüfung von Befehlen kann kein ferner Warteschlangenmanager angegeben werden. Das heißt, Sie können das Flag **-w** nicht angeben.

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

2. Wenn die Befehlssyntax richtig ist, entfernen Sie das Flag **-v** und führen den Befehl **runmqsc** anschließend erneut aus.

Sie können jetzt einen fernen WS-Manager festlegen.

- Führen Sie beispielsweise den folgenden Befehl aus:

```
runmqsc -f myprog.in QmgrName
```

[Abbildung 1](#) auf Seite 17 zeigt einen Auszug aus einer Befehlsdatei wie `myprog.in` und [Abbildung 2](#) auf Seite 18 den entsprechenden Auszug der Ausgabe aus einer Berichtsdatei wie `results.out`.

Nächste Schritte

Ausführliche Informationen zur **runmqsc** -Befehlssyntax, zu optionalen Parametern sowie zu Rückkehrcodes finden Sie im Abschnitt [runmqsc \(MQSC-Befehle ausführen\)](#).

Zugehörige Tasks

[„Festlegen der MQSC-Eingabeaufforderung“](#) auf Seite 21

Verwenden Sie unter AIX, Linux, and Windows die Umgebungsvariable **MQPROMPT**, um die Eingabeaufforderung festzulegen, die beim Ausführen des Befehls **runmqsc** angezeigt wird. Dadurch können Sie leichter erkennen, dass Sie sich in einer MQSC-Umgebung befinden, und einige Details der aktuellen Umgebung anzeigen.

[„MQSC-Befehle interaktiv unter runmqsc ausführen“](#) auf Seite 18

Unter AIX, Linux, and Windows können Sie über die **runmqsc** -Eingabeaufforderung MQSC-Befehle interaktiv an einen Warteschlangenmanager ausgeben. Interaktives Laufen eignet sich besonders für Schnelltests.

Zugehörige Verweise

[Referenzinformationen zu MQSC-Befehlen](#)

Automatische Konfiguration aus einem MQSC-Script beim Start

Sie können Ihren Warteschlangenmanager so konfigurieren, dass er den Inhalt eines MQSC-Scripts oder einer Gruppe von MQSC-Scripts bei jedem Start des Warteschlangenmanagers automatisch anwendet.

Durch die Verwendung dieser Funktion kann eine Konfiguration geändert und beim nächsten Warteschlangenmanagerneustart automatisch wiederholt werden. Wenn sich das Script oder die Scripts beispielsweise in einem angehängten Laufwerk befinden, kann eine zentrale Konfiguration verwendet werden, bei der die neueste Version beim Start für jeden Warteschlangenmanager angewendet wird.

Wenn Sie beispielsweise sicherstellen möchten, dass alle Warteschlangenmanagern in einem Uniform-Cluster die gleichen Definitionen enthalten, können Sie eine einzelne Konfigurationsgruppe verwenden, in der diese alle angewendet werden. Ein Beispiel hierfür finden Sie im Abschnitt [Neuen Uniform-Cluster erstellen](#).

Vorbereitungen

Sie können Folgendes verwenden:

1. Ein einzelnes Script, um eine Textdatei mithilfe von MQSC-Befehlen zu erstellen.
2. Eine Gruppe von MQSC-Scripts,
 - um ein Verzeichnis anzugeben, in dem die Konfigurationen vorhanden sein sollen
 - Erstellen Sie in diesem Verzeichnis Dateien mit der Erweiterung `.mqsc`, z. B. `queues.mqsc`.

Da dieses Script bei jedem Start eines Warteschlangenmanagers erneut angewendet wird, ist es wichtig, dass die Befehle wiederholt werden können. Ein **DEFINE**-Befehl muss beispielsweise die Zeichenfolge **REPLACE** enthalten, da der Befehl andernfalls beim zweiten Warteschlangenmanagerstart als Fehler angezeigt wird, da das Objekt bereits vorhanden ist.

Beachten Sie, dass jede Zeile in einem MQSC-Script mit dem Präfix `*` als Kommentar verarbeitet wird.

Automatische Konfiguration von MQSC-Scripts aktivieren

Wichtig: Sie dürfen keine Befehle für Kanäle vom Typ MQTTausgeben, da diese nicht für die automatische Konfiguration während des Starts unterstützt werden.

Sie können einen neuen Warteschlangenmanager mit dem Flag **-ic** für den Befehl **crtmqm** konfigurieren und auf eine bestimmte Datei oder ein bestimmtes Verzeichnis verweisen. Der angegebener Wert wird in der Datei `qm.ini` in der Zeilengruppe 'AutoConfig' als Attribut **MQSCConfig** gespeichert.

Sie können einen vorhandenen Warteschlangenmanager für die Aktivierung der automatischen MQSC-Konfiguration einrichten, indem Sie das AutoConfig-Zeilengruppenattribut **MQSCConfig** hinzufügen, das auf eine gültige Datei oder ein gültiges Verzeichnis verweist. For example:

```
AutoConfig:
MQSCConfig=C:\mq_configuration\uniclus.mqsc
```

Wie funktioniert die automatische Konfiguration?

Während des Starts des Warteschlangenmanagers wird die durch das **AutoConfig** Zeilengruppenattribut **MQSCConfig** angegebene Konfiguration durch die **runmqsc** -Validierung übergeben, um eine gültige Syntax sicherzustellen, und anschließend in der Datenbaumstruktur des Warteschlangenmanagers im Verzeichnis `autocfg` als einzelne Datei `cached.mqsc` gespeichert.

Wenn mehrere Dateien aus einem Verzeichnis verarbeitet werden, geschieht dies in alphabetische Reihenfolge, und wenn ein MQSC-Befehl zum Beenden oder Abbrechen des Vorgangs enthalten ist, werden die restlichen Inhalte dieser Datei übersprungen.

Wenn beim ersten Start des Warteschlangenmanagers die Datei oder das Verzeichnis nicht gelesen werden können oder eine Datei mit einer ungültigen MQSC-Syntax vorhanden ist, wird verhindert, dass der

Warteschlangenmanager starten kann, und es wird eine entsprechende Fehlernachricht in die Konsole und in das Fehlerprotokoll des Warteschlangenmanagers geschrieben.

Wenn die Datei oder das Verzeichnis, auf die bzw. auf das verwiesen wird, bei nachfolgenden Neustarts nicht lesbar ist oder eine ungültige MQSC-Syntax enthält, wird die zuvor zwischengespeicherte Datei verwendet und es wird eine Nachricht in das Fehlerprotokoll des Warteschlangenmanagers geschrieben, in der dies angezeigt wird.

An der Stelle, an der der Inhalt der `cached.mqsc` auf den Warteschlangenmanager angewendet wird, wenn alle MQSC-Befehle angewendet wurden, ist der Warteschlangenmanager für Anwendungen aktiviert, um eine Verbindung herzustellen. Das Protokoll `runmqsc` der Konfiguration, die angewendet wird, wird im Fehlerverzeichnis des Warteschlangenmanagers in einer Datei mit der Bezeichnung `autocfgmqsc.LOG` gespeichert.

Außerdem werden alle MQSC-Befehle, die nicht erfolgreich abgeschlossen werden, im Fehlerprotokoll des Warteschlangenmanagers protokolliert und zeigen an, warum der Befehl fehlgeschlagen ist.

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 (*MsqType*) 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 27.

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 auf allen IBM MQ -Plattformen unterstützt.

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 28](#)
- [„PCF-Antworten in IBM MQ“ auf Seite 31](#)
-  [„Extended responses“ auf Seite 33](#)
- [Regeln für die Benennung von IBM MQ -Objekten](#)
- [„Berechtigungsprüfung für PCF-Befehle in IBM MQ“ auf Seite 34](#)


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.

 Unter [Multiplatform](#) wird die Warteschlange, an die die PCF-Befehle gesendet werden, stets als `SYSTEM.ADMIN.COMMAND.QUEUE`.

 Unter [z/OS](#) werden Befehle an `SYSTEM.COMMAND.INPUT`, obwohl `SYSTEM.ADMIN.COMMAND.QUEUE` kann ein Aliasname für sie sein. 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 ein IBM MQ for Multiplatforms-System senden, setzen Sie dieses Feld auf die Codierung, die für die Nachrichtendaten verwendet wird. Bei Bedarf wird eine Konvertierung durchgeführt.

CodedCharSetId

Wenn Sie an ein IBM MQ for Multiplatforms-System senden, setzen Sie dieses Feld auf die codierte Zeichensatz-ID, die für die Nachrichtendaten verwendet wird. Bei Bedarf wird eine Konvertierung durchgeführt.

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-Nachrichtenoption 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 Nachrichtendeskriptorfeld *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 nur auf Multiplatforms unterstützt. Ältere Anwendungen unterstützen PCF unter z/OS nicht. IBM MQ Windows Explorer ist eine solche Anwendung (PCF wird jedoch von IBM WebSphere MQ 6.0 oder höher IBM MQ Explorer unter z/OS unterstützt).

Wenn der PCF-Header-Typ der Befehlsnachricht MQCFT_COMMAND_XR lautet, werden entweder erweiterte oder Standardantworten generiert. Solche Befehle werden unter z/OS und einigen Multiplatforms unterstützt. Unter z/OS abgesetzte Befehle generieren nur erweiterte Antworten.

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 Befehlsformate](#)) 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.

Multi Standardantworten

Befehlsnachrichten mit dem Headertyp MQCFT_COMMAND, Standardantworten werden generiert. Solche Befehle werden nur auf Multiplatforms 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 Fehlermeldungen gesendet (mehr als eine Nachricht kann auch für einen Befehl gesendet werden, der normalerweise nur eine einzige Antwortnachricht hätte). Diese Fehlerantwortnachrichten 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.

Extended responses

Commands issued on z/OS generate extended responses.

There are three types of extended response:

- Message response, with type MQCFT_XR_MSG
- Item response, with type MQCFT_XR_ITEM
- Summary response, with type MQCFT_XR_SUMMARY

Each command can generate one, or more, sets of responses. Each set of responses comprises one or more messages, numbered sequentially from 1 in the *MsgSeqNumber* field of the PCF header. The *Control* field of the last (or only) response in each set has the value MQCFC_LAST. For all other responses in the set, this value is MQCFC_NOT_LAST.

Any response can include one, or more, optional MQCFBS structures in which the *Parameter* field is set to MQBACF_RESPONSE_SET, the value being a response set identifier. Identifiers are unique and identify the set of responses which contain the response. For every set of responses, there is an MQCFBS structure that identifies it.

Extended responses have at least two parameter structures:

- An MQCFBS structure with the *Parameter* field set to MQBACF_RESPONSE_ID. The value in this field is the identifier of the set of responses to which the response belongs. The identifier in the first set is arbitrary. In subsequent sets, the identifier is one previously notified in an MQBACF_RESPONSE_SET structure.
- An MQCFST structure with the *Parameter* field set to MQCACF_RESPONSE_Q_MGR_NAME, the value being the name of the queue manager from which the set of responses come.

Many responses have additional parameter structures, and these structures are described in the following sections.

You cannot determine in advance how many responses there are in a set other than by getting responses until one with MQCFC_LAST is found. Neither can you determine in advance how many sets of responses there are as any set might include MQBACF_RESPONSE_SET structures to indicate that additional sets are generated.

Extended responses to Inquire commands

Inquire commands normally generate an item response (type MQCFT_XR_ITEM) for each item found that matches the specified search criteria. The item response has a *CompCode* field in the header with a value of MQCC_OK, and a *Reason* field with a value of MQRC_NONE. It also includes other parameter structures describing the item and its requested attributes, as described in [Definitions of the Programmable Command Formats](#).

If an item is in error, the *CompCode* field in the header has a value of MQCC_FAILED and the *Reason* field identifies the particular error. Additional parameter structures are included to identify the item.

Certain Inquire commands might return general (not name-specific) message responses in addition to the item responses. These responses are informational, or error, responses of the type MQCFT_XR_MSG.

If the Inquire command succeeds, there might, optionally, be a summary response (type MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_OK, and a *Reason* field value of MQRC_NONE.

If the Inquire command fails, item responses might be returned, and there might optionally be a summary response (type MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_FAILED, and a *Reason* field value of MQRCCF_COMMAND_FAILED.

Extended responses to commands other than Inquire

Successful commands generate message responses in which the *CompCode* field in the header has a value of MQCC_OK, and the *Reason* field has a value of MQRC_NONE. There is always at least one message; it might be informational (MQCFT_XR_MSG) or a summary (MQCFT_XR_SUMMARY). There might optionally be additional informational (type MQCFT_XR_MSG) messages. Each informational message might include a number of additional parameter structures with information about the command; see the individual command descriptions for the structures that can occur.

Commands that fail generate error message responses (type MQCFT_XR_MSG), in which the *CompCode* field in the header has a value of MQCC_FAILED and the *Reason* field identifies the particular error. Each message might include a number of additional parameter structures with information about the error: see the individual error descriptions for the structures that can occur. Informational message responses might be generated. There might, optionally, be a summary response (MQCFT_XR_SUMMARY), with a *CompCode* value of MQCC_FAILED, and a *Reason* field value of MQRCCF_COMMAND_FAILED.

Extended responses to commands using CommandScope

If a command uses the **CommandScope** parameter, or causes a command using the **CommandScope** parameter to be generated, there is an initial response set from the queue manager where the command was received. Then a separate set, or sets, of responses is generated for each queue manager to which the command is directed (as if multiple individual commands were issued). Finally, there is a response set from the receiving queue manager which includes an overall summary response (type MQCFT_XR_SUMMARY). The MQCACF_RESPONSE_Q_MGR_NAME parameter structure identifies the queue manager that generates each set.


The initial response set has the following additional parameter structures:

- MQIACF_COMMAND_INFO (MQCFIN). Possible values in this structure are MQCMDI_CMDSCOPE_ACCEPTED or MQCMDI_CMDSCOPE_GENERATED.
- MQIACF_CMDSCOPE_Q_MGR_COUNT (MQCFIN). This structure indicates the number of queue managers to which the command is sent.

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 2 auf Seite 36](#) 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 2 auf Seite 36](#) dargestellt.

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

Anmerkung: Nur für Windows kann die Benutzer-ID zur Gruppe *Administratoren* oder zur Gruppe *mqmge* gehören.

- 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

Multi

<i>Tabelle 2. Objektberechtigungen</i>		
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
Warteschlange ändern	dsp und chg	nicht zutreffend
Warteschlangenmanager ändern	chg <i>siehe Anmerkung 3 und Anmerkung 5</i>	nicht zutreffend
Service ändern	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

Tabelle 2. Objektberechtigungen (Forts.)

Befehl	IBM MQ-Objektberechtigung	Klassenberechtigung (für Objekttyp)
Authentifizierungsinformationen erstellen (Ersetzen) <i>siehe Anmerkung 1</i>	(Systemstandardauthentifizierungsinformationen) dsp to: chg	crt
Kanal erstellen	(Systemstandardkanal) dsp	crt
Kanal erstellen (Ersetzen) <i>siehe Anmerkung 1</i>	(Systemstandardkanal) dsp to: chg	crt
Kanallistener erstellen	(Systemstandardlistener) dsp	crt
Create Channel Listener (Ersetzen) <i>siehe Anmerkung 1</i>	(system default listener) dsp to: chg	crt
Clientverbindungskanal erstellen	(Systemstandardkanal) dsp	crt
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

Tabelle 2. Objektberechtigungen (Forts.)

Befehl	IBM MQ-Objektberechtigung	Klassenberechtigung (für Objekttyp)
Berechtigungsdatensätze anfragen	siehe Anmerkung 4	siehe Anmerkung 4
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
Warteschlangenmanager abfragen	siehe Anmerkung 3	nicht zutreffend
Warteschlangenstatus abgefragt	DSP	nicht zutreffend
Inquire Service	DSP	nicht zutreffend
Pingkanal	Strg	nicht zutreffend
Ping-WS-Manager	siehe Anmerkung 3	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	(Warteschlangenmanagerobjekt) chg siehe Anmerkung 4	siehe Anmerkung 4
Kanal starten	Strg	nicht zutreffend
Kanal stoppen	Strg	nicht zutreffend
Verbindung stoppen	(Warteschlangenmanagerobjekt) chg	nicht zutreffend
Listener starten	Strg	nicht zutreffend
Listener stoppen	Strg	nicht zutreffend
Service starten	Strg	nicht zutreffend
Service stoppen	Strg	nicht zutreffend
Escape	siehe Anmerkung 2	siehe Anmerkung 2

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, IBM i, Linux, 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

In den folgenden Beispielprogrammen wird die Verwendung der MQAI für die Ausführung der verschiedenen Tasks gezeigt:

- `amqsaicq.c`: Erstellen Sie eine lokale Warteschlange.
- `amqsailem.c`: Anzeigen von Ereignissen auf dem Bildschirm mit Hilfe eines einfachen Ereignismonitors.
- `amqsailq.c`: Drucken Sie eine Liste mit allen lokalen Warteschlangen und ihren aktuellen Tiefen.
- `amqsaicl.c`: Drucken Sie eine Liste mit allen Kanälen und deren Typen..

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 40.

Zugehörige Verweise

[Referenzinformationen zur IBM MQ Administration Interface](#)

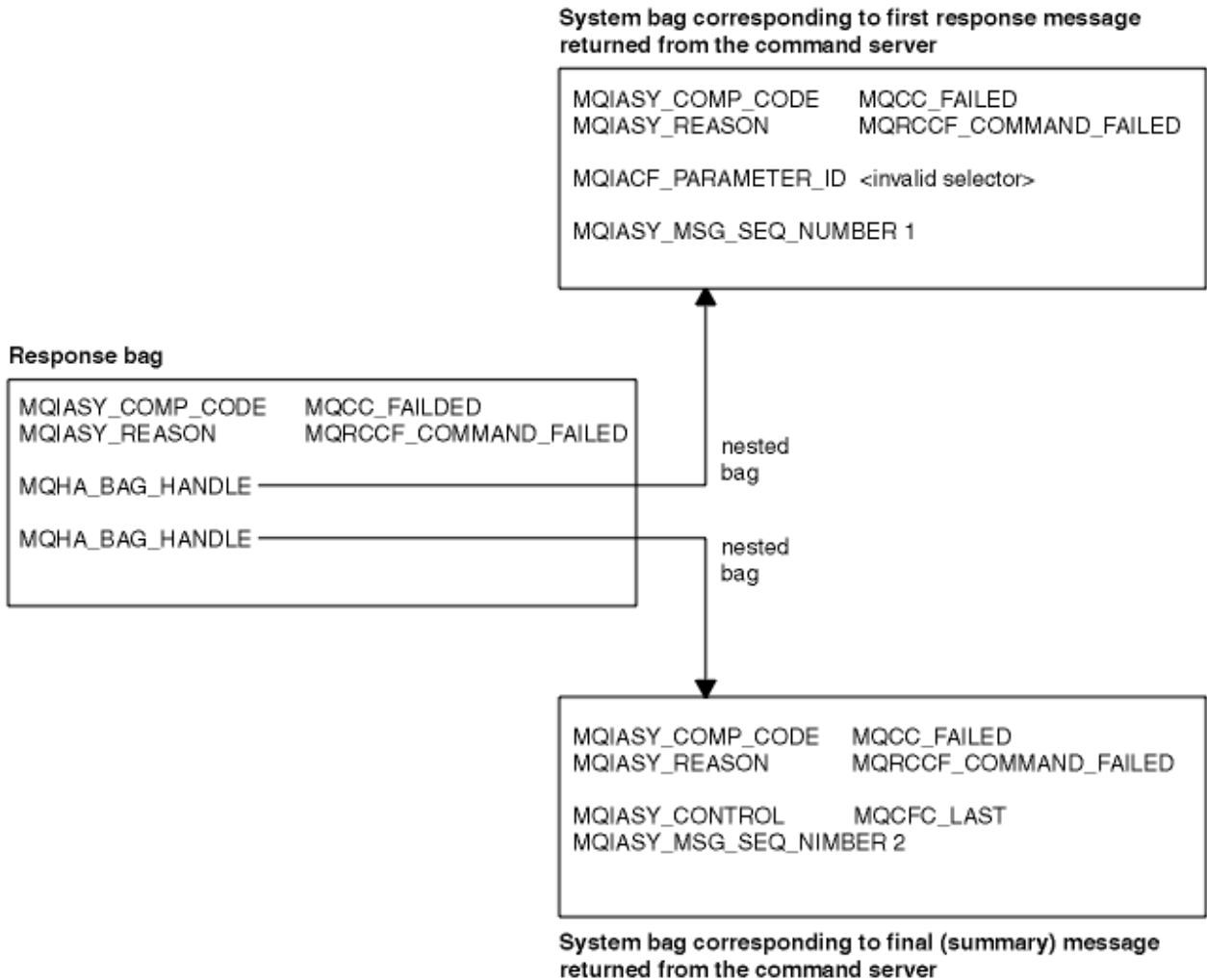
Hinweise und Tipps zur Verwendung von MQAI für die Konfiguration von IBM MQ

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.
- Um den Wert eines Zeichenfolgeattributs zu löschen, setzen Sie es 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 Warteschlangename 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 unter [„Verwaltungsbefehle mit dem mqExecute-Aufruf an den Befehl 'qm' senden“](#) auf Seite 75

Das folgende Diagramm zeigt dieses Szenario:



Multi **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.

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.

Verwendung des Nachrichtendeskriptors

Die MQAI generiert einen Nachrichtendeskriptor, der bei der Erstellung des Datensacks auf einen Anfangswert gesetzt wird.

Multi **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 3 auf Seite 42](#) dargestellt.

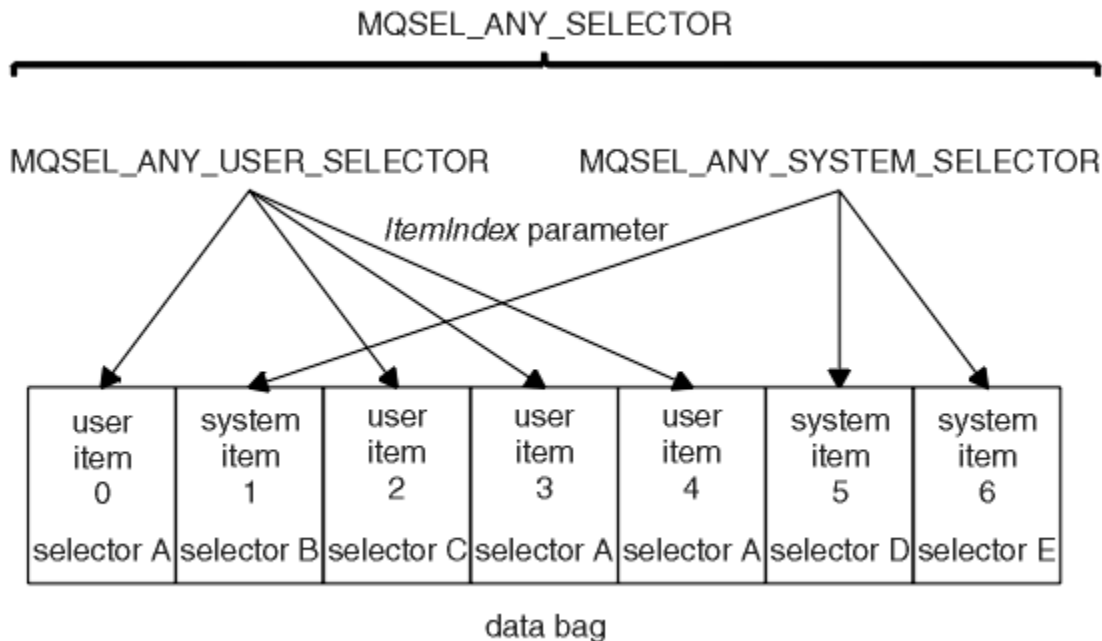


Abbildung 3. Indexierung

In [Abbildung 3 auf Seite 42](#) kann mit den folgenden Indexpaaren auf Benutzerelement 3 (Selektor A) verwiesen werden:

- selector A (ItemIndex 1)
- MQSEL_ANY_USER_SELECTOR (ItemIndex 2)
- MQSEL_ANY_SELECTOR (itemIndex 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 einer Tasche ändern“](#) auf Seite 71 und [„Datenelemente löschen“](#) auf Seite 74.

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.

Multi 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 dem 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 3 auf Seite 43 sind die Regeln dargestellt, die beim Konvertieren von Datenbehältern in Nachrichten und umgekehrt angewendet werden:

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

Tabelle 3. CCSID-Verarbeitung (Forts.)			
MQAI-Aufruf	CCSID	Eingabe des Aufrufs	Ausgabe des Aufrufs
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 Konzepte

„Datenkonvertierung zwischen codierten Zeichensätzen“ auf Seite 214

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.

„Die ccsid_part2.tbl -Datei“ auf Seite 216

Die Datei ccsid_part2.tbl wird verwendet, um zusätzliche CCSID-Informationen zu liefern. Die Datei ccsid_part2.tbl ersetzt die Datei ccsid.tbl, die vor IBM MQ 9.0 verwendet wurde.

Multi 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 4. 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 4 auf Seite 44 dargestellt.

Tabelle 5. 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 5 auf Seite 44 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 6 auf Seite 45](#) 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 5 auf Seite 44
<i>Expiry</i>	30 Sekunden (Anmerkung „1“ auf Seite 45)
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>CodedCharSetId</i>	Je nach Behälter-CCSID (Anmerkung „2“ auf Seite 45)
<i>Format</i>	Siehe Tabelle 5 auf Seite 44
<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 45
<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. Weitere Informationen finden Sie unter „Verarbeitung der Datenkonvertierung in der MQAI“ auf Seite 42.
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 Beispiel-C-Programm amqsaicq.c erstellt mithilfe der MQAI eine lokale Warteschlange.

```

/*****/
/*
/* Program name: AMQSAICQ
/*
/* 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, 2024
/*
/*****/
/*
/* 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)
    stncpy(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\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(MQCBO_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);
}
}

```

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

Das Beispiel-C-Programm amqsaiem.c veranschaulicht 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, 2024. 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(MQCBO_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, &iItemCount, &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 hdl*/
             &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;
}

```

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

Das Beispiel-C-Programm amqsaicl.c untersucht Kanalobjekte mit der MQAI.

```

/*****
/*
/* 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, 2024. 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 <cmqcfh.h> /* PCF */
#include <cmqbc.h> /* MQAI */
#include <cmqxc.h> /* MQCD

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

/*****
/* DataTypes */
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
typedef _RFIL FILE;
#else
typedef FILE FILE;
#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 */
"cluscdr  ", /* 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(hdl, 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);
}

```

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

Das Beispiel-C-Programm amqsailq.c stellt die aktuelle Tiefe der lokalen Warteschlangen mit der MQAI (MQAI) in Frage.

```

/*****
/*
/* 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, 2024
/*
/*
/*****
/*
/* 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);
}

```

Multi 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 Zeichenfolgefiter, eine Bytefolge, ein Bytefolgefiter oder ein Handle eines anderen Beutels sein kann. Eine detaillierte Beschreibung von Datenelementen finden Sie unter „Typen des in der MQAI verfügbaren Datenelements“ auf Seite 68

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 41 beschrieben.

Eine Hierarchie dieser Konzepte wird in Abbildung 1 dargestellt.

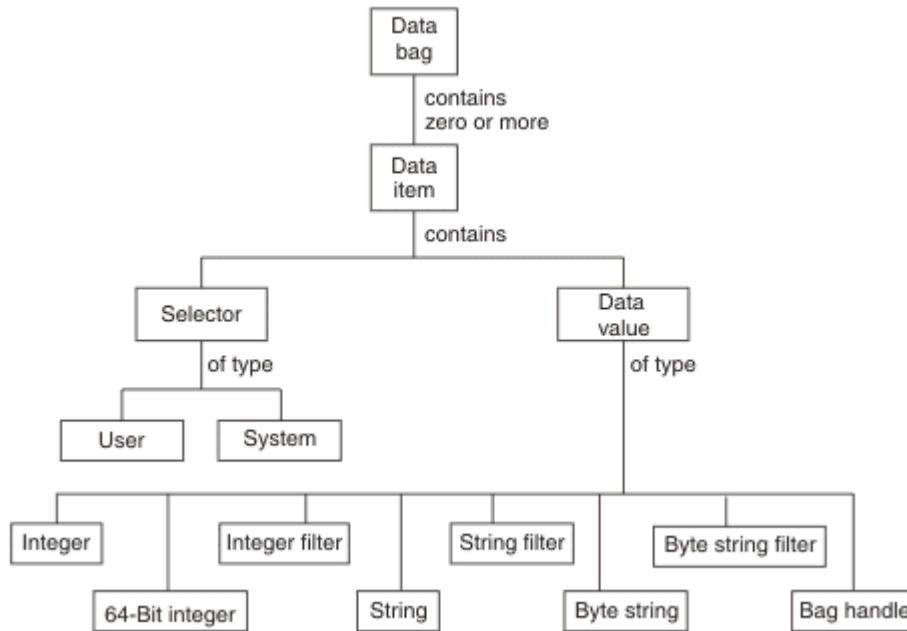


Abbildung 4. Hierarchie von MQAI-Konzepten

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 67 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 67.

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 67.
- Sie können mithilfe von Datenbehältern Daten zwischen Anwendungen versenden „[Datensäcke mit dem MQAI einschalten und empfangen](#)“ auf Seite 68.
- Sie können Datenbehältern Datenelemente hinzufügen „[Hinzufügen von Datenelementen zu Taschen mit der MQAI](#)“ auf Seite 69.
- Sie können innerhalb eines Datenbehälters einen Abfragebefehl hinzufügen „[Abfragebefehl zu einem Behälter hinzufügen](#)“ auf Seite 70.
- Sie können innerhalb von Datenbehältern Abfragen vornehmen „[Inquiring innerhalb von Datensäcken](#)“ auf Seite 71.
- Sie können Datenelemente innerhalb von Datenbehältern zählen „[Datenelemente zählen](#)“ auf Seite 73.
- Sie können Daten innerhalb eines Datenbehälters ändern „[Informationen in einer Tasche ändern](#)“ auf Seite 71.
- Sie können einen Datenbehälter leeren „[Inhalt einer Tasche mit dem mqClearBag-Aufruf löschen](#)“ auf Seite 72.
- Sie können einen Datenbehälter abschneiden „[Abschneiden eines Behälters mit dem mqTruncateBag-Aufruf](#)“ auf Seite 72.
- Sie können Behälter und Puffer konvertieren „[Konvertieren von Taschen und Puffern](#)“ auf Seite 73.

Multi

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 68.
- 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 75. Der Benutzercode kann eine Systemhülle nicht löschen.

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

Multi *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 unter [„PCF-Nachrichten in einer angegebenen Warteschlange senden und empfangen“](#) auf Seite 30

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.

Multi *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 69.

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 73.
- „[Datenelemente löschen](#)“ auf Seite 74.
- „[Hinzufügen von Datenelementen zu Taschen mit der MQAI](#)“ auf Seite 69.
- „[Datenelemente filtern und abfragen](#)“ auf Seite 70.

Multi *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 71), aber die Optionen zur Erstellung von Behältern können vom Benutzer nicht geändert werden. Systemelemente können nicht gelöscht werden. (Siehe „[Datenelemente löschen](#)“ auf Seite 74.)

Multi *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 68.

Mit der MQAI können Sie ganze Zahlen, 64-Bit-Ganzzahlelemente, Ganzzahlfilterelemente, Zeichenfolgeelemente, Zeichenfolgefilter, Bytefolgeelemente und Bytefolgefilterelemente zu Behältern hinzufügen. Dies wird in [Abbildung 5](#) auf Seite 69 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.

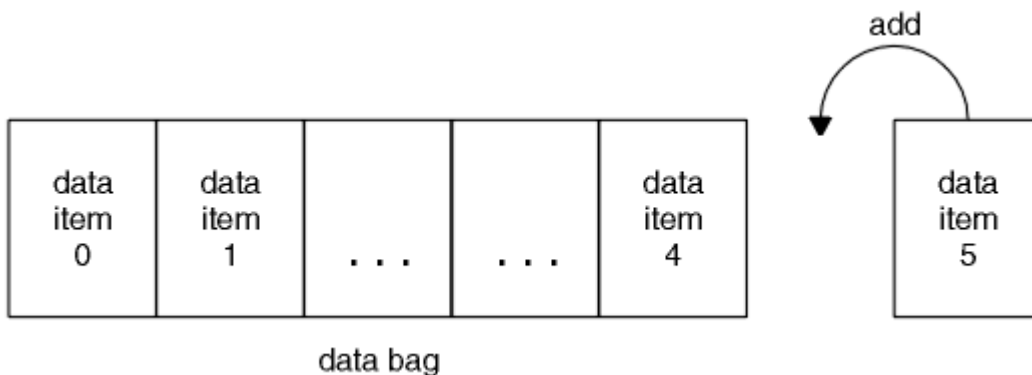


Abbildung 5. Datenelemente hinzufügen

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 69.

Multi

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](#).

Multi

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 27 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**. Geben 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, ...)
```

Multi 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](#).

Multi 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 6](#) auf Seite 71).

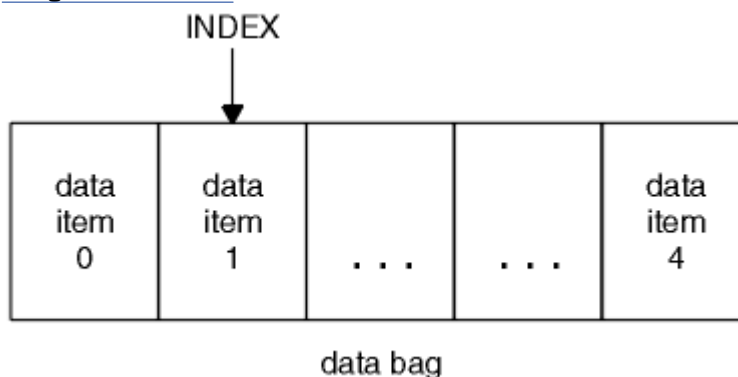


Abbildung 6. Ein einzelnes Datenelement ändern

2. Löschen Sie alle vorhandenen Vorkommen des angegebenen Selektors und fügen Sie dem Ende des Sacks ein neues Vorkommen hinzu. (Siehe [Abbildung 7](#) auf Seite 72.) Ein spezieller Indexwert ermöglicht **alle** Instanzen eines Parameters, der ersetzt werden soll.

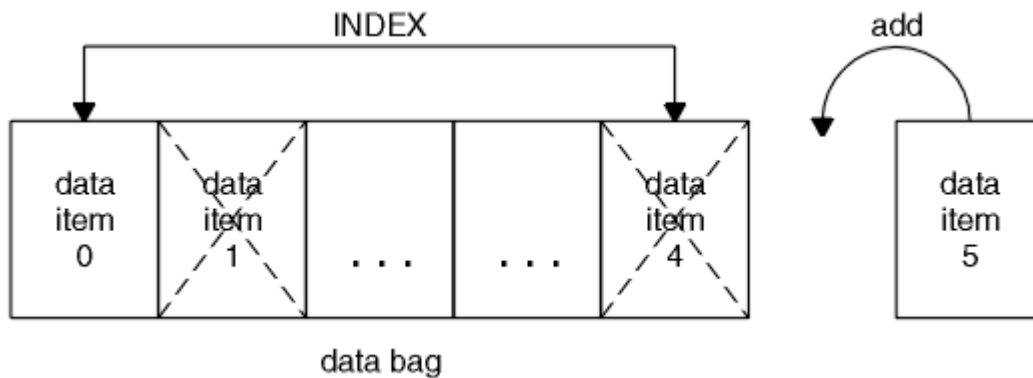


Abbildung 7. Alle Datenelemente ändern

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](#)

Multi *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](#).

Multi *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.

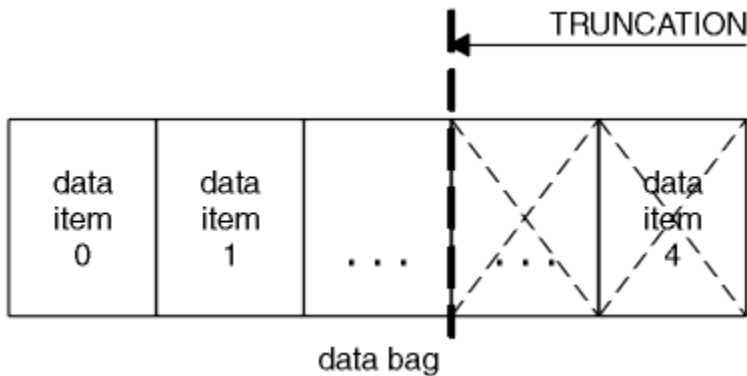


Abbildung 8. Einen Behälter abschneiden

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

Multi 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 9](#) auf Seite 73 gezeigt. Eine vollständige Beschreibung des Aufrufs '`mqBagToBuffer`' finden Sie in [mqBagToBuffer](#).

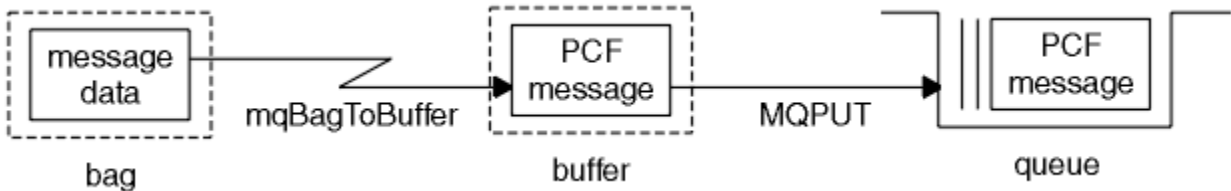


Abbildung 9. Konvertieren von Taschen in PCF-Nachrichten

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 10](#) auf Seite 73 gezeigt. Eine vollständige Beschreibung des `mqBufferToBag`-Aufrufs finden Sie unter [mqBufferToBag](#).

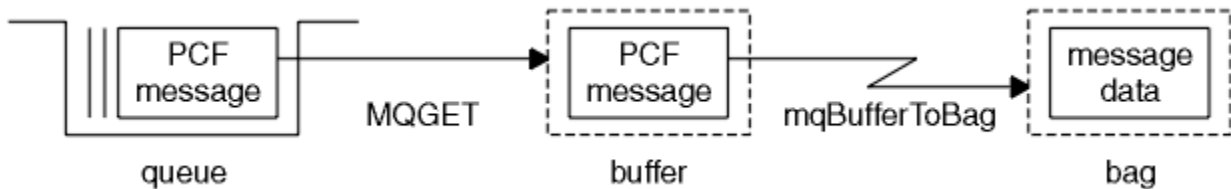


Abbildung 10. PCF-Nachrichten in Beutelformular umwandeln

Multi 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](#).

Multi 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 74.
- Alle Benutzerelemente eines Behälters löschen, d. h. den Behälter leeren. Weitere Informationen finden Sie unter „[Inhalt einer Tasche mit dem mqClearBag-Aufruf löschen](#)“ auf Seite 72.
- Benutzerelemente vom Ende eines Behälters löschen, d. h. einen Behälter abschneiden. Ausführliche Informationen hierzu finden Sie unter „[Abschneiden eines Behälters mit dem mqTruncateBag-Aufruf](#)“ auf Seite 72.

Multi 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 11](#) auf Seite 74.)

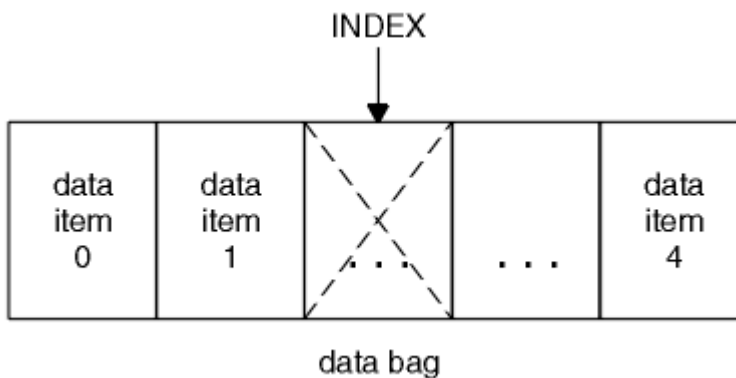


Abbildung 11. Löschen eines einzelnen Datenelements

oder

2. Alle Vorkommen des angegebenen Selektors. (Siehe [Abbildung 12](#) auf Seite 74.)

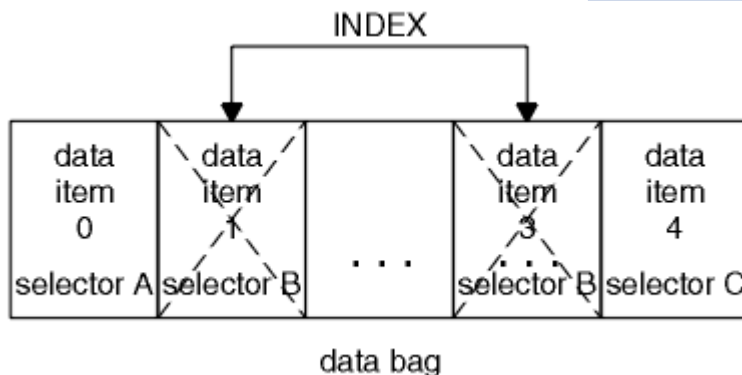


Abbildung 12. Alle Datenelemente löschen

Anmerkung: Der Index erfüllt die Einfügeföolge 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 13 auf Seite 75](#) gezeigt.

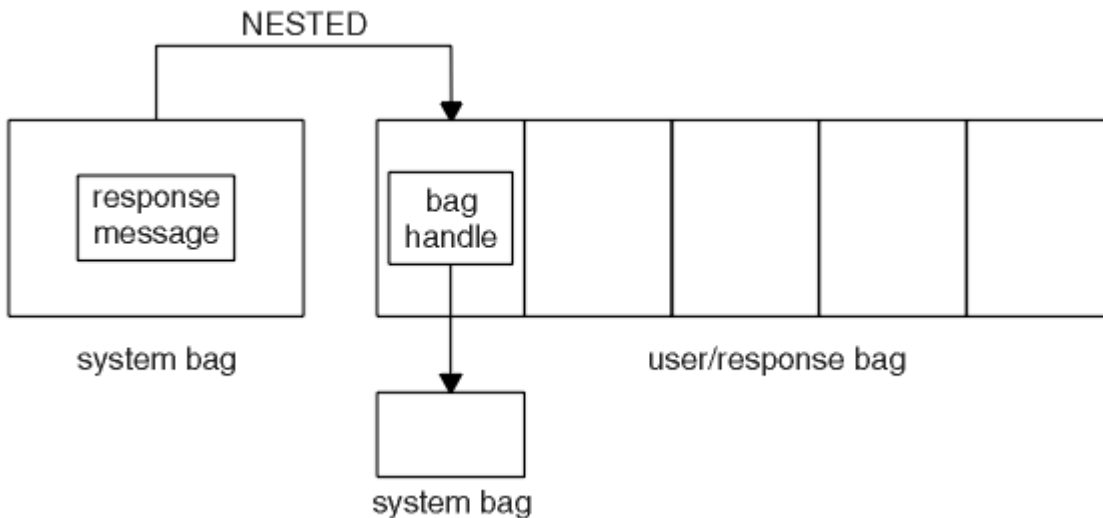


Abbildung 13. Verschachteln

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](#).

Verwaltung über die REST API

Sie können die administrative REST API verwenden, um IBM MQ-Objekte zu verwalten, wie z. B. Warteschlangenmanager und Warteschlangen sowie Managed File Transfer-Agenten und -Übertragungen. Informationen werden an die administrative REST API in JSON-Format gesendet und von dieser empfangen. Mithilfe dieser REST-konformen APIs können Sie die IBM MQ-Verwaltung in gängige DevOps und Automatisierungstools einbetten.

Vorbereitende Schritte

Anmerkung: **V9.4.0** administrative REST API ist in einer eigenständigen IBM MQ Web Server -Installation nicht verfügbar. Weitere Informationen zu den Installationsoptionen für die IBM MQ -Komponente, die die administrative REST API ausführt, finden Sie unter [IBM MQ Console und REST API](#).

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

Prozedur

- „[Erste Schritte mit der administrative REST API](#)“ auf Seite 76
- „[administrative REST API verwenden](#)“ auf Seite 80
- „[Fernverwaltung über die REST API](#)“ auf Seite 82
- „[REST API-Zeitmarken](#)“ auf Seite 86
- „[REST API-Fehlerbehandlung](#)“ auf Seite 86
- „[REST API-Erkennung](#)“ auf Seite 89
- „[Unterstützung landessprachunterstützter Sprachen in REST API](#)“ auf Seite 90

Erste Schritte mit der administrative REST API

Starten Sie schnell mit der administrative REST API und probieren Sie einige Beispielanforderungen aus, indem Sie cURL verwenden, um eine Warteschlange zu erstellen, zu aktualisieren, anzuzeigen und zu löschen.

Vorbereitende Schritte

Um Ihnen den Einstieg in die Verwendung der administrative REST API zu erleichtern, gelten für die Beispiele in dieser Aufgabe folgende Voraussetzungen:

- Die Beispiele verwenden cURL, um REST-Anforderungen zum Anzeigen von Informationen zu Warteschlangenmanagern auf dem System zu machen und eine Warteschlange zu erstellen, eine Warteschlange zu aktualisieren, anzuzeigen und zu löschen. Deshalb muss cURL auf Ihrem System installiert sein, damit Sie diese Aufgabe ausführen können.
- Für die Ausführung dieser Task müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie den Befehl `dspmweb` verwenden können:

- **z/OS** Unter z/OS benötigen Sie die Berechtigung zur Ausführung des Befehls **dspmqweb** sowie Schreibzugriff für die Datei `mqwebuser.xml`.
- **Multi** Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
- **IBM i** Unter IBM i sollten die Befehle in QSHELL ausgeführt werden.

Vorgehensweise

1. Stellen Sie sicher, dass Sie den mqweb-Server für die Verwendung durch die administrative REST API, die administrative REST API für MFT, die messaging REST API oder IBM MQ Console konfiguriert haben. Weitere Informationen zur Konfiguration des mqweb-Servers mit einer Basisregistry enthält der Artikel [Basiskonfiguration für den mqweb-Server](#).

2. **z/OS** Setzen Sie unter z/OS die Umgebungsvariable `WLP_USER_DIR` so, dass Sie den Befehl **dspmqweb** verwenden können. Geben Sie folgenden Befehl ein, um die Variable so zu setzen, dass sie auf Ihre mqweb-Serverkonfiguration zeigt:

```
export WLP_USER_DIR=WLP_user_directory
```

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

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

Weitere Informationen finden Sie im Abschnitt [mqweb-Server erstellen](#).

3. Bestimmen Sie die REST API URL, indem Sie den folgenden Befehl eingeben:

```
dspmqweb status
```

Die Beispiele in den folgenden Schritten gehen davon aus, dass Ihre REST API URL die Standard-URL `https://localhost:9443/ibmmq/rest/v1/` ist. Sollte Ihre URL von der Standard-URL abweichen, ersetzen Sie die URL in den folgenden Schritten durch Ihre URL.

4. Versuchen Sie eine GET-Anforderung für die `qmgr`-Ressource, indem Sie die Basisauthentifizierung mit dem `mqadmin`-Benutzer verwenden:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/qmgr -X GET -u mqadmin:mqadmin
```

5. Verwenden Sie zum Erstellen, Anzeigen, Ändern und Löschen einer Warteschlange die Ressource `mqsc`:

In diesem Beispiel wird ein WS-Manager `QM1` verwendet. Erstellen Sie entweder einen Warteschlangenmanager mit diesem Namen oder ersetzen Sie einen vorhandenen Warteschlangenmanager auf Ihrem System.

- a) Erstellen Sie eine POST-Anforderung für die Ressource `mqsc`, um die lokale Warteschlange zu erstellen:

Im Hauptteil der Anforderung wird der Name der neuen Warteschlange auf `Q1` gesetzt. Es wird die Basisauthentifizierung verwendet und ein HTTP-Header namens `ibm-mq-rest-csrf-token` mit einem beliebigen Wert in der cURL-REST-Anforderung festgelegt. Dieser zusätzliche Header ist für POST-, PATCH- und DELETE-Anforderungen erforderlich:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "define", "qualifier": "qlocal", "name": "Q1"}'
```

- b) Erstellen Sie eine POST-Anforderung für die Ressource `mqsc`, um die in Schritt „5.a“ auf Seite 77 erstellte lokale Warteschlange anzuzeigen:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1"}'
```

- c) Erstellen Sie eine POST-Anforderung für die Ressource mqsc, um die Beschreibung der Warteschlange zu aktualisieren:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "alter", "qualifier": "qlocal", "name": "Q1", "parameters": {"descr": "new description"}'}
```

- d) Erstellen Sie eine POST-Anforderung für die Ressource mqsc, um die Beschreibung der neuen Warteschlange anzuzeigen. Geben Sie das Attribut **responseParameters** im Anforderungshauptteil an, damit die Antwort das Beschreibungsfeld enthält:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1", "responseParameters": [{"descr"}]}'
```

- e) Erstellen Sie eine POST-Anforderung für die Ressource mqsc, um die Warteschlange zu löschen:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "delete", "qualifier": "qlocal", "name": "Q1"}'
```

- f) Erstellen Sie eine POST-Anforderung für die Ressource mqsc, um zu beweisen, dass die Warteschlange gelöscht wurde:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/action/qmgr/QM1/mqsc -X POST -u mqadmin:mqadmin -H "ibm-mq-rest-csrf-token: value" -H "Content-Type: application/json" --data '{"type": "runCommandJSON", "command": "display", "qualifier": "qlocal", "name": "Q1"}'
```

Nächste Schritte

- In den Beispielen wird die Anforderung durch die Basisauthentifizierung geschützt. Stattdessen können Sie auch die tokenbasierte Authentifizierung oder die clientbasierte Authentifizierung verwenden. Weitere Informationen finden Sie im Abschnitt [Clientzertifikatsauthentifizierung mit der REST API und IBM MQ Console verwenden](#) und [Tokenbasierte Authentifizierung mit der REST API verwenden](#).
- Weitere Informationen zur Verwendung der administrative REST API und zum Erstellen von URLs mit Abfrageparametern finden Sie im Abschnitt „administrative REST API verwenden“ auf Seite 80.
- Durchsuchen Sie die Referenzinformationen für die verfügbaren administrative REST API-Ressourcen und alle verfügbaren optionalen Abfrageparameter: [administrative REST API-Referenz](#).
- Hier erfahren Sie, wie Sie die administrative REST API zum Verwalten von IBM MQ-Objekten auf fernen Systemen verwenden: „Fernverwaltung über die REST API“ auf Seite 82.
- Hier erfahren Sie, wie Sie die administrative REST API mit MFT: „Erste Schritte mit der REST API für MFT“ auf Seite 78 verwenden.
- Lernen Sie die messaging REST API kennen, eine REST-konforme Schnittstelle für IBM MQ-Messaging: [Messaging mit der REST API](#).
- Lernen Sie die IBM MQ Console kennen, eine browserbasierte GUI: „Verwaltung über die IBM MQ Console“ auf Seite 94.

Erste Schritte mit der REST API für MFT

Starten Sie schnell mit der administrative REST API für Managed File Transfer und probieren Sie einige Beispielanforderungen aus, um den Status des MFT-Agenten und eine Liste von Übertragungen anzuzeigen.

Vorbereitende Schritte

- Die Beispiele verwenden cURL, um REST-Anforderungen zu senden und eine Liste der Übertragungen sowie den Status des MFT-Agenten anzuzeigen. Deshalb muss cURL auf Ihrem System installiert sein, damit Sie diese Aufgabe ausführen können.
- Für die Ausführung dieser Task müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie den Befehl **dspmqweb** verwenden können:
 - **z/OS** Unter z/OS benötigen Sie die Berechtigung zur Ausführung des Befehls **dspmqweb** sowie Schreibzugriff für die Datei mqwebuser.xml.
 - **Multi** Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.

Vorgehensweise

1. Stellen Sie sicher, dass der mqweb-Server für die administrative REST API für MFT konfiguriert ist:
 - Stellen Sie sicher, dass Sie den mqweb-Server für die Verwendung durch die administrative REST API, die administrative REST API für MFT, die messaging REST API oder IBM MQ Console konfiguriert haben. Weitere Informationen zur Konfiguration des mqweb-Servers mit einer Basisregistry enthält der Artikel [Basiskonfiguration für den mqweb-Server](#).
 - Wenn der mqweb-Server konfiguriert ist, stellen Sie sicher, dass Schritt 8 der [Basiskonfiguration für den mqweb-Server](#) abgeschlossen wurde, um die administrative REST API für MFT zu aktivieren.

2. **z/OS**

Setzen Sie unter z/OS die Umgebungsvariable WLP_USER_DIR so, dass Sie den Befehl **dspmqweb** verwenden können. Geben Sie folgenden Befehl ein, um die Variable so zu setzen, dass sie auf Ihre mqweb-Serverkonfiguration zeigt:

```
export WLP_USER_DIR=WLP_user_directory
```

, wobei *WLP_user_directory* der Name des Verzeichnisses ist, das an crtmqweb übergeben wird.
Beispiel:

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

Weitere Informationen finden Sie im Abschnitt [mqweb-Server erstellen](#).

3. Bestimmen Sie die REST API URL, indem Sie den folgenden Befehl eingeben:

```
dspmqweb status
```

Die Beispiele in den folgenden Schritten gehen davon aus, dass Ihre REST API URL die Standard-URL `https://localhost:9443/ibmmq/rest/v1/` ist. Sollte Ihre URL von der Standard-URL abweichen, ersetzen Sie die URL in den folgenden Schritten durch Ihre URL.

4. Erstellen Sie eine GET-Anforderung in der `agent`-Ressource, um grundlegende Details zu allen Agenten zurückzugeben, einschließlich Name, Typ und Status:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/agent/ -X GET -u mftadmin:mftadmin
```

5. Erstellen Sie einige anzuzeigende Übertragungen mit dem Befehl **fteCreateTransfer**.

Der mqweb-Server speichert Informationen zu Übertragungen im Cache und gibt diese Informationen zurück, wenn eine Anforderung gestellt wird. Dieser Cache wird zurückgesetzt, wenn der mqweb-Server erneut gestartet wird. Sie können prüfen, ob der Server erneut gestartet wurde, indem Sie die `console.log`- und `messages.log`-Dateien oder unter z/OS die Ausgabe der gestarteten Task anzeigen.

6. Erstellen Sie eine GET-Anforderung in der `transfer`-Ressource, um Details von bis zu vier Übertragungen zurückzugeben, die seit dem Start des mqweb-Servers vorgenommen wurden:

```
curl -k https://localhost:9443/ibmmq/rest/v2/admin/mft/transfer?limit=4 -X GET -u mftadmin:mftadmin
```

Nächste Schritte

- In den Beispielen wird die Anforderung durch die Basisauthentifizierung geschützt. Stattdessen können Sie auch die tokenbasierte Authentifizierung oder die clientbasierte Authentifizierung verwenden. Weitere Informationen finden Sie in den Abschnitten [Tokenbasierte Authentifizierung mit der REST API verwenden](#) und [Clientzertifikatsauthentifizierung mit der REST API und IBM MQ Console verwenden](#).
- Weitere Informationen zur Verwendung der administrative REST API und zum Erstellen von URLs mit Abfrageparametern finden Sie im Abschnitt „[administrative REST API verwenden](#)“ auf Seite 80.
- Durchsuchen Sie die Referenzinformationen für die verfügbare administrative REST API für MFT-Ressourcen und alle verfügbaren optionalen Abfrageparameter: [administrative REST API-Referenz](#).
- Lernen Sie die messaging REST API kennen, eine REST-konforme Schnittstelle für IBM MQ-Messaging: [Messaging mit der REST API](#).
- Lernen Sie die IBM MQ Console kennen, eine browserbasierte GUI: „[Verwaltung über die IBM MQ Console](#)“ auf Seite 94.

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:

- 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 unter [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 (/) muss als %2F codiert werden.
 - Ein Prozentzeichen (%) muss als %25 codiert werden.
- 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 81 und „1.b“ auf Seite 81 ausführen.

Vorgehensweise

1. Konstrukt die URL:

- a) Ermitteln Sie die Präfix-URL, indem Sie den folgenden Befehl eingeben:

```
dspmweb status
```

Die URL, die Sie verwenden möchten, enthält die Phrase `/ibmmq/rest/`.

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

Die folgenden IBM MQ-Ressourcen sind verfügbar:

- [/admin/installation](#)
- [/admin/qmgr](#)
- [/admin/queue](#)
- [/admin/subscription](#)
- [/admin/channel](#)
- [/action/qmgr/{ qmgrname }/mqsc](#)

Die folgenden Managed File Transfer-Ressourcen sind verfügbar:

- [/admin/agent](#)
- [/admin/transfer](#)
- [/admin/monitor](#)

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

```
https://localhost:9443/ibmmq/rest/v2/admin/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:

```
https://localhost:9443/ibmmq/rest/v2/admin/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:

```
https://localhost:9443/ibmmq/rest/v2/admin/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 Sicherheitsberechtigungs-nachweise für die Authentifizierung an. For example:

- 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.

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 76.
- 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 an die Ressource `queue` zurückgegeben wird.
- Sie müssen eine der folgenden unterstützten REST-Ressourcen verwenden:
 - `/queue`
 - `/subscription`
 - `/channel`
 - `/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 entweder der Befehl **setmqweb** verwendet wird, um das Attribut `mqRestGatewayQmgr` auf den Namen des Gateway-Warteschlangenmanagers zu setzen, 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 [„WS-Manager für die Fernverwaltung konfigurieren“](#) auf Seite 207.
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.

Wenn Sie Schwierigkeiten bei Verwendung der administrative REST API zur Verwaltung ferner Warteschlangenmanager haben:

- Überprüfen Sie, ob der ferne WS-Manager aktiv ist.
- Überprüfen Sie, ob der Befehlsserver auf dem fernen System ausgeführt wird.
- Überprüfen Sie, ob das Intervall für die Kanalunterbrechung abgelaufen ist. Beispiel: Wenn ein Kanal gestartet wurde, aber dann nach einiger Zeit heruntergefahren wurde. Dies ist besonders wichtig, wenn Sie die Kanäle manuell starten.

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 mquser1 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:

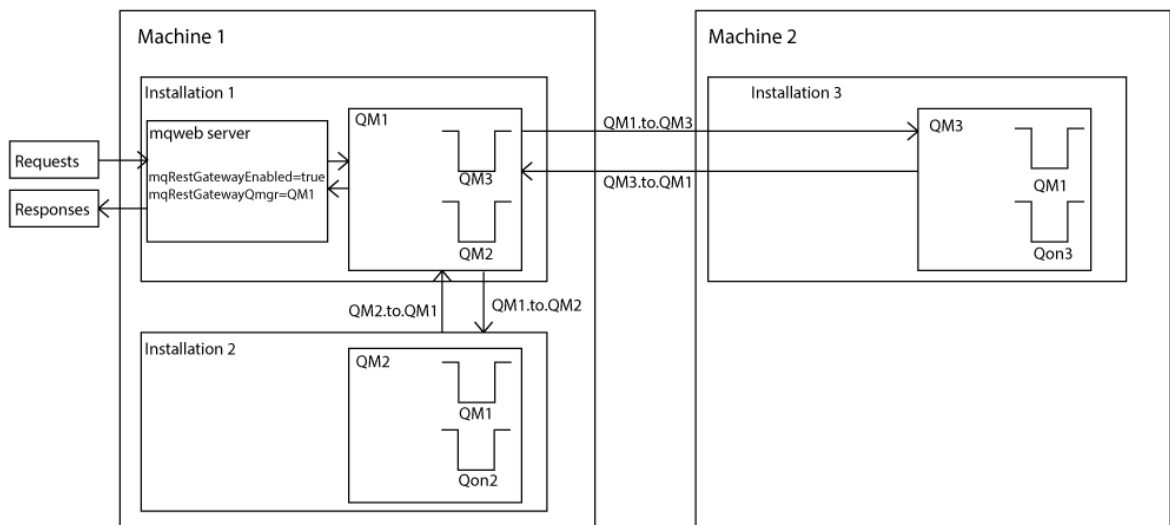


Abbildung 14. Diagramm einer Beispielkonfiguration für Fernverwaltung unter Verwendung der REST API

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"
  }
]
```

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.

Die Genauigkeit dieser Zeitmarke ist nicht gewährleistet. 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, in dem angezeigt wird, dass die Nachricht ursprünglich in der IBM MQ Rest-API erstellt wurde.

nnnn

Eine eindeutige Zahl, mit der die Nachricht angegeben wird.

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.

das Kundenstamms

Zeichenfolge.

Eine Beschreibung des Fehlers.

Erläuterung

Zeichenfolge.


Eine Erläuterung des Fehlers.

Handlung

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** angegeben haben.

Zeichenfolge.

Der Name des Warteschlangenmanagers, der den Fehler festgestellt hat.

Dieses Feld ist für die messaging REST API nicht zutreffend.

completionCode

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 von Java -oder JMS-Ausnahmen. 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




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 MFT REST API -Services aktiviert sind und der Koordinationswarteschlangenmanager nicht in der `mqwebuser.xml` -Datei festgelegt ist, erhalten Sie die folgende Ausnahme:

```
{"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"
}]}
```

REST API-Erkennung

Die Dokumentation für die REST API ist in der IBM Documentation und im Swagger-Format verfügbar. Swagger ist eine gängige Methode zur Dokumentation von REST-APIs. Die Swagger-Dokumentation für REST API kann angezeigt werden, indem die API-Erkennungsfunktion (`apiDiscovery`) auf dem `mqweb`-Server aktiviert wird.

Vorbereitende Schritte

Stabilized

Wichtig: Das Feature `apiDiscovery` wurde stabilisiert. Sie können diese Funktion weiterhin nutzen. Derzeit unterstützt IBM MQ die Verwendung des Features `mpOpenAPI` nicht.

Sie müssen die Sicherheit für den `mqweb`-Server aktivieren, um die Swagger-Dokumentation mithilfe der API-Erkennung anzeigen zu können. Weitere Informationen über die Schritte, die zur Aktivierung der Sicherheit erforderlich sind, finden Sie im Abschnitt [IBM MQ Console- und REST API-Sicherheit](#).

Vorgehensweise

1. Suchen Sie die Datei `mqwebuser.xml` in einem der folgenden Verzeichnisse:

- ▶ **ALW** `MQ_DATA_PATH/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` ausgeführt wurde, um die `mqweb`-Serverdefinition zu erstellen.

2. Fügen Sie die entsprechende XML-Datei 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. Der Inhalt dieses Headers kann eine beliebige Zeichenfolge sein, einschließlich leer.

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-Attacken 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. Verwenden Sie den Befehl `dspmweb status`, um die URL anzuzeigen.

Zugehörige Informationen

[dspmweb-Status \(mqweb-Serverstatus anzeigen\)](#)

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 [91](#) 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

Tschechisch

Durch den Sprachentag cs verweigert

Französisch

Durch den Sprachentag fr verweigert

Ungarisch

Durch den Sprachentag hu verweigert

Italienisch

Durch den Sprachentag it verweigert

Japanisch

Durch den Sprachentag ja verweigert

Koreanisch

Durch den Sprachentag ko verweigert

Polnisch

Durch den Sprachentag pl verweigert

Portugiesisch (Brasilien)

Durch den Sprachentag pt_BR verweigert

Russisch

Durch den Sprachentag ru verweigert

Spanisch

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 Antwortheadern 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 Antwortheadern 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 Antwortheadern 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.

Versionen der REST API

Die Versionsnummer der REST API stellt einen Teil der Basis-URL für REST-Anforderungen dar. Beispiel: `https://localhost:9443/ibmmq/rest/v2/admin/installation`. Die Versionsnummer wird verwendet, um Clients von Änderungen an der REST API zu isolieren, die möglicherweise in künftigen Releases eingeführt werden.

IBM MQ 9.2.0 führt die Version 2 des REST API ein. Diese Versionserhöhung gilt für die administrative REST API, messaging REST API und MFT REST API. Mit der Versionserhöhung ändert sich die Ressourcen-URL, die für die REST API verwendet wird. Das URL-Präfix für die Ressourcen-URLs in Version 2 ist folgende URL:

```
https://host:port/ibmmq/rest/v2/
```

Stabilized Einige Änderungen, die in der REST API eingeführt werden, können die vorhandene REST API-Funktion so ändern, dass Clients, die die REST API verwenden, möglicherweise aktualisiert werden müssen. Um zu verhindern, dass solche Änderungen zu einer erzwungenen Aktualisierung von Clients führen, wird die REST API-Versionsnummer erhöht und die vorhandene Funktion mit der vorherigen Nummer stabilisiert. Die neue Funktion, mit der die vorhandene Funktion geändert werden kann, wird der REST API mit der neuen Versionsnummer hinzugefügt. Daher können Clients weiterhin die REST API in der vorherigen Version verwenden, ohne aktualisiert zu werden.

Zu REST API-Änderungen, die dazu führen könnten, dass eine Clientaktualisierung erforderlich ist, gehören folgende Änderungen:

- Entfernen der Unterstützung für ein vorhandenes Attribut im JSON, das an die REST API gesendet oder von ihr zurückgegeben wird.
- Entfernen einer URL, eines HTTP-Verbs oder eines Headers. Beispiel: Wenn eine URL oder ein Header umbenannt wird oder wenn ein anderes Verb verwendet wird.
- Das Hinzufügen eines neuen obligatorischen JSON-Attributs zu Daten, die an eine vorhandene URL gesendet werden.
- Das Hinzufügen eines neuen obligatorischen HTTP-Headers zu Daten, die an eine vorhandene URL gesendet werden.
- Das Hinzufügen eines neuen obligatorischen Abfrageparameters zu einer vorhandenen URL.

Wenn diese Art der Änderung in der Funktion REST API eingeführt wird, die in einem Release von Long Term Support (LTS) vorhanden war, wird die Versionsnummer des REST API für den ersten dieser Änderungen erhöht. Alle nachfolgenden Änderungen, die in einem Continuous Delivery-(CD-)Release vorgenommen werden und möglicherweise Änderungen an Clients, die die REST API verwenden, erfordern, verwenden die neue Versionsnummer.

Diese Versionsnummer bleibt in allen nachfolgenden CD-Releases bis zum nächsten Release von LTS unverändert. Deshalb wird die Versionsnummer zwischen LTS-Releases höchstens einmal erhöht.

Stabilized Bei einer Erhöhung der Versionsnummer wird die vorhandene REST API-Funktion auf der alten Versionsnummer stabilisiert. Das heißt, die vorhandene REST API-Funktion, die im LTS-Release verfügbar war, bleibt bei der alten Versionsnummer verfügbar, aber es werden keine weiteren Änderungen an dieser Version vorgenommen. Alle neuen Funktionen, die zur REST API hinzugefügt werden, werden zur neuen REST API-Version hinzugefügt. Für Hinzufügungen, die an der REST API in CD-Releases vor der Versionserhöhung vorgenommen werden, gibt es jedoch keine Garantie, dass sie in die ältere Version der REST API aufgenommen werden.

Deprecated Vorhandene Clients können die REST API weiterhin mit der alten Versionsnummer verwenden, ohne dass Änderungen erforderlich sind. Ältere Versionen von REST API werden möglicherweise nicht weiter unterstützt und irgendwann entfernt.

Einige Änderungen erfordern keine Änderungen an Clients, die die REST API verwenden. Diese Änderungen führen nicht zu einer Erhöhung der Versionsnummer. Stellen Sie daher sicher, dass jeder Client, der die REST API verwendet, nicht aktualisiert werden muss, wenn diese Arten von Änderungen vorgenommen werden. Zu diesen Änderungen der REST API können folgende Änderungen gehören:

- Das Hinzufügen eines neuen JSON-Attributs zu vorhandenen Daten, die von der REST API zurückgegeben werden.
- Das Hinzufügen einer neuen URL.
- Das Hinzufügen eines neuen HTTP-Verbs zu einer vorhandenen URL.
- Hinzufügen eines neuen Statuscodes zu einer vorhandenen URL.
- Das Hinzufügen neuer optionaler JSON-Attribute zu Daten, die an eine vorhandene URL gesendet werden.
- Das Hinzufügen neuer Abfrageparameter in einer vorhandenen URL.
- Das Hinzufügen neuer Header zu Daten, die an eine vorhandene URL gesendet werden.
- Rückgabe neuer Header von der REST API.

Änderungen an neuer REST-API-Funktion von Continuous Delivery

Bei einer neuen REST API -Funktion, die in einem CD -Release hinzugefügt wird, erhöhen alle Änderungen, die an dieser neuen Funktion vorgenommen werden, die dann möglicherweise Änderungen an REST API -Clients erfordern, die Versionsnummer nicht. Dies bedeutet, dass die neue Funktion vor dem nächsten LTS-Release geändert werden kann, ohne dass die Versionsnummer erhöht wird. Wenn die Funktion in ein LTS-Release eingeschlossen ist, wird bei allen nachfolgenden Änderungen, die möglicherweise Änderungen an REST API-Clients erfordern, die Versionsnummer erhöht.

Beispiel

1. Bei LTS Release X befindet sich der REST API in Version 1.
2. In CD-Release X.0.1 wird Unterstützung für eine neue URL hinzugefügt. Diese Änderung erfordert keine Änderungen an Clients, die die REST API verwenden. Daher bleibt der REST API bei Version 1.
3. In CD X.0.2 wird Unterstützung für eine neue URL hinzugefügt. Diese Änderung erfordert keine Änderungen an Clients, die die REST-API verwenden. Daher bleibt der REST API bei Version 1.
4. Bei LTS -Release Y befindet sich der REST API in Version 1.
5. In CD-Release Y.0.1 wird eine vorhandene URL umbenannt. Diese Änderung macht möglicherweise Änderungen an Clients, die die REST API verwenden, erforderlich. Daher wird eine neue Version von REST API als Version 2 erstellt. Die umbenannte URL ist zusammen mit der gesamten vorhandenen Funktion in Version 2 von REST API enthalten. Jede neue Funktion, die dem REST API hinzugefügt wird, wird zu Version 2 hinzugefügt. Version 1 bleibt auf der Ebene in LTS -Release Y stabilisiert.
6. In CD-Release Y.0.2 wird eine weitere vorhandene URL umbenannt. Da die Version bereits in CD -Release Y erhöht ist, bleibt der REST API bei Version 2. Version 1 bleibt auf der Ebene in LTS -Release Y stabilisiert.

7. Bei LTS -Release Z bleibt der REST API bei Version 2. Version 1 bleibt auf der Ebene in LTS -Release Y stabilisiert.

Verwaltung über die IBM MQ Console

Sie können grundlegende Verwaltungstasks mit IBM MQ Console ausführen.

Anmerkung: Inaktivieren Sie bei Verwendung der IBM MQ Console nicht den Befehlsserver auf einem Ihrer Warteschlangenmanager. Wenn der Befehlsserver für einen Warteschlangenmanager inaktiviert ist:

- IBM MQ Console reagiert nicht mehr mit langen Verzögerungen bei der Verarbeitung von Befehlen
- Alle Befehle, die an den Warteschlangenmanager ausgegeben werden, haben das Zeitlimit überschritten.

Zugehörige Tasks




[Traceerstellung für IBM MQ Console](#)

Erste Schritte mit der IBM MQ Console

Konfigurieren Sie den mqweb-Server; ermitteln Sie den URI für IBM MQ Console; stellen Sie eine Verbindung zur Konsole her; melden Sie sich an der Konsole an.



Vorbereitende Schritte

Für die Ausführung dieser Task müssen Sie ein Benutzer mit bestimmten Berechtigungen sein, damit Sie den Befehl **dspmqweb** verwenden können:

-  Unter z/OS benötigen Sie die Berechtigung zur Ausführung des Befehls **dspmqweb** sowie Schreibzugriff für die Datei mqwebuser.xml.
-  Auf allen anderen Betriebssystemen müssen Sie ein privilegierter Benutzer sein.
-  Unter IBM i sollten die Befehle in QSHELL ausgeführt werden.

Informationen zu diesem Vorgang

Beachten Sie die folgenden Einschränkungen:

- 
 - 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.
 - 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.
- 
 - Sie können die IBM MQ Console nicht für die Arbeit mit AMQP-Kanälen verwenden.
 - Sie können IBM MQ Console nicht für die Arbeit mit MQTT-Kanälen verwenden.

Vorgehensweise

1. Falls der mqweb-Server noch nicht für die Verwendung durch die IBM MQ Console konfiguriert wurde, konfigurieren Sie den mqweb-Server.

Weitere Informationen zur Konfiguration des mqweb-Servers mit einer Basisregistry enthält der Artikel [Basiskonfiguration für den mqweb-Server](#).

2. 

Setzen Sie unter z/OS die Umgebungsvariable `WLP_USER_DIR` so, dass Sie den Befehl **dspmqweb** verwenden können. Geben Sie folgenden Befehl ein, um die Variable so zu setzen, dass sie auf Ihre mqweb-Serverkonfiguration zeigt:

```
export WLP_USER_DIR=WLP_user_directory
```

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

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

Weitere Informationen finden Sie im Abschnitt [mqweb-Server erstellen](#).

3. Ermitteln Sie die URI für die IBM MQ Console, indem Sie den folgenden Befehl eingeben:

```
dspmqweb status
```

Der Befehl generiert eine Ausgabe, die etwa folgendermaßen aussieht:

```
MQWB1124I: Server 'mqweb' is running.  
URLS:  
https://localhost:9443/ibmmq/rest/v1/  
https://localhost:9443/ibmmq/console/
```

Die URI für die IBM MQ Console endet mit dem Suffix `console/`.

4. Stellen Sie eine Verbindung zu IBM MQ Console her, indem Sie die URL aus dem vorherigen Schritt in einem Browser eingeben.

Eine Sicherheitsausnahme kann vom Browser erzeugt werden, da das Standardzertifikat, das mit dem mqweb-Server bereitgestellt wird, kein vertrauenswürdiges Zertifikat ist. Gehen Sie zur IBM MQ Console.

5. Melden Sie sich am IBM MQ Console an. Verwenden Sie den Benutzernamen `mqadmin` und das Kennwort `mqadmin`.

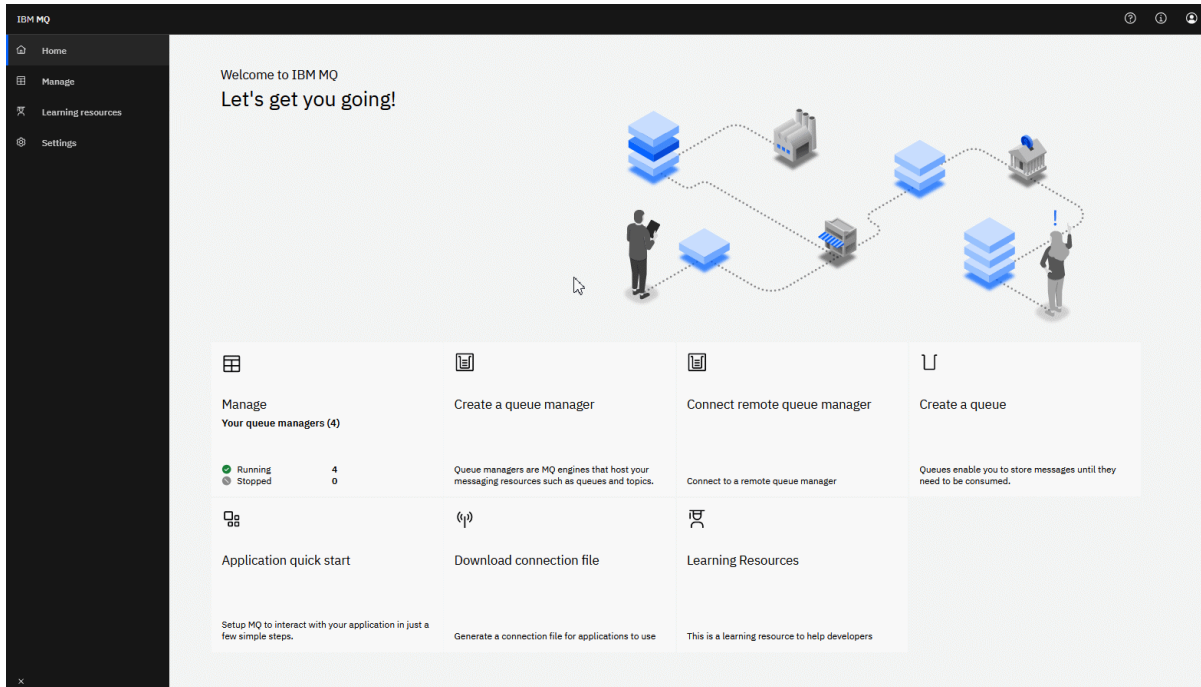
Nächste Schritte

Standardmäßig verwendet die IBM MQ Console tokenbasierte Authentifizierung für die Authentifizierung von Benutzern. Sie können auch die Clientzertifikatsauthentifizierung verwenden. Weitere Informationen finden Sie unter [Clientzertifikatsauthentifizierung mit REST API und IBM MQ Console verwenden](#).

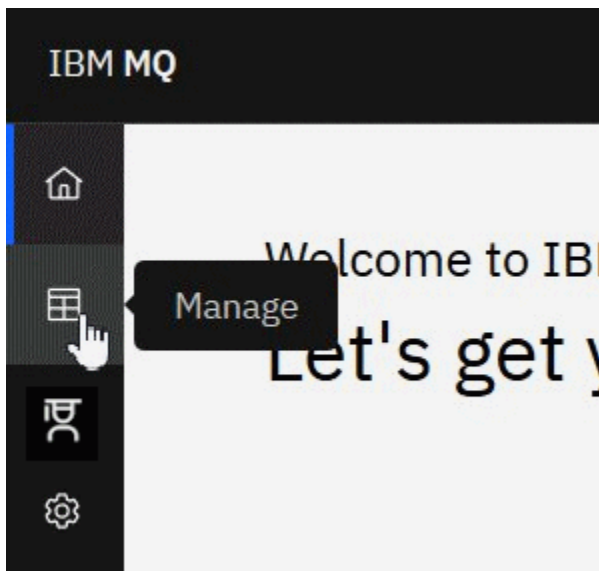
 V 9.4.0

Kurzübersicht über die IBM MQ Console

Nach der ersten Anmeldung bei der IBM MQ Console befinden Sie sich auf der Landing-Page. Von hier aus können Sie vorhandene Warteschlangenmanager verwalten, einen Warteschlangenmanager oder eine Warteschlange erstellen, zu Bildungsthemen navigieren oder die Produktinformationen zu IBM MQ in IBM Documentation öffnen. Sie können auch den Schnelleinstieg für die Anwendung starten, der Sie durch den Prozess der schnellen und einfachen Einrichtung von Messaging zwischen neuen oder vorhandenen Warteschlangenmanagern und Anwendungen führt.



Alternativ können Sie einfach auf das Verwaltungssymbol klicken, um direkt mit der Verwaltung von IBM MQ-Objekten zu beginnen.



In der Verwaltungsansicht werden anfangs Warteschlangenmanager und deren aktueller Status angezeigt. Sie können auch neue Warteschlangenmanager erstellen und ferne Warteschlangenmanager verbinden.

IBM MQ

Manage

Queue managers

Remote queue managers

Local queue managers

Here you can view and manage all of the queue managers co-located with this instance of the MQ console.

4 of 4 queue managers loaded

Unique name ↑	Status
QM931	Running
QMGR1	Running
QMGR2	Running
QMGR3	Running

Items per page: 10 | 1-4 of 4 items | 1 of 1 pages

Jeder Warteschlangenmanager verfügt über ein Menü, in dem Sie einen aktiven Warteschlangenmanager stoppen oder konfigurieren oder einen gestoppten Warteschlangenmanager starten oder löschen können.

● Running

● Stopped

● Stopped

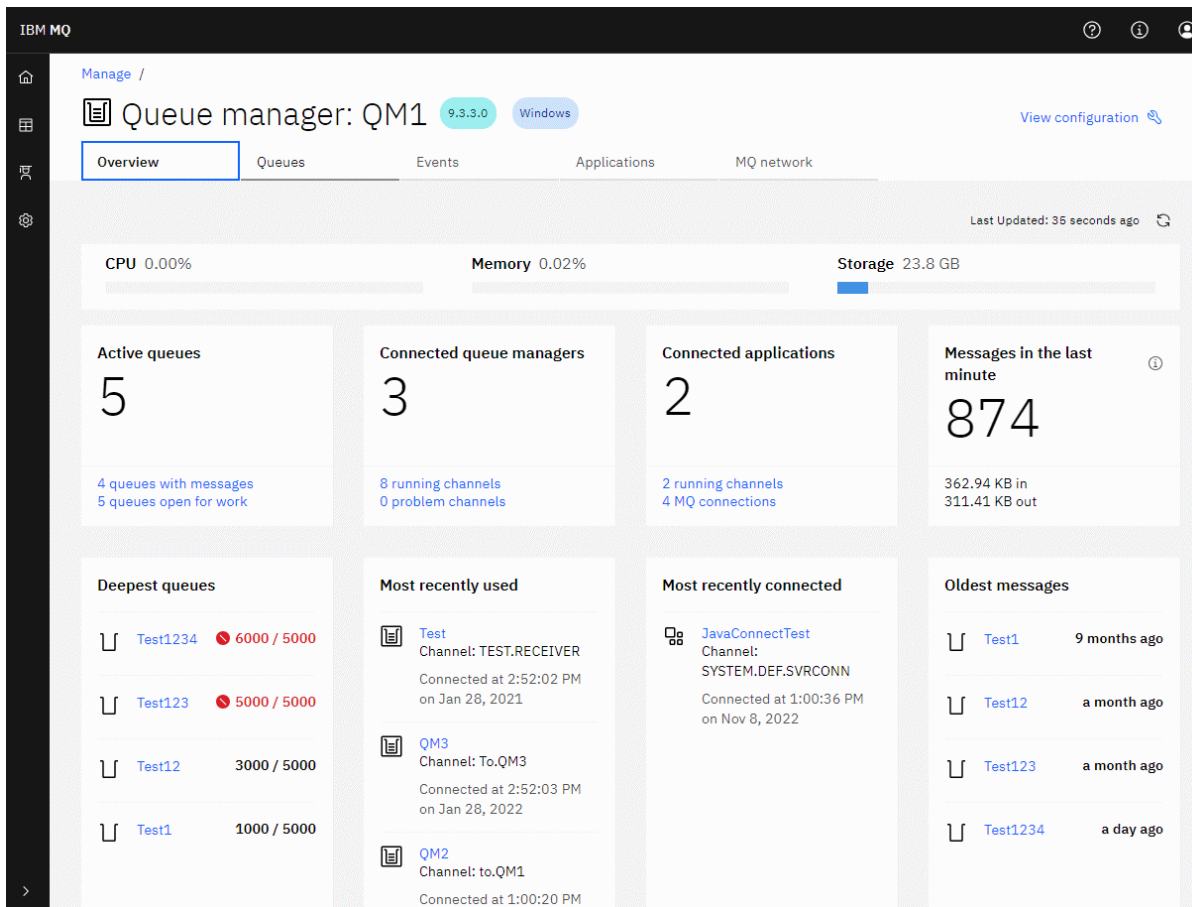
● Stopped

Stop

Configuration

Berechtigungssätze, Authentifizierungsdatenobjekte und Kanalauthentifizierungsdatensätze für den Warteschlangenmanager finden Sie auf der Registerkarte **Sicherheit** auf der Seite **Konfiguration** des Warteschlangenmanagers, auf der Sie Elemente erstellen und neue hinzufügen können.

Klicken Sie auf den Namen eines aktiven Warteschlangenmanagers, um das zugehörige Dashboard zu öffnen.



Im Dashboard des Warteschlangenmanagers können Sie die folgenden Aktionen ausführen:

V 9.4.0 Zeigen Sie auf der Registerkarte **Übersicht** die folgenden Informationen an:

CPU

Geschätzte CPU-Belastung durch den Warteschlangenmanager in Prozent. (Nicht zutreffend unter z/OS)

Hauptspeicher

Geschätzte Hauptspeicherbelegung durch den Warteschlangenmanager in Prozent. (Nicht zutreffend unter z/OS oder Windows.)

Speicher

Geschätzter Prozentsatz des freien Speicherbereichs der Platte, auf der sich der Warteschlangenmanager befindet. (Nicht zutreffend unter z/OS)

Aktive Warteschlangen

Die Anzahl der Warteschlangen, die entweder Nachrichten enthalten oder für die Eingabe oder Ausgabe geöffnet sind.

Verbundene Warteschlangenmanager

Die Anzahl der momentan verbundenen Warteschlangenmanager, die von aktiven Kanälen abgeleitet wurden.

Verbundene Anwendungen

Anzahl der momentan verbundenen Anwendungen

Nachrichten in der letzten Minute

Zeigt eine Zusammenfassung der PUT/GET-Systemthemen an, die den Nachrichtendurchsatz alle 10 Sekunden anzeigen. (Nicht zutreffend unter z/OS)

Subskriptionen

Zeigt die Anzahl der Subskriptionen an. Nur sichtbar unter z/OS und auf anderen Plattformen, auf denen die Überwachung von Systemthemen unterdrückt wird (siehe [setmqweb properties](#))

Tiefste Warteschlangen

Listet Warteschlangen in der Reihenfolge ihrer Länge auf. Zeigt die aktuelle Warteschlangenlänge und die maximale Warteschlangenlänge an

Zuletzt verwendet

Listet die derzeit verbundenen Warteschlangenmanager sortiert nach dem Datum der letzten Nachricht auf

Zuletzt verbunden

Listet die aktuell verbundenen Anwendungen auf, die von aktiven Serververbindungskanälen abgeleitet sind, sortiert nach Kanalstartdatum und -zeit.

Älteste Nachrichten

Listet die Warteschlangen nach Datum und Uhrzeit der ältesten Nachricht sortiert auf.

Die auf der Registerkarte **Übersicht** angezeigten Informationen werden aus der Überwachung von Systemthemen abgeleitet (siehe [In den Systemthemen veröffentlichte Metriken](#)). z/OS unterstützt die Überwachung von Systemthemen nicht und die Überwachung für Konsolanzeigezwecke kann auf anderen Plattformen inaktiviert werden (siehe [setmqweb properties](#)). In diesen Fällen werden auf der Registerkarte **Übersicht** begrenzte Informationen angezeigt. Die Darstellung ähnelt dem folgenden Beispiel:


The screenshot displays the IBM MQ Queue manager: remote-MQ53 Overview page. The page features a navigation menu on the left with icons for home, menu, and settings. The main content area is titled 'Queue manager: remote-MQ53' and includes a 'View configuration' link. The overview section shows four key metrics: Active queues (0), Connected queue managers (0), Connected applications (0), and Subscriptions (0). Below these are four informational cards: 'Deepest queues' (No queues with messages found), 'Most recently used' (No connected queue managers), 'Most recently connected' (No connected applications), and 'Oldest messages' (No queues with messages found). The page also includes a 'Last Updated: a few seconds ago' indicator and a refresh icon.

Auf der Registerkarte **Warteschlangen**:


- Neue Warteschlangen erstellen
- Klicken Sie auf einen Warteschlangennamen, um vorhandene Nachrichten anzuzeigen und neue zu erstellen und um die Warteschlange zu konfigurieren.

Auf der Registerkarte **Ereignisse** :

Themen

- Neue Themen erstellen
- Vorhandene Themen konfigurieren 
- Auf einen Themennamen klicken, um übereinstimmende Subskriptionen anzuzeigen

Subskriptionen

- Neue verwaltete oder nicht verwaltete Subskriptionen erstellen
- Vorhandene Subskriptionen konfigurieren 

V 9.4.0 Auf der Registerkarte **Anwendungen** :

Übersicht

Enthält Kacheln mit Übersichten zu den folgenden Statistiken:

Verbundene Anwendungen

Zeigt die Anzahl der verbundenen Anwendungen an. Stellt Links zu den folgenden Registerkarten bereit:

- **Anwendungsinstanzen**
- **Verbindungen**

Kanalinstanzen ausführen

Zeigt die Anzahl der SVRCONN-Kanalinstanzen und von diesen Links zu den auf der Registerkarte **App-Kanal** definierten oder gestoppten Instanzen an.

Verbindungen

Zeigt die Anzahl der Verbindungen an. Stellt Links zu den folgenden Informationen auf der Registerkarte **Verbindungen** bereit:

- Lokale Verbindungen (Verbindungen ohne Kanalnamen)
- Ferne Verbindungen (Verbindungen mit einem Kanalnamen)

Am häufigsten verwendete Anwendungen

Zeigt eine Liste häufiger Anwendungen an, sortiert nach der Anzahl der Verbindungen, die sie verwenden.

Am häufigsten verwendete Kanäle

Zeigt eine nach der Anzahl der aktiven Instanzen sortierte Liste häufiger Kanäle an.

Älteste Transaktionen

Zeigt eine Liste der ältesten Transaktionen nach Anwendungsname an. Diese Transaktionen haben Verbindungen mit offenen Arbeitseinheiten und sind nach dem Startdatum und der Startzeit der Arbeitseinheit sortiert.

Ferne verbundene Versionen

Zeigt eine Liste allgemeiner verbundener IBM MQ -Versionen an, d. h. die Kanalinstanzen, für die REMOTE_VERSION angegeben wurde.

Sicherheit des Anwendungskanals

Zeigt eine Liste der allgemeinen verbundenen Kanalsicherheitsprotokolle an, d. h. der Kanalinstanzen, für die SECURITY_PROTOCOL angegeben ist.

Kanalübertragungsraten

Zeigt eine Liste allgemeiner Kanäle sortiert nach den Übertragungsraten von Nachrichten und Byte an. Verwendet das Kanalstartdatum und die Kanalstartzeit zur Berechnung der Dauer und verwendet MSGS und MQIACH_BYTES_SENT/ MQIACH_BYTES_RCVD zur Berechnung der Rate.


Anwendungen

Informationen zu Anwendungen anzeigen, die mit dem Warteschlangenmanager verbunden sind.

Kanäle

Aktivität auf Kanälen anzeigen, die mit Anwendungen verbunden sind.

Anwendungskanäle

- Starten, stoppen, mit Pingsignal überprüfen und konfigurieren von Kanälen 
- Erstellen von neuen Kanälen

- Kanäle zurücksetzen

App-Kanalinstanzen

- Status von Anwendungskanalinstanzen anzeigen
- Unbestätigte Nachrichten in Kanälen auflösen ⋮

V 9.4.0 Auf der Registerkarte **MQ -Netz** :

Übersicht

Enthält Kacheln mit Übersichten zu den folgenden Statistiken:

Aktive Instanzen des Warteschlangenmanagerkanals

Zeigt die Anzahl der Nicht-SVRCONN-Kanalinstanzen an. Zeigt Links zu den folgenden Kanalinstanztypen auf der Registerkarte **Warteschlangenmanager verbunden** an:

- Definierte Kanäle
- Gestoppte Kanäle

Verbundene Warteschlangenmanager

Zeigt die Anzahl der Warteschlangenmanager an, die über MQCA_REMOTE_Q_MGR_NAME verbunden sind. Gibt außerdem die Anzahl der Warteschlangenmanager an, die von MQCMD_INQUIRE_CLUSTER_Q_MGR zurückgegeben werden.

Clusterzugehörigkeit

Wenn nur ein Warteschlangenmanagercluster vorhanden ist, wird der Name des Clusters angezeigt und es wird angezeigt, ob der Warteschlangenmanager ein vollständiges Repository oder ein Teilrepository ist. Zeigt an, wie viele Warteschlangenmanager im Cluster sichtbar sind. Wenn mehrere Cluster vorhanden sind, wird die Anzahl der Cluster sowie die Anzahl der vollständigen und Teilrepository-Warteschlangenmanager in jedem Cluster angezeigt.

Fehlgeschlagene Warteschlangenmanagerkanäle

Zeigt eine Liste der Kanäle an, die sich im Wiederholungsstatus befinden (nicht gestoppt/aktiv). Berechnet die Anzahl der verbleibenden Wiederholungen, wenn sie sich im Wiederholungsstatus befinden. Die Liste enthält Kanäle mit den folgenden Statustypen:

- MQCHS_PAUSED
- MQCHS_RETRYING

Längste Nachrichtenverzögerungen

Zeigt eine Liste der Kanäle mit einem XMIT-Zeitindikator (langer Zeitraum) an.

Unbeaufsichtigte Übertragungswarteschlangen

Zeigt eine Liste der Übertragungswarteschlangen mit einer Warteschlangenlänge ungleich null und ohne zugeordnete Kennungen an.

Fernverbindungsversionen

Zeigt eine Liste allgemeiner verbundener IBM MQ -Versionen an, d. h. die Kanalinstanzen, für die REMOTE_VERSION angegeben wurde.

Kanalsicherheit des Warteschlangenmanagers

Zeigt eine Liste der allgemeinen verbundenen Kanalsicherheitsprotokolle an, d. h. der Kanalinstanzen, für die SECURITY_PROTOCOL angegeben ist.

Clusterstatus

Zeigt eine Reihe unabhängiger Statistiken an, die sich auf den Zustand des Clusters beziehen. Der Status umfasst:

- Die Anzahl der Clusterobjekte (Warteschlangen, Themen, Warteschlangenmanager)
- Die Anzahl der ausgesetzten Warteschlangenmanager (MQIACF_SUSPEND ist auf YES gesetzt)
- Die Tiefe von SYSTEM.CLUSTER.COMMAND.QUEUE -Warteschlange.
- Die Anzahl der Clusterwarteschlangenmanagereinträge, die mit SYSTEM.TEMP

Wenn alle diese Werte null sind, wird diese Kachel nicht angezeigt und stattdessen die Kachel **Listener** angezeigt.


Empfangsprogramme

Zeigt eine Liste der Listener an und gibt an, ob sie aktiv sind. Wird nur angezeigt, wenn die Kachel **Clusterzustand** nicht angezeigt wird.


Verbundene Warteschlangenmanager

Details zu den Warteschlangenmanagern anzeigen, die momentan mit diesem Warteschlangenmanager verbunden sind.

WS-Manager-Kanäle


- Starten, stoppen, mit Ping-Signal überprüfen und konfigurieren von Kanälen 
- Erstellen von neuen Kanälen
- Kanäle zurücksetzen

Warteschlangenmanagerkanalinstanzen


- Status von Warteschlangenmanagerkanalinstanzen anzeigen
- Unbestätigte Nachrichten in Kanälen auflösen 


IBM MQ Console: Mit lokalen Warteschlangenmanagern arbeiten

Sie können lokale Warteschlangenmanager aus der höchsten Ebene der Verwaltungsansicht

 erstellen, konfigurieren und steuern.

Informationen zu diesem Vorgang

 In der Verwaltungsansicht werden die lokalen Warteschlangenmanager aufgeführt, 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.

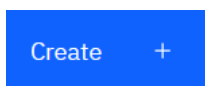
 In z/OS werden in der Ansicht "Verwalten" die Warteschlangenmanager aufgelistet, die sich in derselben Version wie die IBM MQ Console befinden, und auf dem System, auf dem IBM MQ Console ausgeführt wird, definiert. Warteschlangenmanager, deren Version von derjenigen der IBM MQ Console abweicht, werden nicht aufgelistet.

Sie können einzelne Warteschlangenmanager aus der Liste auswählen, um mit ihnen zu arbeiten.






Anmerkung: Der IBM MQ Console kann eine Verbindung zu einem lokalen RDQM-Warteschlangenmanager herstellen, wenn er aktiv ist (d. h. die primäre Rolle hat), bietet jedoch keine RDQM-spezifischen Funktionen.

Prozedur

- So erstellen Sie einen neuen lokalen Warteschlangenmanager:
 - a) Klicken Sie in der Liste mit den Warteschlangenmanagern auf die Schaltfläche 'Erstellen'



- 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) Suchen Sie den Warteschlangenmanager, den Sie starten möchten, in der Liste.
 - b) Wählen Sie den Eintrag **Start** im Menü aus  .
- 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) Wählen Sie den Eintrag **Stop** im Menü aus  .
- So löschen Sie einen lokalen Warteschlangenmanager:
 - a) Falls der Warteschlangenmanager aktiv ist, stoppen Sie ihn.
 - b) Wählen Sie **Konfiguration anzeigen** im Menü  und dann **Warteschlangenmanager löschen** aus.
 - c) Bestätigen Sie, dass Sie den Warteschlangenmanager löschen möchten, indem Sie den zugehörigen Namen im Bestätigungsfenster eingeben. 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 suchen Sie ihn in der Warteschlangenmanagerliste.
 - b) Wählen Sie im Menü **Konfiguration anzeigen**  aus.
 - c) Stellen Sie sicher, dass die Registerkarte **Eigenschaften** ausgewählt ist. 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. Um Informationen zu einer Eigenschaft zu erhalten, können Sie die Eigenschaftsinformationen in den Eigenschaften des Warteschlangenmanagers anzeigen.
- So arbeiten Sie mit Sicherheitseinstellungen für den lokalen Warteschlangenmanager:
 - a) Stellen Sie sicher, dass der Warteschlangenmanager aktiv ist, und wählen Sie ihn in der Warteschlangenmanagerliste aus.
 - b) Wählen Sie im Menü **Konfiguration anzeigen**  aus.
 - c) Stellen Sie sicher, dass die Registerkarte **Sicherheit** ausgewählt ist.
 - d) Sie können mit Authentifizierungsobjekten, Berechtigungssätzen oder Kanalauthentifizierungsobjekten arbeiten. Weitere Informationen finden Sie in den folgenden Abschnitten:
 - „IBM MQ Console: Mit Authentifizierungsinformationsobjekten arbeiten“ auf Seite 103
 - „IBM MQ Console: Mit Berechtigungsdatensätzen des Warteschlangenmanagers arbeiten“ auf Seite 105
 - „IBM MQ Console: Mit Kanalauthentifizierungsdatensätzen arbeiten“ auf Seite 106

V 9.4.0 IBM MQ Console: Mit Authentifizierungsinformationsobjekten arbeiten

Sie können die Konsole verwenden, um Authentifizierungsdatenobjekte auf einem Warteschlangenmanager hinzuzufügen und zu löschen. Außerdem können die Eigenschaften angezeigt und festgelegt und die Berechtigungssätze für die Objekte verwaltet werden.


Informationen zu diesem Vorgang

Die Ansicht Authentifizierungsdaten listet die Authentifizierungsdaten auf, die für einen bestimmten Warteschlangenmanager verfügbar sind. Sie können einzelne Authentifizierungsinformationen aus der Liste auswählen, um mit ihnen zu arbeiten.

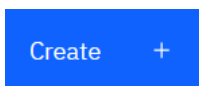
Die Warteschlangenmanager-Authentifizierungsinformationen sind Bestandteil der IBM MQ-Unterstützung für Transport Layer Security (TLS). Die Objekte enthalten die Definitionen, die zur Durchführung der Überprüfung des Zertifikatswiderrufs mithilfe von OCSP oder Zertifikatswiderrufslisten (CRLs) auf LDAP-Servern erforderlich sind, sowie die zur Aktivierung der Benutzer-ID- und Kennwortprüfung erforderlichen Definitionen.

Prozedur

- So zeigen Sie die Authentifizierungsdaten für einen Warteschlangenmanager an:
 - a) Stellen Sie sicher, dass der Warteschlangenmanager aktiv ist, und wählen Sie ihn in der Warteschlangenmanagerliste aus.

- b) Wählen Sie im Menü **Konfiguration anzeigen**  aus.
- c) Stellen Sie sicher, dass die Registerkarte **Sicherheit** ausgewählt ist.
- d) Wählen Sie im Navigationsbereich **Authentifizierungsdaten** aus.

- So fügen Sie ein Authentifizierungsinformationsobjekt hinzu:





- a) Klicken Sie auf die Schaltfläche 'Erstellen' in der Listenansicht der Authentifizierungsinformationen.
- b) Geben Sie den Namen des Authentifizierungsinformationsobjekts an. Gültige Zeichen sind Buchstaben und Zahlen sowie die Zeichen ".", "/", "_" und "%".
- c) Geben Sie den Typ des Authentifizierungsinformationsobjekts an.
- d) Geben Sie zusätzliche objekttypspezifische Informationen an:

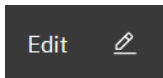
- Geben Sie für **CRL LDAP** den **LDAP-Servernamen** an. Dies ist der Hostname (eine IPv4-Adresse in der Schreibweise mit Trennzeichen oder eine IPv6-Adresse in Hexadezimalschreibweise) des Hosts, auf dem der LDAP-Server aktiv ist. Optional kann zusätzlich eine Portnummer angegeben werden. Sie können optional einen Benutzernamen und ein Kennwort für den Benutzer angeben, der auf den LDAP-Server zugreift.
- 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** gibt es keine zusätzlichen Anforderungen, obwohl Sie potentiell weitere Optionen für diesen Authentifizierungstyp angeben können.
- 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. Falls gewünscht können Sie weitere Optionen für diesen Authentifizierungstyp angeben.

- e) Klicken Sie auf **Hinzufügen**.



- So löschen Sie ein Authentifizierungsinformationsobjekt:

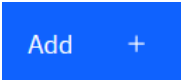
- a) Wählen Sie das Schraubenschlüsselsymbol  für das Authentifizierungsdatenobjekt aus, das Sie aus der Liste löschen möchten.
 - b) Klicken Sie in der Ansicht Objekteigenschaften auf **Authentifizierungsdatenobjekt** löschen.
 - c) Bestätigen Sie durch Klicken auf **Löschen**, dass das Authentifizierungsinformationsobjekt gelöscht werden soll. Das Objekt wird gelöscht.
- So können Sie die Eigenschaften eines Authentifizierungsinformationsobjekts anzeigen und bearbeiten:

- a) Wählen Sie das Schraubenschlüsselsymbol  für das Authentifizierungsdatenobjekt aus, das aus der Liste angezeigt werden soll.
 - b) Um die angezeigten Eigenschaften zu bearbeiten, klicken Sie auf die Schaltfläche 'Bearbeiten'



- c) Bearbeiten Sie die Eigenschaften wie gewünscht. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden.
 - d) Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.
- So können Sie Berechtigungssätze für ein Authentifizierungsinformationsobjekt anzeigen und bearbeiten:

- a) Wählen Sie das Schraubenschlüsselsymbol  für das Authentifizierungsdatenobjekt aus, dessen Berechtigungssatz angezeigt werden soll.
 - b) Wählen Sie die Registerkarte **Sicherheit** aus.
 - c) Um einen vorhandenen Berechtigungssatz zu bearbeiten oder zu löschen, wählen Sie **Bearbeiten** oder **Löschen** aus dem Menü .
 - d) Wenn Sie einen neuen Berechtigungsdatensatz hinzufügen möchten, klicken Sie auf die **Hinzufügen**-Schaltfläche

, geben Sie die Details des neuen Berechtigungsdatensatzes an und klicken Sie auf **Erstellen**.

V 9.4.0 IBM MQ Console: Mit Berechtigungsdatensätzen des Warteschlangenmanagers arbeiten


Sie können den Zugriff von Benutzern und Gruppen auf Warteschlangenmanager steuern, indem Sie einen Berechtigungssatz für jeden Benutzer oder jede Gruppe angeben.

Informationen zu diesem Vorgang

Sie können den Zugriff, den ein Messaging-Benutzer oder eine Gruppe von Messaging-Benutzern auf einen bestimmten Warteschlangenmanager hat, mithilfe von Berechtigungssätzen optimieren. Es gibt zwei Typen von Berechtigungsdatensätzen: die **Warteschlangenmanagerzugriffsdatsätze**, die allgemeine Berechtigungen steuern, und die **Erstellungsberechtigung**, die steuern, welche Benutzer und Gruppen Objekte für den Warteschlangenmanager erstellen können.


Prozedur

- So zeigen Sie die Berechtigungssätze für einen Warteschlangenmanager an:
 - a) Stellen Sie sicher, dass der Warteschlangenmanager aktiv ist, und wählen Sie ihn in der Warteschlangenmanagerliste aus.

- b) Wählen Sie im Menü **Konfiguration anzeigen**  aus.
 - c) Stellen Sie sicher, dass die Registerkarte **Sicherheit** ausgewählt ist.
 - d) Wählen Sie im Navigationsfenster **Berechtigungssätze** aus. In der Anzeige werden die Berechtigungssätze in zwei Fenstern angezeigt, damit Sie mit den allgemeine Berechtigungssätzen arbeiten und Berechtigungssätze erstellen können.
- So fügen Sie einen allgemeinen Berechtigungssatz hinzu:

a) Klicken Sie auf die Schaltfläche 'Hinzufügen'  in der Listenansicht **Warteschlangenmanagerzugriff** .

- b) Wählen Sie aus, ob Sie einen Berechtigungssatz für einen Benutzer oder eine Gruppe hinzufügen möchten.
 - c) Den Namen des Benutzers oder der Gruppe angeben, für den bzw. die ein Berechtigungssatz hinzugefügt wird (der Berechtigungssatz nimmt diesen Namen an).
 - d) Wählen Sie die Berechtigungen aus, die Sie erteilen wollen.
 - e) Klicken Sie auf **Erstellen**.
- So fügen Sie einen Berechtigungssatz zum Erstellen hinzu:

a) Klicken Sie auf die Schaltfläche 'Hinzufügen'  in der Listenansicht **Berechtigung zum Erstellen** .

- b) Wählen Sie aus, ob Sie einen Berechtigungssatz für einen Benutzer oder eine Gruppe hinzufügen möchten.
 - c) Den Namen des Benutzers oder der Gruppe angeben, für den bzw. die ein Berechtigungssatz hinzugefügt wird (der Berechtigungssatz nimmt diesen Namen an).
 - d) Wählen Sie die Typen der Objekte aus, denen die Berechtigung zum Erstellen erteilt werden soll.
 - e) Klicken Sie auf **Erstellen**.
- So löschen Sie einen Berechtigungssatz:
 - a) Wählen Sie den zu löschenden Berechtigungsdatensatz aus und wählen Sie **Löschen** aus.
 - b) Bestätigen Sie durch Klicken auf **Löschen**, dass das Authentifizierungsinformationsobjekt gelöscht werden soll. Das Objekt wird gelöscht.
- So zeigen Sie die Eigenschaften eines Berechtigungssatzes an und bearbeiten diese:
 - a) Klicken Sie auf den Berechtigungsdatensatz, den Sie anzeigen wollen.
 - b) Ändern Sie die Einstellungen wie gewünscht und klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.

IBM MQ Console: Mit Kanalauthentifizierungsdatensätzen arbeiten

Mit IBM MQ Console können Sie Kanalauthentifizierungssätze auf einem Warteschlangenmanager hinzufügen und löschen. Außerdem können die Eigenschaften für Kanalauthentifizierungsdatensätze angezeigt und festgelegt werden.


Informationen zu diesem Vorgang

Die Zugriffsberechtigungen zum Herstellen von Systemverbindungen auf Kanalebene können mithilfe von Kanalauthentifizierungsdatensätzen gezielter gesteuert werden.

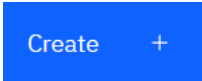
Um Sicherheit zu erzwingen, können Sie mithilfe von Blockierungskanalauthentifizierungsdatensätzen den Zugriff auf Ihre Kanäle sperren. Sie können auch mithilfe von Adresszuordnungskanalauthentifizierungsdatensätzen bestimmten Benutzern den Zugriff ermöglichen. Weitere Informationen zu Kanalauthentifizierungsdatensätzen finden Sie unter [Kanalauthentifizierungsdatensätze](#).

Prozedur


- So zeigen Sie die Kanalauthentifizierungsdaten für einen Warteschlangenmanager an:
 - a) Stellen Sie sicher, dass der Warteschlangenmanager aktiv ist, und wählen Sie ihn in der Warteschlangenmanagerliste aus.

- b) Wählen Sie im Menü **Konfiguration anzeigen**  aus.
- c) Stellen Sie sicher, dass die Registerkarte **Sicherheit** ausgewählt ist.
- d) Wählen Sie im Navigationsbereich **Kanalauthentifizierung** aus.


- So fügen Sie einen Kanalauthentifizierungsdatensatz hinzu:

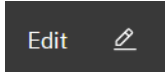
- a) Klicken Sie auf die Schaltfläche 'Erstellen'  in der Listenansicht 'Kanalauthentifizierungsdaten'.
Weitere Informationen zu den verfügbaren Einstellungen für die Kanalauthentifizierungsdatensätze finden Sie unter [Kanalauthentifizierungsdatensätze](#) und [SET CHLAUTH](#)
- b) Wählen Sie den Regeltyp, den Sie verwenden möchten. Wählen Sie **Erlauben**, **Blockieren** oder **Warnen** aus.
- c) Wählen Sie den Identitätstyp, für den Sie eine Kanalauthentifizierungsregel konfigurieren möchten. Verschiedene Identitätstypen sind je nach ausgewähltem Regeltyp verfügbar.
- d) Geben Sie die erforderlichen Informationen für die von Ihnen angegebene Identität an. Standardmäßig werden die empfohlenen Mindesteigenschaften angezeigt, für die Sie Werte bereitstellen sollten. Sie können alle verfügbaren Eigenschaften anzeigen, indem Sie **Angepasste Erstellung** auswählen.
- e) Klicken Sie auf **Create (Erstellen)**, um den Kanalauthentifizierungsdatensatz zu erstellen.

- So löschen Sie einen Kanalauthentifizierungsdatensatz:

- a) Klicken Sie auf das Schraubenschlüsselsymbol  neben dem Kanalauthentifizierungsdatensatz, den Sie löschen möchten.
- b) Klicken Sie in der Ansicht Kanalauthentifizierung bearbeiten auf **Kanalauthentifizierungsobjekt löschen**.
- 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) Klicken Sie neben dem Kanalauthentifizierungsdatensatz auf das Schraubenschlüsselsymbol , den Sie bearbeiten oder anzeigen möchten. Die Eigenschaften werden angezeigt.

- b) Klicken Sie auf die Schaltfläche Bearbeiten .
- c) Bearbeiten Sie die Eigenschaften wie gewünscht. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden.
- d) Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.


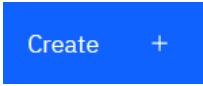





Multi V 9.4.0 **IBM MQ Console: Mit Listnern arbeiten**


Mit der IBM MQ Console können Sie Listener hinzufügen und löschen, starten und stoppen, die Listenereigenschaften anzeigen und festlegen und die Berechtigungssätze für Listener verwalten.

Informationen zu diesem Vorgang

In der Ansicht mit den Listnern werden die Listener angezeigt, die für einen bestimmten Warteschlangenmanager vorhanden sind. Sie können einzelne Listener auswählen, mit denen Sie arbeiten möchten.

Prozedur

- Gehen Sie wie folgt vor, um die Listener eines Warteschlangenmanagers anzuzeigen:
 - a) Stellen Sie sicher, dass der Warteschlangenmanager aktiv ist, und wählen Sie ihn in der Warteschlangenmanagerliste aus.
 - b) Wählen Sie im Menü **Konfiguration anzeigen**  aus.
 - c) Wählen Sie die Registerkarte **Listener** aus.
- So erstellen Sie einen Listener:
 - a) Klicken Sie auf die Schaltfläche Erstellen .
 - b) Geben Sie die erforderlichen Informationen für den Listener an, den Sie erstellen.
 - c) Klicken Sie auf **Erstellen**. Das neue Empfangsprogramm wird erstellt.
- So starten Sie ein Empfangsprogramm:
 - a) Suchen Sie den Listener, den Sie starten möchten, in der Liste.
 - b) Wählen Sie den Eintrag **Start** im Menü aus .
- So stoppen Sie ein Empfangsprogramm:
 - a) Suchen Sie den Listener, den Sie starten möchten, in der Liste.
 - b) Wählen Sie den Eintrag **Stop** im Menü aus .
- So können Sie die Eigenschaften eines Empfangsprogramms anzeigen und bearbeiten:
 - a) Suchen Sie den Listener in der Liste.
 - b) Wählen Sie im Menü **Konfiguration anzeigen**  aus.
 - c) Stellen Sie sicher, dass die Registerkarte **Eigenschaften** ausgewählt ist. Klicken Sie zum Bearbeiten der Eigenschaften auf die Schaltfläche 'Bearbeiten' .
 - d) Bearbeiten Sie die Eigenschaften wie gewünscht. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Weitere Informationen zu den Eigenschaften finden Sie im Abschnitt [Eigenschaften des Empfangsprogramms](#) in der MQ Explorer-Dokumentation.
 - e) Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.
- So können Sie Berechtigungssätze für ein Empfangsprogramm anzeigen und bearbeiten:
 - a) Suchen Sie den Listener in der Liste.
 - b) Wählen Sie im Menü **Konfiguration anzeigen**  aus.
 - c) Klicken Sie auf die Registerkarte **Sicherheit**.
 - d) Bearbeiten Sie die Berechtigungssätze, wie dies für Berechtigungssätze für Warteschlangenmanager beschrieben ist. Weitere Informationen finden Sie unter „IBM MQ Console: Mit Berechtigungsdatensätzen des Warteschlangenmanagers arbeiten“ auf Seite 105.

- So löschen Sie ein Empfangsprogramm:
 - a) Suchen Sie den Listener in der Liste.
 - b) Wählen Sie im Menü **Konfiguration anzeigen**  aus.
 - c) Klicken Sie auf **Listener löschen**.

V 9.4.0 IBM MQ Console: Fernen Warteschlangenmanager hinzufügen

Mit dem IBM MQ Console können Sie einen Warteschlangenmanager verwalten, der auf einem fernen System läuft.

Vorbereitende Schritte

- Sie müssen den Warteschlangenmanager auf dem fernen System so vorbereiten, dass er über Fernzugriff verwaltet werden kann (siehe Schritt „1“ auf Seite 111, „2“ auf Seite 111, „3“ auf Seite 112 und „4“ auf Seite 112 von „[Fernen Warteschlangenmanager über die Befehlszeile zur IBM MQ Console hinzufügen](#)“ auf Seite 111).
- Außerdem müssen Sie Fernverbindungen über die IBM MQ Console aktivieren. Weitere Informationen finden Sie im Abschnitt [Verbindungsverhalten des fernen Warteschlangenmanagers konfigurieren](#).

Informationen zu diesem Vorgang

Sie verwenden eine Clientkanal-Definitionstabelle (CCDT) im JSON-Format, um die Details der Remote-Verbindung anzugeben. Sie können ein JSON-CCDT mit einem Texteditor (siehe Schritt „5“ auf Seite 112 von „[Fernen Warteschlangenmanager über die Befehlszeile zur IBM MQ Console hinzufügen](#)“ auf Seite 111) oder mithilfe des IBM MQ Console erstellen.

Alternativ können Sie die CCDT durch den IBM MQ Console erstellen, indem Sie die Verbindungsdetails direkt beim Hinzufügen des fernen Warteschlangenmanagers eingeben.

Sie können auch einen fernen Warteschlangenmanager mit dem IBM MQ Console verbinden, indem Sie die Befehlszeile für alle erforderlichen Aufgaben verwenden (zusätzlich zur Vorbereitung des entfernten Warteschlangenmanagers und der Erstellung einer CCDT). Siehe „[Fernen Warteschlangenmanager über die Befehlszeile zur IBM MQ Console hinzufügen](#)“ auf Seite 111.



Achtung: wenn Sie die folgenden Nachrichten empfangen:

```
MQWB2026E: The request to connect to the remote queue manager 'rqmgr-qmgr_name' failed
with the error message:
'JMSSC0051: The property 'JMS_IBM_MQMD_AccountingToken' should be set using type '[B',
not 'java.lang.Object'.'
```

Sie versuchen, eine `java.lang.Object` an das Abrechnungstoken zu übergeben, obwohl ein Java-Objekttyp `byte []` erwartet wird.

Prozedur

- So fügen Sie einen fernen Warteschlangenmanager hinzu, indem Sie eine vorhandene CCDT angeben:
 - a) Klicken Sie auf der Startseite auf **Warteschlangenmanager verbinden**.
 - b) Geben Sie den Namen des fernen Warteschlangenmanagers an.
 - c) Geben Sie optional einen eindeutigen Namen für den Warteschlangenmanager ein. Falls Sie keinen eindeutigen Namen angeben, wird der tatsächliche Name mit dem Prefix „remote-“ verwendet.
 - d) Stellen Sie sicher, dass **Mithilfe einer JSON CCDT verbinden** ausgewählt ist.
 - e) Klicken Sie auf **Durchsuchen** und wählen Sie die Datei, die die JSON CCDT enthält, die Sie verwenden möchten.
 - f) Klicken Sie auf **Weiter**, um zur Benutzerseite zu navigieren und optional einen Benutzernamen sowie Kennwort einzugeben, um sich mit dem fernen Warteschlangenmanager zu verbinden. Falls

Sie diese Angaben weglassen, werden die Authentifizierungsdaten von der Konfigurationsdatei der fernen Verbindung bezogen.

- g) Klicken Sie auf **Weiter**, um zur Zertifikatsseite zu navigieren. Wenn Ihr CCDT „transmissionSecurity“-Informationen angibt, werden diese Informationen verwendet. Sie können optional ein Zertifikat (als Base64-codierten öffentlichen Schlüssel) einfügen, das dem globalen Truststore hinzugefügt wird.

Das Zertifikat wird vorübergehend in `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` gespeichert, bevor es dem Truststore hinzugefügt wird. Wenn die Verbindung erfolgreich hinzugefügt wird, wird das Zertifikat von diesem Speicherort gelöscht.
- h) Klicken Sie auf **Weiter**, um die Zusammenfassungsseite anzuzeigen. Über die Schaltfläche **Zurück** können Sie die vorherigen Seiten erneut aufrufen und Korrekturen vornehmen. Wenn Sie mit den Informationen einverstanden sind, klicken Sie auf **Verbinden**, um eine Verbindung zum fernen Warteschlangenmanager herzustellen.
- So fügen Sie einen fernen Warteschlangenmanager hinzu und geben die Verbindungsinformationen manuell an:
 - a) Klicken Sie auf der Startseite auf **Warteschlangenmanager verbinden**.
 - b) Geben Sie den Namen des fernen Warteschlangenmanagers an.
 - c) Geben Sie optional einen eindeutigen Namen für den Warteschlangenmanager ein. Falls Sie keinen eindeutigen Namen angeben, wird der tatsächliche Name mit dem Prefix „remote-“ verwendet.
 - d) Wählen Sie **Manuelle Eingabe**.
 - e) Geben Sie den Namen des Clientverbindungskanal ein, den die Verbindung verwendet.
 - f) Geben Sie den Namen des Hosts an, auf dem der ferne Warteschlangenmanager ausgeführt wird. Wenn ferne MQ-Installationen erkannt werden, werden die Hostnamen angezeigt und Sie können den Host des fernen Warteschlangenmanagers auswählen, zu dem Sie eine Verbindung herstellen möchten. In einigen Netzwerkkonfigurationen ist es nicht möglich, ferne MQ-Instanzen zu erkennen. Fügen Sie in diesem Fall den Hostnamen und den Port manuell hinzu.
 - g) Klicken Sie auf **Weiter**, um zur Benutzerseite zu navigieren und optional einen Benutzernamen sowie Kennwort einzugeben, um sich mit dem fernen Warteschlangenmanager zu verbinden. Falls Sie diese Angaben weglassen, werden die Authentifizierungsdaten von der Konfigurationsdatei der fernen Verbindung bezogen.
 - h) Klicken Sie auf **Weiter**, um zur Zertifikatsseite zu navigieren. Sie können eine SSL-CipherSpec aus der Dropdown-Liste auswählen. Sie können optional ein Zertifikat (als Base64-codierten öffentlichen Schlüssel) einfügen, das dem globalen Truststore hinzugefügt wird.

Das Zertifikat wird vorübergehend in `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` gespeichert, bevor es dem Truststore hinzugefügt wird. Wenn die Verbindung erfolgreich hinzugefügt wird, wird das Zertifikat von diesem Speicherort gelöscht.
 - i) Klicken Sie auf **Weiter**, um die Zusammenfassungsseite anzuzeigen. Über die Schaltfläche **Zurück** können Sie die vorherigen Seiten erneut aufrufen und Korrekturen vornehmen. Wenn Sie mit den Informationen einverstanden sind, klicken Sie auf **Verbinden**, um eine Verbindung zum fernen Warteschlangenmanager herzustellen.

Die von Ihnen angegebenen Verbindungsinformationen werden in die CCDT-Datei in Ihrem Webverzeichnis geschrieben. Der Pfad ist `WLP_USER_DIR/generated.ccdt/ccdt-uniqueName`.

Ergebnisse

Der ferne Warteschlangenmanager erscheint in der Liste der fernen Warteschlangenmanager im IBM MQ Console. Wenn die Verbindung erfolgreich ist, können Sie die Objekte des entfernten Warteschlangenmanagers genauso verwalten, wie die Objekte eines lokalen Warteschlangenmanagers.

V 9.4.0 Fernen Warteschlangenmanager über die Befehlszeile zur IBM MQ Console hinzufügen

Sie können einen fernen Warteschlangenmanager zur IBM MQ Console hinzufügen, indem Sie den Befehl **setmqweb remote** in der Befehlszeile verwenden. Ein ferner Warteschlangenmanager kann entweder ein Warteschlangenmanager sein, der in einer anderen Installation auf demselben System wie der IBM MQ Console läuft, oder ein Warteschlangenmanager, der auf einem anderen System läuft.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen:

- **ALW** An AIX, Linux, and Windows geben Sie MQSC-Befehle aus einer **runmqsc** Eingabeaufforderung. Sehen [Interaktives Ausführen von MQSC-Befehlen unter runmqsc](#) und [Ausführen von MQSC-Befehlen aus Textdateien unter runmqsc](#). Öffnen Sie für diese Task unter AIX, Linux, and Windows eine **runmqsc**-Eingabeaufforderung, die QM1:

```
runmqsc QM1
```

- **IBM i** An IBM i erstellen Sie eine Liste von Befehlen in einer Skriptdatei und führen die Datei dann mit dem **STRMQMQSC** Befehl. Sehen [Administration mit MQSC-Befehlen auf IBM i](#).
- **z/OS** An z/OS, MQSC-Befehle können je nach Befehl aus einer Reihe von Quellen ausgegeben werden. Sehen [Quellen, aus denen Sie MQSC- und PCF-Befehle ausgeben können auf IBM MQ for z/OS](#).

Stellen Sie sicher, dass der mqweb-Server so konfiguriert ist, dass Verbindungen des fernen Warteschlangenmanagers zum IBM MQ Console erlaubt werden können. Weitere Informationen finden Sie unter [Verhalten des fernen Warteschlangenmanagers für Verbindungen konfigurieren](#).

Vorgehensweise

1. Erstellen Sie auf dem fernen Warteschlangenmanager einen Serververbindungskanal, um die Fernverwaltung des Warteschlangenmanagers mit dem MQSC-Befehl **DEFINE CHANNEL** zu ermöglichen. Um beispielsweise einen Serververbindungskanal QM1.SVRCONN für Warteschlangenmanager QM1 zu erstellen, geben Sie den folgenden MQSC-Befehl ein:

```
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
```

Weitere Informationen über **DEFINE CHANNEL** und die verfügbaren Optionen finden Sie unter [DEFINE CHANNEL](#).

2. Stellen Sie sicher, dass ein entsprechender Benutzer berechtigt ist, den Warteschlangenmanager und MQ -Objekte zu verwalten, die dem Warteschlangenmanager zugeordnet sind.
- **ALW** Verwenden Sie unter AIX, Linux, and Windows den Steuerbefehl **setmqaut** in einer Standardbefehlszeile.
 - **z/OS** Definieren Sie unter z/OS RACF -Profile, um dem berechtigten Benutzer Zugriff auf den Warteschlangenmanager zu erteilen.

Geben Sie beispielsweise unter AIX, Linux, and Windows den folgenden Steuerbefehl ein, um den Benutzer `exampleUser` für den Zugriff auf den Warteschlangenmanager QM1 zu berechtigen:

```
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

Dieser berechtigte Benutzer kann einer der folgenden Benutzer sein:

- Eine Benutzer-ID, die mit der Benutzer-ID identisch ist, die den mqweb-Server startet, auf dem IBM MQ Console auf dem System ausgeführt wird, von dem aus Sie diesen Warteschlangenmanager über Fernzugriff verwalten möchten.
- Eine Benutzer-ID, die mit einer Benutzer-ID und einem Kennwort übereinstimmt, die dann im Befehl **setmqweb remote** in Schritt „7“ auf Seite 113 enthalten ist. Wenn Sie die Benutzer-ID und

das Kennwort in den Befehl **setmqweb remote** aufnehmen, werden diese Benutzer-ID und das Kennwort für die Authentifizierung verwendet, wenn der IBM MQ Console eine Verbindung zum Warteschlangenmanager herstellt.


- Eine Benutzer-ID, die durch Kanalsicherheitsregeln bestimmt wird. Sie können beispielsweise eine Kanalauthentifizierungsregel im Serververbindungskanal festlegen, um Verbindungen von der IP-Adresse zuzulassen, von der aus Sie IBM MQ Console für die Fernverwaltung verwenden, und alle diese Verbindungen einer bestimmten Benutzer-ID zuordnen, die berechtigt ist, den Warteschlangenmanager zu verwenden. Weitere Informationen finden Sie unter [Neue CHLAUTH-Regeln für Kanäle erstellen](#).

3. ALW


Wenn auf dem fernen Warteschlangenmanager kein Listener ausgeführt wird, erstellen Sie mit dem MQSC-Befehl **DEFINE LISTENER** einen Listener, der eingehende Netzverbindungen akzeptiert. Um beispielsweise ein Empfangsprogramm REMOTE.LISTENER an Port 1414 für den fernen Warteschlangenmanager QM1 zu erstellen, geben Sie den folgenden MQSC-Befehl ein:

```
runmqsc QM1
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
end
```

4. Stellen Sie sicher, dass der Listener mit dem MQSC-Befehl **START LISTENER** ausgeführt wird.

 ALW Geben Sie beispielsweise unter AIX, Linux, and Windows den folgenden MQSC-Befehl ein, um den Listener REMOTE.LISTENER für Warteschlangenmanager QM1 zu starten:

```
runmqsc QM1
START LISTENER(REMOTE.LISTENER)
end
```

 z/OS Geben Sie beispielsweise unter z/OS den folgenden MQSC-Befehl ein, um den Listener zu starten:

```
/cpcf START LISTENER TRPTYPE(TCP) PORT(1414)
```

Beachten Sie, dass der Adressraum des Kanalinitiators gestartet werden muss, bevor Sie einen Listener auf z/OS starten können.

5. Erstellen Sie eine JSON-CCDT-Datei, die die Verbindungsinformationen des entfernten Warteschlangenmanagers enthält:

- Generieren Sie eine CCDT-Datei, indem Sie die IBM MQ Console verwenden, die derselben Installation wie der Warteschlangenmanager zugeordnet ist, zu dem Sie über Fernzugriff eine Verbindung herstellen wollen.

Klicken Sie im Bereich **Home** auf die Kachel **Verbindungsdatei herunterladen**.

- Erstellen Sie eine CCDT-Datei im JSON-Format, die die Verbindung definiert. Weitere Informationen über das Erstellen einer CCDT-Datei im JSON-Format finden Sie unter [Konfiguration einer CCDT im JSON-Format](#).

Die CCDT-Datei muss die Informationen zu name, clientConnection und type enthalten. Optional können Sie zusätzliche Informationen wie z. B. zu transmissionSecurity hinzufügen. Weitere Informationen über alle CCDT-Kanalattributdefinitionen finden Sie unter [Vollständige Liste der CCDT-Kanalattributdefinitionen](#).

Das folgende Beispiel zeigt eine einfache JSON-CCDT-Datei für die Verbindung eines fernen Warteschlangenmanagers. Er setzt den Namen des Kanals auf denselben Namen wie der in Schritt „1“ auf Seite 111 erstellte Beispielserververbindungskanal und den Verbindungsport mit demselben Wert wie der Port, der von der Empfangsfunktion verwendet wird. Der Verbindungshost wird auf den Hostnamen des Systems gesetzt, auf dem der ferne Warteschlangenmanager, QM1, ausgeführt wird:

```
{
  "channel": [{
```



```

    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }
]
}

```

6. Kopieren Sie die JSON-CCDT-Datei auf das System, auf dem IBM MQ Console ausgeführt wird.
7. Verwenden Sie in der Installation, in der IBM MQ Console ausgeführt wird, den Befehl **setmqweb remote**, um die Informationen zum fernen Warteschlangenmanager zur IBM MQ Console -Konfiguration hinzuzufügen.

Um einen fernen Warteschlangenmanager zum IBM MQ Console hinzuzufügen, müssen Sie als Minimum den Namen des Warteschlangenmanagers, einen eindeutigen Namen für den Warteschlangenmanager (zur Unterscheidung zwischen anderen fernen Warteschlangenmanagern, die möglicherweise denselben Warteschlangenmanagernamen aufweisen) und die CCDT-URL für den Warteschlangenmanager angeben. Der eindeutige Name ist der Anzeigename in der IBM MQ Console. Geben Sie daher einen Namen an, der deutlich macht, dass es sich um einen fernen Warteschlangenmanager handelt, z. B. "remote-QM2". Es gibt mehrere zusätzliche Optionen, die Sie angeben können, z. B. den Benutzernamen und das Kennwort, die für die Verbindung mit dem fernen Warteschlangenmanager verwendet werden sollen, oder Details zum Truststore und Keystore. Eine vollständige Liste der Parameter, die mit dem Befehl **setmqweb remote** angegeben werden können, enthält der Artikel [setmqweb remote](#).

Geben Sie beispielsweise den folgenden Befehl ein, um den fernen Warteschlangenmanager QM1 unter Verwendung der CCDT-Beispieldatei hinzuzufügen:

```
setmqweb remote add -uniqueName "MACHINEAQM1" -qmgrName "QM1" -ccdtURL "c:\myccdt\ccdt.json"
```

Ergebnisse

Der ferne Warteschlangenmanager wird in der Liste ferner Warteschlangenmanager in der IBM MQ Console angezeigt, wenn die Liste ferner Verbindungen das nächste Mal aktualisiert wird. Wenn die Verbindung erfolgreich ist, können Sie die Objekte des entfernten Warteschlangenmanagers genauso verwalten, wie die Objekte eines lokalen Warteschlangenmanagers.

Beispiel

Im folgenden Beispiel wird die ferne Warteschlangenmanagerverbindung für einen Warteschlangenmanager QM1 eingerichtet. Der IBM MQ Console ist berechtigt, den Warteschlangenmanager auf der Basis der Berechtigung zu verwalten, die dem Benutzer `exampleUser` erteilt wird. Die Berechtigungsnachweise dieses Benutzers werden an den IBM MQ Console übergeben, wenn der Befehl **setmqweb remote** verwendet wird, um die Verbindungsinformationen des fernen Warteschlangenmanagers zu konfigurieren.

1. Auf dem System, auf dem sich der ferne Warteschlangenmanager QM1 befindet, werden ein Serververbindungskanal und ein Listener erstellt. Der Listener wird gestartet und der Benutzer `exampleUser` erhält die Berechtigung, den Warteschlangenmanager zu verwalten. Führen Sie unter AIX, Linux, and Windows beispielsweise die folgenden Befehle aus:

```

runmqsc QM1
#Define the server connection channel that will accept connections from the Console
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
# Define the listener to use for the connection from the Console
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
# Start the listener
START LISTENER(REMOTE.LISTENER)
end

#Set mq authorization for exampleUser to access the queue manager
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp

```

2. Auf dem System, auf dem IBM MQ Console ausgeführt wird, wird eine Datei `QM1_ccdt.json` mit den folgenden Verbindungsinformationen erstellt:

```
{
  "channel": [{
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }]
}
```

3. Auf dem System, auf dem IBM MQ Console ausgeführt wird, werden die Verbindungsinformationen des fernen Warteschlangenmanagers für Warteschlangenmanager QM1 zum mqweb-Server hinzugefügt. Die Berechtigungsnachweise für `exampleUser` sind in den Verbindungsinformationen enthalten:

```
setmqweb remote add -uniqueName "remote-QM1" -qmgrName "QM1" -ccdtURL
"c:\myccdt\QM1_ccdt.json" -username "exampleUser" -password "password"
```

4. IBM MQ Console zeigt den fernen Warteschlangenmanager QM1.

V 9.4.0 IBM MQ Console: Mit Objekten arbeiten

Jedem IBM MQ-Warteschlangenmanager sind mehrere unterschiedliche Objekttypen zugeordnet.

Informationen zu diesem Vorgang

Sie können die Konsole verwenden, um mit den folgenden Typen von IBM MQ-Objekten zu arbeiten:

- Warteschlangen
- Ereignisobjekte:
 - Themen
 - Subskriptionen
- Anwendungsobjekte:
 - Verbindungen
 - Anwendungskanäle
 - App-Kanalinstanzen
- MQ -Netzobjekte:
 - Verbundene Warteschlangenmanager
 - WS-Manager-Kanäle
 - Warteschlangenmanagerkanalinstanzen

Vorgehensweise

So arbeiten Sie mit einem IBM MQ-Objekt:


1. Klicken Sie in der Listenansicht mit den Warteschlangenmanagern auf den Warteschlangenmanager, der Eigner der Objekte ist, mit denen Sie arbeiten möchten.
2. Klicken Sie auf die Registerkarte "Warteschlangen", "Ereignisse", "Anwendungen" oder "MQ", um den Objekttyp auszuwählen, mit dem Sie arbeiten möchten.
3. Ausführliche Anweisungen zum Arbeiten mit den Objekten finden Sie in einem der folgenden Abschnitte.

V 9.4.0 IBM MQ Console: Mit Warteschlangen arbeiten

Sie können die Warteschlangen anzeigen, die für einen bestimmten Warteschlangenmanager auf der Registerkarte **Warteschlangen** vorhanden sind. Sie haben 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.

Informationen zu diesem Vorgang

In der Warteschlangenansicht werden die Warteschlangen aufgeführt, die es für einen bestimmten Warteschlangenmanager gibt. Sie greifen auf die Liste mit dem Warteschlangen zu, indem Sie auf einen Warteschlangenmanager klicken und die Registerkarte **Warteschlangen** auswählen. 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 auf der Registerkarte **Warteschlangen** auf die Schaltfläche 'Erstellen' .

b) Wählen Sie den Typ der Warteschlange aus, die Sie erstellen möchten:

- Lokale Warteschlange - Nachrichten werden im zugehörigen Warteschlangenmanager gespeichert.
- Aliaswarteschlange - Verweis auf eine andere Warteschlange im gleichen Warteschlangenmanager.
- Ferne Warteschlange - Verweis auf eine andere Warteschlange in einem anderen Warteschlangenmanager.
- Modellwarteschlange - Vorlage für eine Warteschlange, die beim Erstellen eines dynamischen Warteschlangenmanagers verwendet wird.

c) Geben Sie die erforderlichen Informationen für die Warteschlange an, die Sie erstellen. Standardmäßig werden die empfohlenen Mindesteigenschaften angezeigt, für die Sie Werte bereitstellen sollten. Sie können alle verfügbaren Eigenschaften anzeigen, indem Sie **Angepasste Erstellung** auswählen.

d) Klicken Sie auf **Erstellen**. Die neue Warteschlange wird erstellt.

- So reihen Sie Nachrichten in eine Warteschlange ein:

a) Klicken Sie in der Listenansicht mit den Warteschlangen auf die Warteschlange, der Sie Nachrichten hinzufügen möchten. Es kann keine Modellwarteschlange ausgewählt werden.

b) Klicken Sie auf die Schaltfläche Erstellen .

c) Geben Sie die Nachricht ein, die Sie in die Warteschlange einreihen möchten.

d) Klicken Sie auf **Erstellen**.

- So entfernen Sie Nachrichten aus einer Warteschlange:

a) Klicken Sie in der Warteschlangenliste auf die lokale Warteschlange, aus der Sie Nachrichten löschen möchten.


b) Klicken Sie auf das Symbol "Warteschlange löschen" .

c) Bestätigen Sie durch Klicken auf **Warteschlange leeren**, dass die Warteschlange geleert werden soll.

V 9.4.0

So löschen Sie eine einzelne Nachricht aus einer Warteschlange:

a) Suchen Sie die Nachricht, die Sie löschen möchten.

b) Klicken Sie auf das Löschsymbolsymbol neben der Nachricht .

c) Bestätigen Sie, dass Sie die Nachricht löschen möchten, indem Sie auf **Löschen** klicken.

• Um Nachrichten in einer Warteschlange zu suchen, klicken Sie auf die Warteschlange in der Listensicht mit den Warteschlangen. Es wird eine Liste der Nachrichten in dieser Warteschlange angezeigt.

• So löschen Sie eine Warteschlange:


a) Klicken Sie in der Warteschlangenliste auf die lokale Warteschlange, die Sie löschen möchten.


Actions ⋮

b) Klicken Sie auf die Schaltfläche 'Aktionen' und wählen Sie **Warteschlange löschen** aus.

c) 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 **Konfiguration anzeigen** im Menü  neben der Warteschlange aus, die Sie bearbeiten möchten.

b) Klicken Sie auf die Schaltfläche Bearbeiten .

c) Bearbeiten Sie die Eigenschaften wie gewünscht. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Informationen zu den Eigenschaften finden Sie unter [-Warteschlangeneigenschaften](#) in der IBM MQ Explorer -Dokumentation .

d) Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.

• So können Sie Berechtigungssätze für eine Warteschlange anzeigen und bearbeiten:

a) Wählen Sie **Konfiguration anzeigen** im Menü  neben der Warteschlange aus, die Sie bearbeiten möchten.

b) Klicken Sie auf die Registerkarte **Sicherheit**.

c) Bearbeiten Sie die Berechtigungssätze, wie dies für Berechtigungssätze für Warteschlangenmanager beschrieben ist. Weitere Informationen finden Sie unter [„IBM MQ Console: Mit Berechtigungsdatensätzen des Warteschlangenmanagers arbeiten“](#) auf Seite 105.

V 9.4.0

So zeigen Sie die IBM MQ -Objekte an, die einer Warteschlange zugeordnet sind:

a) Wählen Sie im Menü **Zugeordnete Objekte anzeigen** aus  neben der Warteschlange an, die Sie anzeigen wollen.

b) Zeigen Sie die Objekte in der Anzeige an, die angezeigt wird. Klicken Sie auf die Links, um weitere Details zu den aufgelisteten Objekten anzuzeigen.


Sie können die Anzeige verwenden, um anzuzeigen, welche Anwendungen Nachrichten in Warteschlangen einreihen, und die Beziehungen zwischen verschiedenen Warteschlangen anzuzeigen. Dies kann Ihnen bei der Identifizierung und Behebung von Problemen helfen.

IBM MQ Console: Mit Themen arbeiten

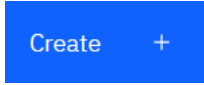


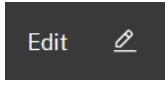
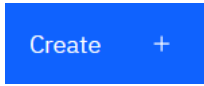
In der IBM MQ Console können Themen hinzugefügt und gelöscht sowie die Eigenschaften eines Themas angezeigt und festgelegt werden.

Informationen zu diesem Vorgang

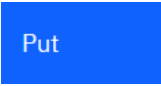

In der Themenansicht werden die Themen aufgeführt, die es für einen bestimmten Warteschlangenmanager gibt. Sie greifen auf die Themen über die Registerkarte **Ereignisse** des Warteschlangenmanagers zu. Sie können einzelne Themen aus der Liste auswählen, um mit ihnen zu arbeiten.

 Für Themen unter z/OS können keine Berechtigungssätze angezeigt oder bearbeitet werden.

Prozedur

- So fügen Sie ein Thema hinzu:
 - a) Öffnen Sie in der Warteschlangenmanageransicht die Registerkarte **Ereignisse** und klicken Sie auf **Themen**.
 - b) Klicken Sie auf die Schaltfläche Erstellen .
 - c) Geben Sie die erforderlichen Informationen für das Thema an, das Sie erstellen. Standardmäßig werden die empfohlenen Mindesteigenschaften angezeigt, für die Sie Werte bereitstellen sollten. Sie können alle verfügbaren Eigenschaften anzeigen, indem Sie **Angepasste Erstellung** auswählen.
 - d) Klicken Sie auf **Erstellen**. Das neue Thema wird erstellt.
- So löschen Sie ein Thema:
 - a) Klicken Sie neben dem Thema, das Sie löschen möchten, auf das Schraubenschlüsselsymbol .
 - b) Klicken Sie in der Ansicht zum Bearbeiten der Warteschlange auf **Thema löschen**.
 - 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) Klicken Sie neben dem Thema, das Sie bearbeiten möchten, auf das Schraubenschlüsselsymbol .
 - b) Klicken Sie auf die Schaltfläche Bearbeiten .
 - c) Bearbeiten Sie die Eigenschaften wie gewünscht. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Weitere Informationen zu den Eigenschaften finden Sie in der Dokumentation zu MQ Explorer unter [Themeneigenschaften](#).
 - d) Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.
- Zur Veröffentlichung einer Nachricht für ein Thema benötigen Sie mindestens eine übereinstimmende Subskription.
 - a) Klicken Sie in der Themenliste auf das Thema, in das Sie veröffentlichen möchten.
 - b) Klicken Sie auf den entsprechenden Subskriptionsnamen.
 - c) Klicken Sie auf die Schaltfläche Erstellen .
 - d) Geben Sie die Nachricht ein, die Sie veröffentlichen möchten.

Put

- e) Klicken Sie auf die Schaltfläche "Put"  Sie Nachricht wird in alle übereinstimmenden Subskriptionen geschrieben.
- Informationen zum Subskribieren eines Themas finden Sie unter „[IBM MQ Console: Mit Subskriptionen arbeiten](#)“ auf Seite 118:
- So können Sie Berechtigungssätze für ein Thema anzeigen und bearbeiten:
 - a) Klicken Sie neben dem Thema, für das Sie Berechtigungssätze bearbeiten möchten, auf das Schraubenschlüsselsymbol  .
 - b) Klicken Sie auf die Registerkarte **Sicherheit**.
 - c) Bearbeiten Sie die Berechtigungssätze, wie dies im Abschnitt „[IBM MQ Console: Mit Berechtigungsdatensätzen des Warteschlangenmanagers arbeiten](#)“ auf Seite 105 für Berechtigungssätze für Warteschlangenmanager beschrieben ist.

V 9.4.0

IBM MQ Console: Mit Subskriptionen arbeiten

In der IBM MQ Console können Subskriptionen hinzugefügt und gelöscht sowie die Eigenschaften einer Subskription angezeigt und festgelegt werden.

Informationen zu diesem Vorgang


In der Subskriptionsansicht werden die Subskriptionen aufgeführt, die es für einen bestimmten Warteschlangenmanager gibt. Sie greifen über die Registerkarte **Ereignisse** des Warteschlangenmanagers auf die Subskriptionen zu. Sie können einzelne Themen aus der Liste auswählen, um mit ihnen zu arbeiten. Sie können einzelne Subskriptionen aus der Liste auswählen, um mit ihnen zu arbeiten.

Weitere Informationen zu Subskriptionen finden Sie unter [Subskribenten und Subskriptionen](#) und [DEFINE SUB](#).

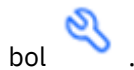
z/OS

Für Subskriptionen unter z/OS können keine Berechtigungssätze angezeigt oder bearbeitet werden.

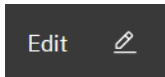
Prozedur

- So fügen Sie eine Subskription hinzu:
 - a) Öffnen Sie in der Warteschlangenmanageransicht die Registerkarte **Ereignisse** und klicken Sie auf **Subskriptionen**.
 - b) Geben Sie an, ob Sie eine verwaltete oder eine nicht verwaltete Subskription erstellen möchten.
 - c) Geben Sie die erforderlichen Informationen für die Subskription an, die Sie erstellen. Standardmäßig werden die empfohlenen Mindesteigenschaften angezeigt, für die Sie Werte bereitstellen sollten. Sie können alle verfügbaren Eigenschaften anzeigen, indem Sie **Angepasste Erstellung** auswählen.
 - d) Klicken Sie auf **Erstellen**. Die neue Subskription wird erstellt.
- So löschen Sie eine Subskription:
 - a) Klicken Sie neben der Subskription, die Sie löschen möchten, auf das Schraubenschlüsselsymbol  .
 - b) Klicken Sie in der Ansicht zum Bearbeiten der Warteschlange auf **Subskription löschen**.
 - c) Bestätigen Sie durch Klicken auf **Löschen**, dass die Subskription gelöscht werden soll. Die Subskription wird gelöscht.
- So können Sie die Eigenschaften einer Subskription anzeigen und bearbeiten:

a) Klicken Sie neben der Subskription, die Sie bearbeiten möchten, auf das Schraubenschlüsselsym-



b) Klicken Sie auf die Schaltfläche Bearbeiten

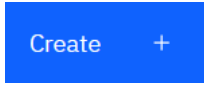


c) Bearbeiten Sie die Eigenschaften wie gewünscht. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden.

d) Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.

- So veröffentlichen Sie eine Nachricht im Thema, für das die Subskription subskribiert ist:

a) Klicke Sie auf die Subskription, deren Thema Sie in der Subskriptionsliste veröffentlichen möchten.



b) Klicken Sie auf die Schaltfläche Erstellen

c) Geben Sie die Nachricht ein, die Sie veröffentlichen möchten.



d) Klicken Sie auf die Schaltfläche "Put" Die Nachricht wird in alle Subskriptionen geschrieben, die mit den Thema übereinstimmen, für das Sie eine Veröffentlichung vorgenommen haben.

V 9.4.0

IBM MQ Console: Mit Warteschlangenmanagerkanälen arbeiten

Mit der IBM MQ Console können Sie Warteschlangenmanagerkanäle bearbeiten: Sie können Warteschlangenmanagerkanäle hinzufügen und löschen, Kanäle starten und stoppen, Kanäle zurücksetzen und auflösen sowie Pingsignale an Kanäle absetzen. Außerdem können die Eigenschaften eines Warteschlangenmanagerkanals angezeigt und bearbeitet und Berechtigungssätze für den Kanal verwaltet werden.

Informationen zu diesem Vorgang

Ein Warteschlangenmanagerkanal ist eine logische Kommunikationsverbindung für die Übertragung von Nachrichten zwischen Warteschlangenmanager in einem Netz. Die Ansicht mit den Warteschlangenmanagerkanälen enthält eine Anzeige, in der Sie direkt sehen können, wie viele Kanäle ausgeführt und wiederholt werden oder gestoppt sind.

z/OS

Für Kanäle unter z/OS können keine Berechtigungssätze angezeigt oder bearbeitet werden.

Prozedur

- So fügen Sie einen Warteschlangenmanagerkanal hinzu:

a) Öffnen Sie in der Warteschlangenmanageransicht die Registerkarte **MQ -Netz** und klicken Sie auf **Warteschlangenmanagerkanäle** . Klicken Sie dann auf die Schaltfläche 'Erstellen'



b) Wählen Sie den Typ des Warteschlangenmanagerkanals aus, den Sie erstellen möchten, und kli-




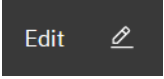







cken Sie auf die nächste Schaltfläche

c) Geben Sie die erforderlichen Informationen für den Kanal an, den Sie erstellen. Standardmäßig werden die empfohlenen Mindesteigenschaften angezeigt, für die Sie Werte bereitstellen sollten. Sie können alle verfügbaren Eigenschaften anzeigen, indem Sie **Angepasste Erstellung** auswählen.

d) Klicken Sie auf **Erstellen**. Der neue Kanal wird mit dem Status **inaktiv** erstellt.

- So starten Sie einen Warteschlangenmanagerkanal:

- a) Suchen Sie den Kanal, den Sie starten möchten, in der Liste.
- b) Wählen Sie den Eintrag **Start** im Menü aus  .
- So stoppen Sie einen Warteschlangenmanagerkanal:
 - a) Suchen Sie den Kanal, den Sie stoppen möchten, in der Liste.
 - b) Wählen Sie den Eintrag **Stop** im Menü aus  .
- So zeigen Sie die Eigenschaften eines Warteschlangenmanagerkanals an:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie **Konfigurationen anzeigen** im Menü aus  .
 - c) Stellen Sie sicher, dass die Registerkarte **Eigenschaften** ausgewählt ist. Klicken Sie zum Bearbeiten der Eigenschaften auf die Schaltfläche 'Bearbeiten'  .
 - d) Bearbeiten Sie die Eigenschaften wie gewünscht. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Weitere Informationen zu den Eigenschaften finden Sie im Abschnitt [Kanaleigenschaften](#) in der MQ Explorer-Dokumentation.
 - e) Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.
- So setzen Sie einen Warteschlangenmanagerkanal zurück:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie den Eintrag **Erweitert** im Menü aus  .
 - c) Geben Sie im Abschnitt **Zurücksetzen** 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 Sender- oder Serverkanal auf:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie den Eintrag **Erweitert** im Menü aus  .
 - c) Geben Sie im Abschnitt **Auflösen** an, ob der aktuelle Batch festgeschrieben oder zurückgesetzt werden soll, indem Sie auf **Nachrichten in Übertragungswarteschlange wiederherstellen** oder auf **Nachrichten löschen** klicken.
- So führen Sie eine Ping-Abfrage für einen Warteschlangenmanagerkanal aus:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie den Eintrag **Ping** im Menü aus  .
- So können Sie Berechtigungssätze für einen Warteschlangenmanagerkanal anzeigen und bearbeiten:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie **Konfiguration anzeigen** im Menü aus  aus.
 - c) Klicken Sie auf die Registerkarte **Sicherheit**.


- d) Bearbeiten Sie die Berechtigungssätze, wie dies im Abschnitt „IBM MQ Console: Mit Berechtigungssätzen des Warteschlangenmanagers arbeiten“ auf Seite 105 für Berechtigungssätze für Warteschlangenmanager beschrieben ist.
- So löschen Sie einen Warteschlangenmanagerkanal:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie den Eintrag **Konfigurieren** im Menü aus .
 - c) Klicken Sie auf **Kanal löschen**.

IBM MQ Console: Mit Anwendungskanälen arbeiten

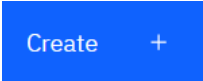




Mit der IBM MQ Console können Sie Anwendungskanäle bearbeiten: Sie können Kanäle hinzufügen und löschen, Kanäle starten und stoppen, Kanäle zurücksetzen und auflösen sowie Pingsignale an Kanäle absetzen. Außerdem können die Eigenschaften eines Anwendungskanals angezeigt und bearbeitet und Berechtigungssätze für den Kanal verwaltet werden.

Informationen zu diesem Vorgang


Ein Anwendungskanal ist eine logische Kommunikationsverbindung, die von Anwendungen verwendet wird, um eine Verbindung zu einem Warteschlangenmanager in einem Netz herzustellen. Die Ansicht mit den Anwendungskanälen enthält eine Anzeige, in der Sie direkt sehen können, wie viele Kanäle ausgeführt und wiederholt werden oder gestoppt sind.

 Für Kanäle unter z/OS können keine Berechtigungssätze angezeigt oder bearbeitet werden.






Prozedur

- So fügen Sie einen Anwendungskanal hinzu:
 - a) Öffnen Sie in der Warteschlangenmanageransicht die Registerkarte **Anwendungen** und klicken Sie auf **App-Kanäle**. Klicken Sie dann auf die Schaltfläche 'Erstellen' .
 - b) Klicken Sie auf die nächste Schaltfläche .
 - c) Geben Sie die erforderlichen Informationen für den Kanal an, den Sie erstellen. Standardmäßig werden die empfohlenen Mindesteigenschaften angezeigt, für die Sie Werte bereitstellen sollten. Sie können alle verfügbaren Eigenschaften anzeigen, indem Sie **Angepasste Erstellung** auswählen.
 - d) Klicken Sie auf **Erstellen**. Der neue Kanal wird mit dem Status **inaktiv** erstellt.
- So starten Sie einen Anwendungskanal:
 - a) Suchen Sie den Kanal, den Sie starten möchten, in der Liste.
 - b) Wählen Sie den Eintrag **Start** im Menü aus .
- So stoppen Sie einen Anwendungskanal:
 - a) Suchen Sie den Kanal, den Sie stoppen möchten, in der Liste.
 - b) Wählen Sie den Eintrag **Stop** im Menü aus .
- So zeigen Sie die Eigenschaften eines Anwendungskanals an:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie **Konfiguration anzeigen** im Menü  aus.

c) Stellen Sie sicher, dass die Registerkarte **Eigenschaften** ausgewählt ist. Klicken Sie zum Bearbeiten

Edit 

der Eigenschaften auf die Schaltfläche 'Bearbeiten' .

- d) Bearbeiten Sie die Eigenschaften wie gewünscht. Wenn das Eigenschaftstextfeld inaktiviert ist, ist die Eigenschaft schreibgeschützt oder sie kann nur aus der Befehlszeile bearbeitet werden. Weitere Informationen zu den Eigenschaften finden Sie im Abschnitt [Kanaleigenschaften](#) in der MQ Explorer-Dokumentation.
- e) Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.
- So setzen Sie einen Anwendungskanal zurück:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie den Eintrag **Erweitert** im Menü aus  .
 - c) Geben Sie im Abschnitt **Zurücksetzen** 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 Sender- oder Serverkanal auf:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie den Eintrag **Erweitert** im Menü aus  .
 - c) Geben Sie im Abschnitt **Auflösen** an, ob der aktuelle Batch festgeschrieben oder zurückgesetzt werden soll, indem Sie auf **Nachrichten in Übertragungswarteschlange wiederherstellen** oder auf **Nachrichten löschen** klicken.
 - So setzen Sie ein Pingsignal an einen Kanal ab:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie den Eintrag **Ping** im Menü aus  .
 - So können Sie Berechtigungssätze für einen Anwendungskanal anzeigen und bearbeiten:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie den Eintrag **Konfigurieren** im Menü aus  .
 - c) Klicken Sie auf die Registerkarte **Sicherheit**.
 - d) Bearbeiten Sie die Berechtigungssätze, wie dies im Abschnitt [„IBM MQ Console: Mit Berechtigungssätzen des Warteschlangenmanagers arbeiten“](#) auf Seite 105 für Berechtigungssätze für Warteschlangenmanager beschrieben ist.
 - So löschen Sie einen Anwendungskanal:
 - a) Suchen Sie den Kanal in der Liste.
 - b) Wählen Sie den Eintrag **Konfigurieren** im Menü aus  .
 - c) Klicken Sie auf **Kanal löschen**.

IBM MQ Console: Mit Anwendungen arbeiten

Sie können die IBM MQ Console verwenden, um Informationen zu Anwendungen anzuzeigen, die mit einem Warteschlangenmanager verbunden sind.

Informationen zu diesem Vorgang

Eine Anwendung ist über einen Serververbindungskanal mit einem Warteschlangenmanager über ein Netz verbunden. Die Anwendungsansicht enthält eine Anzeige, in der Sie schnell sehen können, wie viele Anwendungen mit einem Warteschlangenmanager verbunden sind.

Prozedur

- So zeigen Sie Anwendungsinformationen an:
 - a) Öffnen Sie in der Warteschlangenmanageransicht die Registerkarte **Anwendungen**.
 - b) Klicken Sie auf **verbundene Anwendungen**, um die Anwendungsansicht zu öffnen.
 - c) Wenn mehrere Instanzen einer Anwendung vorhanden sind, klicken Sie auf den Abwärtspfeil, um Details zu jeder Instanz anzuzeigen.
 - d) Klicken Sie auf Objekte in der Ansicht, um weitere Details abzurufen.

IBM MQ Console: Working with storage classes

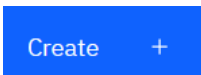
You can use the IBM MQ Console to add, view, delete and update storage classes on z/OS queue managers.

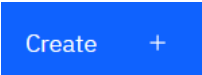
About this task

The storage classes view lists the storage classes that exist for a specific queue manager. You access **Storage classes** from the sidebar on the queue manager **Queues** tab.

See [Storage classes for IBM MQ for z/OS](#) and [DEFINE STGLASS](#) for more information about storage classes.


Procedure

- To add a storage class:
 - a) From the queue manager view, open the **Queues** tab, and click **Storage classes**.
 - b) On the **Storage classes** screen, click the **Create**  button.
 - c) Provide the required information for the storage class you are creating.
By default, the minimum recommended properties you need to provide values for are displayed. You can view all of the available properties by selecting **Custom create**.

- d) Click the **Create**  button.
The new storage class is created.

- To delete a storage class:



- a) Click the spanner button  next to the storage class that you want to delete.
 - b) In the Edit storage class view, click **Delete storage class**.
 - c) Confirm that you want to delete the queue by clicking **Delete**. The storage class is deleted.
- To view and edit the properties of a storage class:



a) Click the spanner button next to the storage class that you want to edit.

Edit

b) Click the Edit button

c) Edit the properties as required. If the property text box is disabled, the property is read-only, or can be set only at the time of creation.

d) Click **Save** to save your changes.

V 9.4.0 z/OS V 9.4.0 **IBM MQ Console: Working with page sets and buffer pools**

You can use the IBM MQ Console to view page sets and buffer pools on z/OS queue managers.

About this task

The page sets and buffer pools views list the page sets and buffer pools that exist for a specific queue manager. You access the **Page sets** and **Buffer pools** views from the sidebar of the queue manager **Queues** tab

See [Page sets for IBM MQ for z/OS](#) for more information about page sets, and [Buffers and buffer pools for IBM MQ for z/OS](#) for more information about buffer pools.

Procedure

- To view the properties of a page set



Click the spanner button next to the page set that you want to view.

- To view the properties of a buffer pool



Click the spanner button next to the buffer pool that you want to view.

IBM MQ Console Einstellungen

Sie können einige allgemeine Einstellungen für IBM MQ Console angeben.

Klicken Sie auf das Symbol für Einstellungen **Settings**, um zur Ansicht für IBM MQ Console -Einstellungen zu wechseln.

Mithilfe der Einstellungen können Sie die folgenden Funktionen steuern:

- Automatische Aktualisierung von Warteschlangenmanagern alle zehn Sekunden. Diese Funktion kann aktiviert oder inaktiviert werden.
- Angabe, ob Systemobjekte angezeigt werden sollen. Sie können dies für alle Objekttypen angeben oder die Objekttypen einzeln auswählen.
- Angabe, ob Traceinformationen erfasst werden.

Mit der IBM MQ Explorer können Sie die lokale oder ferne Verwaltung Ihres Netzes von einem Computer aus ausführen, auf dem Windows oder Linux x86-64 nur.

IBM MQ for Windows und IBM MQ for Linux x86-64 stellen eine Verwaltungsschnittstelle mit dem Namen IBM MQ Explorer bereit, um Verwaltungstasks alternativ zur Verwendung von Steuerungs- oder MQSC-Befehlen auszuführen. [Befehlssätze vergleichen](#) zeigt Ihnen, was Sie mit IBM MQ Explorer tun können.

Mit der IBM MQ Explorer können Sie die lokale oder ferne Verwaltung Ihres Netzes von einem Computer ausführen, auf dem Windows ausgeführt wird, oder Linux x86-64, indem Sie den IBM MQ Explorer auf die Warteschlangenmanager und Cluster, die Sie interessieren, zeigen. 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 127.

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.

Unter Linux müssen Sie zum erfolgreichen Starten von IBM MQ Explorer in der Lage sein, eine Datei in Ihr Ausgangsverzeichnis zu schreiben, und das Ausgangsverzeichnis muss vorhanden sein.

IBM MQ Explorer kann über den eigenständigen IBM MQ Explorer -Download von Fix Central installiert werden. Weitere Informationen finden Sie unter [IBM MQ Explorer als eigenständige Anwendung unter Linux und Windows installieren und deinstallieren](#).

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](#) eines Warteschlangenmanagers (nur auf Ihrer lokalen Maschine).
- [Starten und stoppen](#) eines Warteschlangenmanagers (nur auf Ihrer lokalen Maschine).
- [Definieren, anzeigen und ändern](#) der Definitionen von IBM MQ-Objekten wie Warteschlangen und Kanälen.
- [Durchsuchen](#) der Nachrichten in einer Warteschlange.
- [Starten und stoppen](#) eines Kanals.
- [Anzeigen von Statusinformationen](#) zu einem Kanal, Listener, einer Warteschlange oder zu Serviceobjekten.
- Anzeigen von Warteschlangenmanagern in einem Cluster.
- Überprüfen, welche [Anwendungen, Benutzer oder Kanäle eine bestimmte Warteschlange geöffnet haben](#).
- [Erstellen eines neuen Warteschlangenmanagerclusters](#) mit dem Assistenten 'Neuen Cluster erstellen'.

- Hinzufügen eines Warteschlangenmanagers zu einem Cluster mit dem Assistenten 'Warteschlangenmanager zu Cluster hinzufügen'.
- Verwalten des Authentifizierungsdatenobjekts, das mit der TLS-Kanalsicherheit (Transport Layer Security) verwendet wird.
- Kanalinitiatoren, Auslösemonitore und Empfangsprogramme erstellen und löschen.
- Starten oder stoppen des Befehlsservers, der Kanalinitiatoren, Auslösemonitore und Empfangsprogramme.
- Festlegen bestimmter Services, die beim Start eines Warteschlangenmanagers automatisch gestartet werden.
- Ändern der Merkmale von Warteschlangenmanagern.
- Ändern des lokalen Standardwarteschlangenmanagers.
- Erstellen von JMS-Objekten aus IBM MQ-Objekten und von IBM MQ-Objekten aus JMS-Objekten.
- Erstellen einer 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.
- Starten oder stoppen des Service-Trace.

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 127](#)
- [„Sicherheit für IBM MQ Explorer“ auf Seite 127](#)
- [„Warteschlangenmanager und Cluster in IBM MQ Explorer ein- und ausblenden“ auf Seite 131](#)
- [„Clusterzugehörigkeit und IBM MQ Explorer“ auf Seite 132](#)
- [„Datenkonvertierung für IBM MQ Explorer“ auf Seite 133](#)

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 AIX 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:

- Ein 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 UPDATE-Berechtigung 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 MQCMDS

Informationen zum Erteilen von Berechtigungen für IBM MQ-Objekte finden Sie unter [Zugriff auf ein IBM MQ-Objekt auf AIX, Linux, and Windows-Systemen gewähren](#).

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 einzelnen 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 mithilfe einer Definitionstabelle für Clientkanäle finden Sie unter [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 129 und „Tasks auf dem System, das den IBM MQ Explorer hostet“ auf Seite 130 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 in [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 *SSL* -Verzeichnis des Warteschlangenmanagers ein:
 -   `/var/mqm/qmgrs/+QMNAME+/SSL` für AIX and Linux -Systeme.
 -  `C:\Programme\IBM\MQ\qmgrs\+QMNAME+\SSL` für Windows -Systeme.Dabei steht *+QMNAME+* für ein Token, das den Namen des Warteschlangenmanagers darstellt.
5. Erstellen Sie eine Schlüsseldatenbankdatei des Typs *CMS* mit dem Namen *key* . *kdb*. Speichern Sie das Schlüsseldatenbankkennwort in einer Datei verdeckt, indem Sie den Parameter *-stash* im Befehl **runmqakm** angeben, der zum Erstellen der Schlüsseldatenbank verwendet wird.
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 AIX, 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 des Typs JKS mit dem Namen `key.jks`. Legen Sie ein Kennwort für diese Schlüsseldatenbankdatei fest.

Der Keystore, den IBM MQ Explorer für die TLS-Sicherheit verwendet, muss eine Java -Keystore-Datei (JKS) sein.
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 `strmqcfcfg` 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 130 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 129 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).

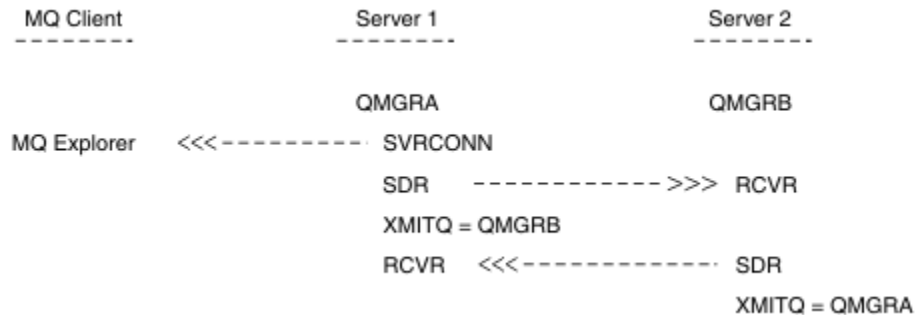
Wenn die Verbindung dann zum Herstellen der Verbindung zum Zielwarteschlangenmanager über einen temporären Warteschlangenmanager verwendet wird, wird die Benutzer-ID erneut im Parameter *UserIdentifier* des Nachrichtendeskriptors (MQMD) ausgegeben. Damit der MCA-Listener auf dem Zielwarteschlangenmanager 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 QMGR A (auf Server1) verbunden.
- Warteschlangenmanager QMGR B auf Server2 kann jetzt über einen temporären Warteschlangenmanager (QMGR A) mit IBM MQ Explorer verbunden werden
- Wenn Sie eine Verbindung zu QMGR B mit IBM MQ Explorer herstellen, wählen Sie QMGR A als temporären Warteschlangenmanager aus.

In dieser Situation besteht keine direkte Verbindung zu QMGR B von IBM MQ Explorer. Die Verbindung zu QMGR B erfolgt über QMGR A.

Der WS-Manager QMGR B auf Server2 ist mit QMGR A auf Server1 mit Sender-Empfänger-Kanälen verbunden. Der Kanal zwischen QMGR A und QMGR B muss so konfiguriert werden, dass die Fernverwaltung möglich ist (siehe „WS-Manager für die Fernverwaltung konfigurieren“ auf Seite 207).

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 Anwendung " IBM MQ Taskbar" 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 Alertnachrichten über das Netz an ein konfigurierbares Benutzerkonto oder an eine Windows-Workstation oder einen Windows-Server

Mit lokalen IBM MQ-Objekten arbeiten

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 die IBM MQ Explorer verwenden, um lokale IBM MQ -Objekte zu verwalten. Weitere Informationen finden Sie unter [„Verwaltung über die IBM MQ Explorer“](#) auf Seite 125.

Prozedur

- Die Informationen in den folgenden Abschnitten helfen Ihnen bei der Verwaltung lokaler IBM MQ-Objekte.
 - [Anwendungsprogramme, die MQI verwenden](#)
 - [„IBM MQ mit MQSC-Befehlen verwalten“](#) auf Seite 12
 - [„Warteschlangenmanagerattribute anzeigen und ändern“](#) auf Seite 142
 - [„Mit lokalen Warteschlangen arbeiten“](#) auf Seite 146
 - [„Mit Aliaswarteschlangen arbeiten“](#) auf Seite 159
 - [„Mit Modellwarteschlangen arbeiten“](#) auf Seite 161
 - [„Mit Services arbeiten“](#) auf Seite 192
 - [„Objekte zum Auslösen verwalten“](#) auf Seite 200

Mit Warteschlangenmanagern arbeiten

Mithilfe von Steuerbefehlen können Sie einen Warteschlangenmanager starten und stoppen. Sie können MQSC-Befehle verwenden, um Warteschlangenmanagerattribute anzuzeigen oder zu ändern.

Zugehörige Tasks



[Warteschlangenmanager auf Multiplatforms erstellen](#)

WS-Manager starten

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

Informationen zu diesem Vorgang

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

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

Windows Unter Windows können Sie einen Warteschlangenmanager über den IBM MQ Explorer automatisch starten, wenn das System gestartet wird. Weitere Informationen finden Sie unter „[Verwaltung über die IBM MQ Explorer](#)“ auf Seite 125.

Prozedur

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

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

```
strmqm QMB
```

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

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

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

Ergebnisse

Der WS-Manager wird gestartet.

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

Multi Stoppen eines Warteschlangenmanagers

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

Informationen zu diesem Vorgang

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

Gesteuerter Abschluss (Quiesce)

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

Sofortige Beendigung

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

Präventiver Systemabschluss

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

Verzögerter Systemabschluss

Dies entspricht in etwa einer kontrollierten Beendigung, allerdings wird die Steuerung erst nach Beendigung des Warteschlangenmanagers an den Benutzer zurückgegeben.

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

Sie haben die Möglichkeit, den Warteschlangenmanager innerhalb einer von Ihnen angegebenen Zielzeit von einer bestimmten Anzahl von Sekunden zu beenden, entweder mit oder ohne Unterbrechung nicht wesentlicher Wartungsaufgaben des Warteschlangenmanagers (siehe „[Warteschlangenmanager innerhalb einer Zielzeit beenden](#)“ auf Seite 138).



Achtung:

- Persistente Nachrichten bleiben unabhängig vom verwendeten Beendigungstyp (einschließlich der manuellen Beendigung von IBM MQ -Prozessen) bestehen, während nicht persistente Nachrichten nicht garantiert werden können, dass sie jeden Beendigungstyp überleben.

Wenn Sie die Warteschlangeneigenschaft NPMCLASS (HIGH) angeben, werden nicht persistente Nachrichten auf einer Best-can-do-Basis gespeichert. Die Verwendung von **endmqm -t**, **endmqm -tp**, **endmqm -p** oder manuell beendeten IBM MQ -Prozessen verringert die Wahrscheinlichkeit, dass NPMCLASS-Nachrichten (HIGH-HIGH-Nachrichten) einen IBM MQ -Systemabschluss- oder -Neustartzyklus überstehen, im Vergleich zu **endmqm -w** oder **endmqm -i**.

- Die kombinierte Zeit bis zum Beenden und Neustart des Warteschlangenmanagers kann aufgrund der Verwendung einer abrupteren Beendigungsmethode länger sein, insbesondere bei Verwendung der Optionen **-p** und **-tp**.

Wenn der Warteschlangenmanager auf das Beenden von IBM MQ-Prozessen zurückgreifen muss, um den Warteschlangenmanager zu beenden, ist wahrscheinlich ein weiterer Abgleich des Warteschlangenmanagerstatus erforderlich, wenn der Warteschlangenmanager erneut gestartet wird.

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

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

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

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



Als Alternative zur Verwendung des Befehls **endmqm** beim Stoppen eines Warteschlangenmanagers können Sie unter Windows und Linux auch IBM MQ Explorer zum gesteuerten oder sofortigen Herunterfahren verwenden.

Prozedur

- Um den Warteschlangenmanager mit dem Befehl **endmqm** zu stoppen, geben Sie den Befehl gefolgt vom entsprechenden Parameter (falls erforderlich) und dem Namen des Warteschlangenmanagers ein, der gestoppt werden soll.

Anmerkung: Sie müssen den Befehl **endmqm** aus der Installation verwenden, die dem Warteschlangenmanager zugeordnet ist, mit dem Sie arbeiten. Um herauszufinden, welcher Installation ein Warteschlangenmanager zugeordnet ist, verwenden sie den Befehl **dspmqs**:

```
dspmqs -o installation
```

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

```
endmqm QMB
```

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

```
endmqm -c QMB
```

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

```
endmqm -w QMB
```


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

```
endmqm -i QMB
```

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

```
endmqm -p QMB
```



Achtung: Ein präventiver Abschluss kann für verbundene Anwendungen unvorhersehbare Auswirkungen haben. Verwenden Sie diese Option nur, wenn alle anderen Versuche, den Warteschlangenmanager mit einem normalen **endmqm**-Befehl zu stoppen, fehlgeschlagen sind.  Wenn der präventive Abschluss nicht funktioniert, versuchen Sie stattdessen „Manuelles Stoppen eines Warteschlangenmanagers“ auf Seite 139.

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

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

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



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

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

Der Warteschlangenmanager wird gestoppt.

Zugehörige Tasks

[Aktualisierungen der Wartungsstufe für Multi-Instanz-Warteschlangenmanager unter AIX anwenden](#)

[Aktualisierungen der Wartungsstufe für Multi-Instanz-Warteschlangenmanager unter Linux anwenden](#)

[Aktualisierungen der Wartungsstufe für Multi-Instanz-Warteschlangenmanager unter Windows anwenden](#)

Zugehörige Verweise

[endmqm \(Warteschlangenmanager beenden\)](#)

Warteschlangenmanager innerhalb einer Zielzeit beenden

Sie können den Warteschlangenmanager innerhalb einer von Ihnen angegebenen Zielzeit von mehreren Sekunden beenden, entweder mit oder ohne Unterbrechung nicht wesentlicher Wartungsaufgaben des Warteschlangenmanagers.

Es gibt zwei Möglichkeiten, eine Zielzeit anzugeben, wenn Sie den Befehl **endmqm** verwenden. Mit der Option **-t** können alle Wartungsaufgaben des Warteschlangenmanagers abgeschlossen werden, was die Phase der Beendigung des Warteschlangenmanagers verlängern kann. Die Option **-tp** unterbricht nicht wichtige Wartungsaufgaben des Warteschlangenmanagers, wenn dies erforderlich ist, um die angegebene Zielzeit einzuhalten.

Zu den nicht wesentlichen Verwaltungstasks gehören die Komprimierung von Warteschlangendateien und die persistente Speicherung von NPMCLASS-Nachrichten (HIGH). Im Rest dieser Seite wird das Wort 'Housekeeping' verwendet.

Abhängig von Ihren Anwendungsnutzungsmustern kann die Komprimierung der Warteschlangendatei viel Zeit in Anspruch nehmen. Wenn Ihr primäres Ziel also das schnelle Beenden des Warteschlangenmanagers ist, verwenden Sie die Option **-tp** .

Wenn Sie eine Zielzeit vorgeben, wird mit dem Beendigungstyp **-w**, **-i** oder **-p** angezeigt, mit welchem Typ der Systemabschluss gestartet wird.

Anmerkung: Ein **immediate** -Systemabschluss ist weiterhin ordnungsgemäß, was sich von einem **controlled** -Systemabschluss in erster Linie dadurch unterscheidet, dass alle aktiven Anwendungen in den Quiescemodus versetzt werden. Ein **immediate** -Systemabschluss führt weiterhin Verwaltungsaufgaben aus. Eine zeitlich begrenzte Beendigung beendet diese Aktionen, wenn sie die Zielzeit beeinträchtigen.

Der Warteschlangenmanager eskaliert bei Bedarf den Beendigungstyp beim Versuch, die Vorgabezeit zu erfüllen. For example:

- Ein Ziel **-t** von 10 Sekunden, das bei **-w** beginnt, kann sieben Sekunden für das Versetzen in den Quiescemodus, zwei Sekunden für die sofortige Beendigung des Warteschlangenmanagers (einschließlich Housekeeping) mit anschließendem Systemabschluss ohne weitere Housekeeping-Aktionen beanspruchen:

```
endmqm -w -t 10 queue_manager
```

- Ein Ziel **-tp** von 10 Sekunden kann sieben Sekunden für das Versetzen in den Quiescemodus, zwei Sekunden für die sofortige Beendigung des Warteschlangenmanagers (einschließlich Housekeeping), eine Sekunde für die sofortige Beendigung ohne weitere Housekeeping-Aktionen mit anschließender Beendigung der IBM MQ-Prozesse beanspruchen:

```
endmqm -c -tp 10 queue_manager
```

- Ein Ziel **-tp** von zwei Sekunden in **-i** kann eine Sekunde für die sofortige Beendigung des Warteschlangenmanagers (einschließlich Housekeeping), eine Sekunde für die sofortige Beendigung ohne weitere Housekeeping-Aktionen und anschließende Beendigung der IBM MQ-Prozesse beanspruchen:

```
endmqm -i -tp 2 queue_manager
```

- Ein Ziel von einer Sekunde in **-w** kann 0,1 Sekunden in wait beanspruchen, was beispielsweise gerade ausreicht, um IBM MQ-Rückkehrcodes an verbundene Anwendungen zu senden, 0,9 Sekunden für die sofortige Beendigung des Warteschlangenmanagers (einschließlich Housekeeping) und anschließend sofortige Beendigung ohne weitere Housekeeping-Aktionen. Die IBM MQ-Prozesse werden anschließend beendet.

Zugehörige Verweise

[endmqm \(Warteschlangenmanager beenden\)](#)

ALW Manuelles Stoppen eines Warteschlangenmanagers

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 „[Stoppen eines Warteschlangenmanagers](#)“ auf Seite 135 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 139.
- **Linux** **AIX**
Wie Sie einen Warteschlangenmanager auf AIX oder Linux anhalten können, erfahren Sie unter „[Warteschlangenmanager unter AIX and Linux manuell stoppen](#)“ auf Seite 140.

Zugehörige Tasks

[Warteschlangenmanager auf Multiplatforms erstellen und verwalten](#)

Zugehörige Verweise

[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 <http://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 7. 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
AMQZMUFO	Utility-Manager
AMQZMGRO	Prozesscontroller
AMQZMUR0	Wiederanlauffähiger Prozessmanager
AMQFQ PUB	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.

Linux AIX **Warteschlangenmanager unter AIX and Linux manuell stoppen**

Wenn Sie einen Warteschlangenmanager auf AIX oder Linux nicht mit dem Befehl **endmqm** stoppen können, können Sie versuchen, den Warteschlangenmanager manuell anzuhalten, indem Sie alle laufenden Prozesse beenden und den IBM MQ-Service stoppen.

Informationen zu diesem Vorgang

Um einen Warteschlangenmanager auf AIX oder Linux anzuhalten, führen Sie die folgenden Schritte aus.

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 mit dem Befehl **ps** nach den Prozess-IDs (PIDs) der Warteschlangenmanagerprogramme, die noch aktiv sind.

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:

<i>Tabelle 8. AIX and Linux-Prozesse, die gestoppt werden sollen, falls sie aktiv sind</i>	
Prozessname	Beschreibung
amqzmuc0	Kritischer Prozessmanager
amqzma0	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

Zugehörige Tasks

„Stoppen eines Warteschlangenmanagers“ auf Seite 135

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

Multi **WS-Manager erneut starten**

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

Informationen zu diesem Vorgang

Sie können einen Warteschlangenmanager mit dem Befehl **strmqm** erneut starten. Eine Beschreibung des Befehls **strmqm** und seiner Optionen finden Sie unter [strmqm](#).

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

Prozedur

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

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

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

Ergebnisse

Der WS-Manager wird erneut gestartet.

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

Warteschlangenmanagerattribute anzeigen und ändern

Mit dem MQSC-Befehl **DISPLAY QMGR** können Sie die Warteschlangenmanagerparameter für einen Warteschlangenmanager anzeigen. Mit dem MQSC-Befehl **ALTER QMGR** können Sie die Warteschlangenmanagerparameter für einen lokalen Warteschlangenmanager ändern.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Prozedur

- Um die Attribute des im Befehl **runmqsc** angegebenen Warteschlangenmanagers anzuzeigen, verwenden Sie den MQSC-Befehl **DISPLAY QMGR** :

```
DISPLAY QMGR
```

Im folgenden Beispiel wird die typische Ausgabe dieses Befehls gezeigt:

```
DISPLAY QMGR
  1 : DISPLAY QMGR
AMQ8408I: Display Queue Manager details.
  QMNAME(QM1)                ACCTCONO(DISABLED)
  ACCTINT(1800)              ACCTMQI(OFF)
  ACCTQ(OFF)                 ACTIVREC(MSG)
  ACTVCONO(DISABLED)        ACTVTRC(OFF)
  ADVCAP(DISABLED)          ALTDATE(2022-05-05)
  ALTTIME(14.24.34)         AMQPCAP(NO)
  AUTHOREV(DISABLED)        CCSID(437)
  CERTLABL(ibmwebspheremqm1) CERTVPOL(ANY)
  CHAD(DISABLED)            CHADEV(DISABLED)
  CHADEXIT( )                CHLEV(DISABLED)
  CHLAUTH(ENABLED)          CLWLDATA( )
  CLWLEXIT( )                CLWLLEN(100)
  CLWLMRUC(999999999)       CLWLUSEQ(LOCAL)
  CMDEV(DISABLED)           CMDLEVEL(930)
  COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) CONFIGEV(DISABLED)
  CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
  CRDATE(2020-12-22)        CRTIME(15.42.49)
  CUSTOM( )                 DEADQ( )
  DEFCLXQ(SCTQ)             DEFXMITQ( )
  DESCR( )                   DISTL(YES)
  IMGINTVL(60)              IMGLOGLN(OFF)
  IMGRCOVO(YES)             IMGRCOVQ(YES)
  IMGSCHEM(MANUAL)         INHIBTEV(DISABLED)
  INITKEY( )                IPADDRV(IPV4)
  LOCALEV(DISABLED)        LOGGERSV(DISABLED)
  MARKINT(5000)            MAXHANDS(256)
  MAXMSGL(4194304)         MAXPROPL(NOLIMIT)
  MAXPRTY(9)               MAXUMSGS(10000)
  MONACLS(QMGR)            MONCHL(OFF)
  MONQ(OFF)                 PARENT( )
  PERFMEV(DISABLED)        PLATFORM(WINDOWS10)
  PSMODE(ENABLED)          PSCLUS(ENABLED)
  PSNPMMSG(DISCARD)        PSNPRES(NORMAL)
  PSRTYCNT(5)              PSSYNCP(IFPER)
  QMID(QM1_2020-12-22_15.42.49) REMOTEEV(DISABLED)
  REPOS( )                  REPOSNL( )
  REVDNS(ENABLED)          ROUTEREC(MSG)
  SCHINIT(QMGR)            SCMDSERV(QMGR)
  SPLCAP(DISABLED)         SSLCRLNL( )
  SSLCRYP( )               SSLEV(DISABLED)
  SSLFIPS(NO)              KEYRPWD( )
  SSLKEYR(C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key)
  SSLRKEYC(32767)          STATACLS(QMGR)
  STATCHL(OFF)             STATINT(1800)
  STATMQI(OFF)             STATQ(OFF)
  STRSTPEV(ENABLED)        SUITEB(NONE)
  SYNCPT                    TREELIFE(1800)
  TRIGINT(999999999)       VERSION(09030000)
  XRCAP(NO)
```

Anmerkung: SYNCPT 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.

- Um die Attribute des im Befehl **runmqsc** angegebenen Warteschlangenmanagers zu ändern, geben Sie mit dem MQSC-Befehl **ALTER QMGR** 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)
```

Der Befehl **ALTER QMGR** ändert die verwendete Warteschlange für nicht zustellbare Nachrichten und aktiviert Sperrereignisse.

Wenn Parameter mit dem Befehl **ALTER QMGR** nicht angegeben werden, werden die vorhandenen Werte für diese Parameter unverändert übernommen.

Zugehörige Tasks

[Warteschlangenmanager auf Multiplatforms erstellen](#)

Zugehörige Verweise

[Attribute für den Warteschlangenmanager](#)

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

[ANZEIGEN QMGR](#)

[ALTER QMGR](#)

Multi

Löschen eines Warteschlangenmanagers

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

Vorbereitende Schritte



Achtung:

- Das Löschen eines Warteschlangenmanagers ist ein drastischer Schritt, da Sie auch alle Ressourcen löschen, die dem Warteschlangenmanager zugeordnet sind, einschließlich aller Warteschlangen und deren Nachrichten und aller Objektdefinitionen. Wenn Sie den Steuerbefehl **dltmqm** verwenden, wird keine Eingabeaufforderung angezeigt, in der Sie Ihre Meinung ändern können. Wenn Sie die Eingabetaste drücken, gehen alle zugeordneten Ressourcen verloren.
- **Windows** Unter Windows wird beim Löschen eines Warteschlangenmanagers auch der Warteschlangenmanager aus der automatischen Startliste gelöscht (siehe „[WS-Manager starten](#)“ auf [Seite 134](#)). Nach Abschluss des Befehls wird die Nachricht `IBM MQ queue manager ending` angezeigt. Sie werden nicht darüber informiert, dass der Warteschlangenmanager gelöscht wurde.
- Wenn Sie einen Cluster-WS-Manager löschen, wird er nicht aus dem Cluster entfernt. Weitere Informationen finden Sie in den Hinweisen zur Verwendung in [dltmqm](#).

Informationen zu diesem Vorgang

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

Windows

Linux


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

Prozedur

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

```
dltmqm QMB
```

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

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

Ergebnisse

Der WS-Manager wird gelöscht.

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-Warteauftrag 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_CONNECTON_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 Konzepte

[Kanäle](#)

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 Verweise

[Benennungseinschränkungen für Warteschlangen](#)

[Benennungseinschränkungen für andere Objekte](#)

Lokale Warteschlange mit DEFINE QLOCAL 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. Mit dem MQSC-Befehl **DEFINE QLOCAL** erstellen Sie eine lokale Warteschlange.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Informationen zu diesem Vorgang

Mit dem MQSC-Befehl **DEFINE QLOCAL** erstellen Sie eine lokale Warteschlange. 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 wird bei der Systeminstallation erstellt.

Prozedur

- Um eine lokale Warteschlange zu erstellen, geben Sie den Befehl **DEFINE QLOCAL** wie im folgenden Beispiel gezeigt ein.
In diesem Beispiel definiert der Befehl **DEFINE QLOCAL** 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)

```

Anmerkungen:

1. Mit Ausnahme des Werts für die Beschreibung sind alle Attributwerte, die in dem Beispiel dargestellt sind, die Standardwerte. Diese Beispiele sind zu Illustrationszwecken enthalten. 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 [„Standardobjektattribut mit DISPLAY QUEUE anzeigen“](#) auf Seite 147.
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 [„Lokale Warteschlangenattribute mit ALTER QLOCAL oder DEFINE QLOCAL ändern“](#) auf Seite 149.

Zugehörige Verweise

[QLOCAL DEFINIER](#)

Standardobjektattribut mit DISPLAY QUEUE anzeigen

Mit dem MQSC-Befehl **DISPLAY QUEUE** können Sie Attribute anzeigen, die aus dem Standardobjekt übernommen wurden, als ein IBM MQ -Objekt definiert wurde.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Informationen zu diesem Vorgang

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. Mit dem Befehl **DISPLAY QUEUE** können Sie diese Attribute exakt anzeigen.

Prozedur

- Verwenden Sie den folgenden Befehl, um die Standardobjektattribute für eine lokale Warteschlange anzuzeigen:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Die Syntax des Befehls **DISPLAY** unterscheidet sich von der Syntax des entsprechenden Befehls **DEFINE**. Im Befehl **DISPLAY** können Sie nur den Warteschlangennamen angeben, während Sie im **DEFINE** -Befehl den Typ der Warteschlange angeben müssen, also QLOCAL, QALIAS, QMODEL oder QREMOTE.

Sie können Attribute selektiv anzeigen, indem Sie sie einzeln angeben. For example:

```
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 gibt die aktuelle Warteschlangenlänge an, 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 Verweise

[ANZEIGEN QUEUE](#)

[Warteschlangen definieren \(DEFINE\)](#)

Lokale Warteschlangendefinition mit DEFINE QLOCAL kopieren

Sie können eine Warteschlangendefinition mit dem Attribut **LIKE** im MQSC-Befehl **DEFINE QLOCAL** kopieren.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Informationen zu diesem Vorgang

Sie können den Befehl **DEFINE** mit dem Attribut **LIKE** verwenden, um eine Warteschlange mit den gleichen Attributen wie in der angegebenen Warteschlange anstelle der Attribute der lokalen Standardwarteschlange zu erstellen. Sie können diese Form des **DEFINE**-Befehls auch verwenden, um eine Warteschlangendefinition zu kopieren, aber eine oder mehrere Änderungen an den Attributen des Originals ersetzen.

Anmerkungen:

1. Wenn Sie das Attribut **LIKE** in einem **DEFINE**-Befehl verwenden, kopieren Sie nur die Warteschlangenattribute. Die Nachrichten werden nicht in die Warteschlange kopiert.
2. Wenn Sie eine lokale Warteschlange definieren, ohne **LIKE** anzugeben, ist sie identisch mit:

```
DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Prozedur

- Wenn Sie anstelle der lokalen Standardwarteschlange des Systems eine Warteschlange mit den gleichen Attributen wie die angegebene Warteschlange erstellen möchten, geben Sie den Befehl **DEFINE** wie im folgenden Beispiel gezeigt ein.

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.

In diesem Beispiel wird eine Warteschlange mit den gleichen Attributen wie die Warteschlange **ORANGE.LOCAL.QUEUE** und nicht die der lokalen Standardwarteschlange für das System erstellt:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +
LIKE (ORANGE.LOCAL.QUEUE)
```

- Wenn Sie eine Warteschlangendefinition kopieren möchten, dabei aber eine oder mehrere Änderungen an den Attributen der ursprünglichen Definition ersetzen möchten, geben Sie den Befehl **DEFINE** wie im folgenden Beispiel gezeigt ein.

Mit diesem Befehl werden die Attribute der Warteschlange ORANGE.LOCAL.QUEUE in die Warteschlange THIRD.QUEUE kopiert, aber es wird angegeben, dass die maximale Nachrichtenlänge in der neuen Warteschlange 1024 Bytes anstelle von 4194304 beträgt:

```
DEFINE QLOCAL (THIRD.QUEUE) +
LIKE (ORANGE.LOCAL.QUEUE) +
MAXMSGL(1024);
```

Zugehörige Verweise

[Warteschlangen definieren \(DEFINE\)](#)

Lokale Warteschlangenattribute mit ALTER QLOCAL oder DEFINE QLOCAL ändern

Sie können Warteschlangenattribute auf zwei Arten ändern, indem Sie entweder den MQSC-Befehl **ALTER QLOCAL** oder den MQSC-Befehl **DEFINE QLOCAL** mit dem Attribut **REPLACE** verwenden.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Informationen zu diesem Vorgang

Sie können mit dem Attribut **REPLACE** des Befehls **ALTER** und **DEFINE** eine vorhandene Definition durch die angegebene neue Definition ersetzen. Der Unterschied bei der Verwendung von **ALTER** und **DEFINE** besteht darin, dass **ALTER** mit **REPLACE** keine nicht angegebenen Parameter ändert, aber bei **DEFINE** mit **REPLACE** alle Parameter festgelegt werden.

Prozedur

- Um Ändern von Warteschlangenattributen verwenden Sie den Befehl **ALTER** oder **DEFINE**, wie in den folgenden Beispielen gezeigt wird.

In diesem Beispiel wird die maximale Nachrichtenlänge in der Warteschlange ORANGE.LOCAL.QUEUE auf 10.000 Byte verringert.

- 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. Wenn die Warteschlange beispielsweise zuvor unterdrückt wurde, wird dieser Status in aktiviert geändert, da dies der Standardwert ist, 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 Verweise

[ALTER-Warteschlangen](#)

[ALTER QLOCAL](#)

[Warteschlangen definieren \(DEFINE\)](#)

[QLOCAL DEFINIER](#)

Lokale Warteschlange mit CLEAR QLOCAL löschen

Mit dem MQSC-Befehl **CLEAR QLOCAL** können Sie den Inhalt einer lokalen Warteschlange löschen.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

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.

Informationen zu diesem Vorgang

Wenn Sie eine lokale Warteschlange mit dem Befehl **CLEAR QLOCAL** löschen möchten muss der Name der Warteschlange für den lokalen Warteschlangenmanager definiert sein.

Anmerkung: Es gibt keine Eingabeaufforderung, mit der Sie Ihre Meinung ändern können. Sobald Sie die Eingabetaste drücken, gehen die Nachrichten verloren.

Vorgehensweise

Um die Nachrichten aus einer lokalen Warteschlange zu löschen, geben Sie den Befehl **CLEAR QLOCAL** wie im folgenden Beispiel gezeigt ein.

In diesem Beispiel werden alle Nachrichten aus einer lokalen Warteschlange mit der Bezeichnung MAGENTA.QUEUE gelöscht:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

Zugehörige Verweise

[CLEAR QLOCAL](#)

Lokale Warteschlange mit DELETE QLOCAL löschen

Mit dem MQSC-Befehl **DELETE QLOCAL** können Sie eine lokale Warteschlange löschen.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Eine Warteschlange kann nicht gelöscht werden, wenn Sie nicht festgeschriebene Nachrichten enthält.

Wenn die Warteschlange eine oder mehrere festgeschriebene Nachrichten und keine nicht festgeschriebenen Nachrichten enthält, kann sie nur gelöscht werden, wenn Sie die Option **PURGE** angeben. Der Löschvorgang wird in diesem Fall selbst dann ausgeführt, wenn sich in der angegebenen Warteschlange festgeschriebene Nachrichten befinden, und diese Nachrichten werden ebenfalls gelöscht.

Wenn Sie **NOPURGE** anstelle von **PURGE** angeben, wird sichergestellt, dass die Warteschlange nicht gelöscht wird, wenn sie festgeschriebene Nachrichten enthält.

Prozedur

- Zum Löschen einer lokalen Warteschlange verwenden Sie den Befehl **DELETE QLOCAL** wie im folgenden Beispiel gezeigt.

In diesem Beispiel wird die Warteschlange PINK.QUEUE gelöscht, selbst wenn sich keine festgeschriebenen Nachrichten in der Warteschlange befinden:

```
DELETE QLOCAL (PINK.QUEUE) NOPURGE
```

In diesem Beispiel wird die die Warteschlange PINK.QUEUE gelöscht, selbst wenn sich festgeschriebene Nachrichten in der Warteschlange befinden:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

Zugehörige Verweise

[DELETE QLOCAL](#)

Warteschlangen mit dem Beispielprogramm durchsuchen

IBM MQ stellt einen Beispielwarteschlangenbrowser bereit, mit dem Sie den Inhalt der Nachrichten in einer Warteschlange anzeigen können.

Informationen zu diesem Vorgang

Der Browser wird sowohl in Quellenformaten und im Format einer ausführbaren Datei in den folgenden Positionen bereitgestellt, wobei *MQ_INSTALLATION_PATH* das übergeordnete Verzeichnis darstellt, 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\amqsbcg.exe
```

Linux **AIX** Auf AIX and Linux sind die Dateinamen und Pfade wie folgt:

Quelle

```
MQ_INSTALLATION_PATH/samp/amqsbcg0.c
```

Ausführbare Datei

```
MQ_INSTALLATION_PATH/samp/bin/amqsbcg
```

Prozedur

- Geben Sie zur Ausführung des Beispielprogramms einen Befehl ein, wie im folgenden Beispiel gezeigt wird.

Für das Beispielprogramm sind zwei Eingabeparameter, der Name der Warteschlange, in der die Nachrichten durchsucht werden sollen, sowie der Warteschlangenmanager, der Eigner dieser Warteschlange ist, erforderlich. For example:

```
amqsbcg SYSTEM.ADMIN.QMGREVENT.tpp01 saturn.queue.manager
```

Ergebnisse

Typische Ergebnisse dieses Befehls werden im folgenden Beispiel gezeigt:

```

AMQSBCG0 - starts here
*****

MQOPEN - 'SYSTEM.ADMIN.QMGR.EVENT'

MQGET of message number 1
****Message descriptor****

  StrucId : 'MD ' Version : 2
  Report  : 0 MsgType : 8
  Expiry  : -1 Feedback : 0
  Encoding : 546 CodedCharSetId : 850
  Format   : 'MQEVENT '
  Priority : 0 Persistence : 0
  MsgId   : X'414D512073617475726E2E71756575650005D30033563DB8'
  CorrelId : X'0000000000000000000000000000000000000000000000000000000000000000'
  BackoutCount : 0
  ReplyToQ      : '
  ReplyToQMgr   : 'saturn.queue.manager'
  ** Identity Context
  UserIdentifier : '
  AccountingToken :
  X'0000000000000000000000000000000000000000000000000000000000000000'
  ApplIdentityData : '
  ** Origin Context
  PutApplType      : '7'
  PutApplName     : 'saturn.queue.manager'
  PutDate         : '19970417' PutTime : '15115208'
  ApplOriginData  : '

  GroupId : X'0000000000000000000000000000000000000000000000000000000000000000'
  MsgSeqNumber    : '1'
  Offset          : '0'
  MsgFlags        : '0'
  OriginalLength  : '104'

**** Message ****

length - 104 bytes

00000000: 0700 0000 2400 0000 0100 0000 2C00 0000 '.....→.....'
00000010: 0100 0000 0100 0000 0100 0000 AE08 0000 '.....'
00000020: 0100 0000 0400 0000 4400 0000 DF07 0000 '.....D.....'
00000030: 0000 0000 3000 0000 7361 7475 726E 2E71 '...0...saturn.q'
00000040: 7565 7565 2E6D 616E 6167 6572 2020 2020 'ueue.manager'
00000050: 2020 2020 2020 2020 2020 2020 2020 2020 '
00000060: 2020 2020 2020 2020 '

No more messages
MQCLOSE
MQDISC

```

Zugehörige Verweise

[Das Beispielprogramm 'Browser'](#)

Große Warteschlangen aktivieren

IBM MQ unterstützt Warteschlangen mit mehr als 2 TB.

Windows Auf Windows-Systemen steht die Unterstützung für große Dateien ohne zusätzliche Unterstützung zur Verfügung.

Linux **AIX** Auf AIX and Linux -Systemen müssen Sie die Unterstützung großer Dateien explizit aktivieren, bevor Sie Warteschlangendateien mit mehreren Gigabyte oder Terabyte erstellen können. Weitere Informationen hierzu finden Sie in der Dokumentation zum Betriebssystem.

Einige Dienstprogramme, wie z. B. tar, können Dateien mit mehreren Gigabyte oder Terabyte nicht verarbeiten. 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 unter [MQ Performance documents](#) für plattformspezifische Leistungsberichte.

Sie können die Größe von Warteschlangendateien mithilfe eines neuen Attributs in lokalen und Modellwarteschlangen steuern. Weitere Informationen finden Sie unter [„IBM MQ-Warteschlangendateien ändern“](#) auf Seite 153.

Multi

IBM MQ-Warteschlangendateien ändern

Sie können die Größe von Warteschlangendateien mithilfe eines Attributs in lokalen Warteschlangen und Modellwarteschlangen steuern. Mithilfe der beiden Attribute für den Warteschlangenstatus können Sie die aktuelle Größe einer Warteschlangendatei anzeigen sowie die maximale Größe, die sie derzeit annehmen kann (auf Basis der Blockgröße, die aktuell in dieser Datei verwendet wird).

Attribut zum Ändern von Warteschlangendateien

Das Attribut in lokalen Warteschlangen und Modellwarteschlangen lautet folgendermaßen:

MAXFSIZE

Gibt die maximale Größe der Warteschlangendatei an, die von der Warteschlange verwendet wird (in Megabyte).

Sie können den Wert dieses Attributs mit MQSC-Befehlen, IBM MQ Explorer oder administrative REST API festlegen oder anzeigen. Sie können den Wert dieses Attributs auch im IBM MQ Console anzeigen.

Weitere Informationen finden Sie in [MAXFSIZE](#) und [„Größe einer IBM MQ-Warteschlangendatei ändern“](#) auf Seite 154.

Das PCF-Äquivalent dieses Attributs ist **MQIA_MAX_Q_FILE_SIZE**. Siehe [Change, Copy und Create Queue](#).

Die beiden Attribute für den Warteschlangenstatus lauten folgendermaßen:

CURFSIZE

Zeigt die aktuelle Größe der Warteschlangendatei in Megabyte an, gerundet auf das nächste Megabyte.

Sie können den Wert dieses Attributs mit MQSC-Befehlen, IBM MQ Explorer oder administrative REST API festlegen oder anzeigen.

Weitere Informationen finden Sie unter [CURFSIZE](#).

Das PCF-Äquivalent dieses Attributs ist **MQIA_CUR_Q_FILE_SIZE**. Siehe [Inquire Queue](#) und [Inquire Queue \(Antwort\)](#).

CURMAXFS

Gibt die aktuelle maximale Größe an, die die Warteschlangendatei erreichen kann, aufgerundet auf das nächsthöhere Megabyte, sofern die aktuelle Blockgröße für eine Warteschlange verwendet wird.

Sie können den Wert dieses Attributs mit MQSC-Befehlen, IBM MQ Explorer oder administrative REST API festlegen oder anzeigen.

Weitere Informationen finden Sie unter [CURMAXFS](#).

Das PCF-Äquivalent dieses Attributs ist **MQIA_CUR_MAX_FILE_SIZE**. Siehe [Inquire Queue](#) und [Inquire Queue \(Antwort\)](#).

Blockgröße und Granularität

Warteschlangendateien sind in Segmente unterteilt, die als Blöcke bezeichnet werden. Um die maximale Größe einer Warteschlangendatei zu erhöhen, muss die Blockgröße oder Granularität der Warteschlange möglicherweise vom Warteschlangenmanager geändert werden.

Wenn eine neu definierte Warteschlange mit einem großen Wert für **MAXFSIZE** erstellt wird, wird die Warteschlange mit einer geeigneten Blockgröße erstellt. Wenn jedoch der Wert für **MAXFSIZE** einer vorhandenen Warteschlange erhöht wird, z. B. durch Verwendung des MQSC-Befehls **ALTER QLOCAL**, kann es erforderlich sein, dass die Warteschlange geleert wird, damit der Warteschlangenmanager die Warteschlange neu konfigurieren kann.

Weitere Informationen finden Sie unter „Berechnung der Menge an Daten, die in einer IBM MQ-Warteschlangendatei gespeichert werden kann“ auf Seite 155.



Achtung: Für einige Dateisysteme und Betriebssysteme gibt es Begrenzungen für die Größe des gesamten Dateisystems oder für die Größe einer einzelnen Datei. Sie sollten überprüfen, welche Begrenzungen für die Systeme in Ihrem Unternehmen festgelegt sind.

Zugehörige Verweise

[ALTER QUEUES](#)

[ANZEIGEN QUEUE](#)

[ANZEIGEN QSTATUS](#)

Multi

Größe einer IBM MQ-Warteschlangendatei ändern

Sie können die maximale Größe einer Warteschlangendatei erhöhen oder verringern.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen:

- **ALW** An AIX, Linux, and Windows geben Sie MQSC-Befehle aus einem `runmqsc` Eingabeaufforderung. Sehen [Interaktives Ausführen von MQSC-Befehlen unter runmqsc](#) Und [Ausführen von MQSC-Befehlen aus Textdateien unter runmqsc](#).
- **IBM i** An IBM i erstellen Sie eine Liste von Befehlen in einer Skriptdatei und führen die Datei dann mit dem `STRMQMQSC` Befehl. Sehen [Administration mit MQSC-Befehlen auf IBM i](#).

Bevor Sie eine neue Größe für eine Warteschlangendatei festlegen, verwenden Sie den MQSC-Befehl `DISPLAY QLOCAL`, um die Größe der Warteschlangendatei anzuzeigen, die Sie ändern möchten. Geben Sie z. B. den folgenden Befehl aus:

```
DISPLAY QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) MAXFSIZE
```

Sie erhalten die folgende Ausgabe:

```
AMQ8409I: Display queue details
QUEUE(SYSTEM.DEFAULT.LOCAL.QUEUE)      TYPE(QLOCAL)
MAXFSIZE(DEFAULT)
```

Darin wird angezeigt, dass die maximale Größe der Warteschlangendatei dem Standardwert von 2.088.960 MB entspricht.

Informationen zu diesem Vorgang

In den folgenden Prozeduren finden Sie Informationen zu diesen Aktionen:

- Verringern der maximalen Größe, auf die eine Warteschlangendatei anwachsen kann.
- Erhöhen der maximalen Größe, auf die eine Warteschlangendatei anwachsen kann.



Achtung: Sie sollten beim Erhöhen der Größe von Warteschlangendateien unbedingt berücksichtigen, wie Ihre Anwendungen geschrieben sind und welche möglichen Auswirkungen die Erhöhung auf die Leistung haben kann. Der zufällige Zugriff auf Nachrichten in einer sehr großen Warteschlange kann sehr langsam sein.

Wenn Sie erwägen, die maximale Größe einer Warteschlangendatei über den Standardwert hinaus zu erhöhen, sollten Sie vorsichtig sein, wenn Sie Nachrichtenselektoren wie Korrelations-IDs und IBM MQ classes for JMS **JM 3.0** -oder [IBM MQ classes for Jakarta Messaging](#) -Selektorzeichenfolgen verwenden. Bei größeren Warteschlangendateien ist es günstiger, zuerst auf die Dateien zuzugreifen, die sich am längsten in der Warteschlange befinden.

Sehr große Mengen von Daten in einzelnen Warteschlangendateien sollten nur in Warteschlangenmanagern verwendet werden, die für die Umlaufprotokollierung konfiguriert sind oder in denen die Medienaufzeichnung nicht für die einzelne Warteschlange aktiviert wurde.

Sie sollten die Größe von SYSTEM-Warteschlangen nicht begrenzen, da dies Auswirkungen auf den Betrieb des Warteschlangenmanagers haben könnte.

Vorgehensweise

1. Verringern der maximale Größe der Warteschlangendatei

- a) Geben Sie den folgenden MQSC-Befehl aus, um eine lokale Datei mit dem Namen SMALLQUEUE mit einer Größe von 500 Gigabyte zu erstellen:

```
DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
2 : DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
AMQ8006I: IBM MQ queue created
```

Sie erhalten die folgende Nachricht: AMQ8006I:

Anmerkung: Wenn Sie eine Warteschlange erstellen, deren Wert kleiner ist als die Menge der Daten, die sich bereits in der Datei befinden, können keine neuen Nachrichten in die Warteschlange eingereicht werden.

Wenn eine Anwendung versucht, eine Nachricht in eine Warteschlangendatei einzureihen, die nicht über ausreichend Speicherplatz verfügt, empfängt die Anwendung den Rückgabecode MQRC_Q_SPACE_NOT_AVAILABLE. Wenn ausreichend Nachrichten aus der Warteschlange gelesen und anschließend gelöscht werden, können Anwendungen wieder neue Nachrichten in die Warteschlange einreihen.

2. Erhöhen der maximalen Größe der Warteschlangendatei

- a) Geben Sie den folgenden MQSC-Befehl aus, um eine lokale Datei mit dem Namen LARGEQUEUE mit einer Größe von 5 Terabyte zu erstellen:

```
DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
3 : DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
AMQ8006I: IBM MQ queue created
```

Multi Berechnung der Menge an Daten, die in einer IBM MQ-Warteschlangendatei gespeichert werden kann

Die Menge der Daten, die in einer Warteschlange gespeichert werden kann, ist auf die Größe der einzelnen Blöcke begrenzt, in die die Warteschlange aufgeteilt ist. Mit MQSC-Befehlen können Sie die Blockgröße und Granularität bestätigen und die Größe einer Warteschlangendatei überprüfen.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen:

- **ALW** An AIX, Linux, and Windows geben Sie MQSC-Befehle aus einem `runmqsc` Eingabeaufforderung. Sehen [Interaktives Ausführen von MQSC-Befehlen unter runmqsc](#) und [Ausführen von MQSC-Befehlen aus Textdateien unter runmqsc](#).
- **IBM i** An IBM i erstellen Sie eine Liste von Befehlen in einer Skriptdatei und führen die Datei dann mit dem `STRMQMQSC` Befehl. Sehen [Administration mit MQSC-Befehlen auf IBM i](#).

Prozedur

- Bestätigen Sie die Blockgröße und Granularität.

Die Standardblockgröße beträgt 512 Byte. Zur Unterstützung von Warteschlangendateien, die größer als zwei Terabyte sind, muss der Warteschlangenmanager die Blockgröße erhöhen.

Die Blockgröße wird automatisch berechnet, wenn Sie den Wert für `MAXFSIZE` für eine Warteschlange konfiguriert wird, aber die überarbeitete Blockgröße nicht auf die Warteschlange angewendet werden kann, wenn die Warteschlange bereits Nachrichten enthält. Sobald eine Warteschlange leer ist, ändert der Warteschlangenmanager automatisch die Blockgröße, um die konfigurierte **MAXFSIZE** zu unterstützen.

Der Befehl **DISPLAY QSTATUS** hat das neue Attribut `CURMAXFS`, mit dem Sie bestätigen können, dass eine Warteschlange für die Verwendung einer neuen Blockgröße geändert wurde.

Im folgenden Beispiel wird mit dem Wert 4177920 für **CURMAXFS** bestätigt, dass die Warteschlangendatei derzeit auf etwa vier Terabyte anwachsen kann. Wenn der Wert von **MAXFSIZE**, der in der Warteschlange konfiguriert ist, größer ist als der Wert von **CURMAXFS**, wartet der Warteschlangenmanager weiterhin darauf, dass die Warteschlange geleert wird, bevor er die Blockgröße der Warteschlangendatei rekonfiguriert.

```
DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
  2 : DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
AMQ8450I: Display queue status details
  QUEUE(LARGEQUEUE)                                TYPE(QUEUE)
  CURMAXFS(4177920)                                CURDEPTH(100000)
```

- Überprüfen Sie die Größe einer Warteschlangendatei.

Mit dem Attribut `CURFSIZE` des Befehls **DISPLAY QSTATUS** können Sie die aktuelle Größe einer Warteschlangendatei auf der Platte in Megabyte anzeigen. Dies kann auf Plattformen wie beispielsweise der IBM MQ Appliance hilfreich sein, auf denen nicht direkt auf das Dateisystem zugegriffen werden kann.

```
DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
  1 : DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
AMQ8450I: Display queue status details
  QUEUE(SMALLQUEUE)                                TYPE(QUEUE)
  CURDEPTH(4024)                                    CURFSIZE(10)
```

Anmerkung: Wenn Nachrichten aus einer Warteschlange entfernt werden, nimmt das Attribut **CURFSIZE** nicht sofort ab.

Normalerweise wird Speicherplatz in einer Warteschlangendatei nur freigegeben, wenn keine Anwendungen die Warteschlange geöffnet haben und keine unbestätigten Nachrichten in der Warteschlange gespeichert sind. Jede erforderliche Abschneidung oder Komprimierung einer Warteschlangendatei, die vom Warteschlangenmanager geladen wird, erfolgt während des Prüfpunkts, während der Beendigung des Warteschlangenmanagers oder während der Aufzeichnung eines Datenträgerimages der Warteschlange.

Zugehörige Verweise

[ALTER QUEUES](#)

[ANZEIGEN QSTATUS](#)

Mit fernen Warteschlangen arbeiten

Eine ferne Warteschlange ist eine Definition in einem lokalen Warteschlangenmanager, die auf eine Warteschlange in einem fernen Warteschlangenmanager verweist. Sie müssen keine ferne Warteschlange von einer lokalen Position aus definieren, aber wenn Sie dies tun, können Anwendungen auf die ferne Warteschlange mit ihrem lokal definierten Namen verweisen, anstatt einen Namen anzugeben, der durch die ID des Warteschlangenmanagers qualifiziert ist, auf dem sich die ferne Warteschlange befindet.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Informationen zu diesem Vorgang

Eine Anwendung stellt eine Verbindung zu einem lokalen WS-Manager her und gibt dann einen MQOPEN -Aufruf aus. Im geöffneten Aufruf ist der angegebene Warteschlangenname 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.

Prozedur

- Eine Nachricht in eine Warteschlange einreihen, deren Eigner ein ferner Warteschlangenmanager 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.

- Nachrichten in eine ferne Warteschlange einreihen (alternative Methode)

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.

- Verwenden Sie andere Befehle mit fernen Warteschlangen.

Sie können MQSC-Befehle verwenden, um die Attribute eines fernen Warteschlangenobjekts anzuzeigen oder zu ändern, oder Sie können das ferne Warteschlangenobjekt löschen. For example:

- 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.

Definitionen ferner Warteschlangen können als Aliasnamen verwendet werden

Zusätzlich zum Lokalisieren einer Warteschlange in einem anderen Warteschlangenmanager können Sie auch eine lokale Definition einer fernen Warteschlange für Warteschlangenmanager-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 Aliasnamen für Warteschlangenmanager 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 Antwortwarteschlangennamen 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](#).

Mit Aliaswarteschlangen arbeiten

Sie können eine Aliaswarteschlange definieren, um indirekt auf eine andere Warteschlange oder ein anderes Thema zu verweisen.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).



Achtung: Verteilerlisten unterstützen nicht die Verwendung von Aliaswarteschlangen, die auf Themenobjekte verweisen. Wenn eine Aliaswarteschlange auf ein Topic-Objekt in einer Verteilerliste verweist, gibt IBM MQ MQRC_ALIAS_BASE_Q_TYPE_ERROR zurück.

Informationen zu diesem Vorgang

Für die Warteschlange, auf die sich eine Aliaswarteschlange bezieht, kann es sich um eine der folgenden Warteschlangen handeln:

- Eine lokale Warteschlange (siehe [„Lokale Warteschlange mit DEFINE QLOCAL definieren“](#) auf Seite [146](#)).
- Eine lokale Definition einer fernen Warteschlange (siehe [„Mit fernen Warteschlangen arbeiten“](#) auf Seite [156](#)).
- 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 TARGET ändern, können Sie MQI-Aufrufe an eine andere Warteschlange umleiten, möglicherweise auf einem anderen Warteschlangenmanager. Dies ist hilfreich bei der Wartung, Migration und Lastverteilung.

Prozedur

- Definieren Sie eine Aliaswarteschlange.

Der folgende MQSC-Befehl erstellt eine Aliaswarteschlange:

```
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. For example:

```
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.

Der folgende MQSC-Befehl definiert einen Alias, der für die Anwendung ALPHA aktiviert und 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 Warteschlangenalienamen auf die gleiche Weise definieren, wie Sie diese Attribute mit lokalen Warteschlangen verwenden.

- Verwenden Sie andere Befehle mit Aliaswarteschlangen.

Sie können die entsprechenden MQSC-Befehle verwenden, um Aliaswarteschlangenattribute anzuzeigen oder zu ändern oder um das Aliaswarteschlangenobjekt zu löschen. For example:

Verwenden Sie den Befehl **DISPLAY QALIAS**, um die Attribute der Aliaswarteschlange anzuzeigen:

```
DISPLAY QALIAS (ALPHAS.ALIAS.QUEUE)
```

Mit dem Befehl **ALTER QALIAS** können Sie den Basiswarteschlangennamen ändern, in den der Aliasname aufgelöst wird, wobei die Option `force` die Änderung erzwingt, auch wenn die Warteschlange geöffnet ist:

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGET(ORANGE.LOCAL.QUEUE) FORCE
```

Verwenden Sie den Befehl **DELETE QALIAS**, um diesen Warteschlangenalienamen zu löschen:

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Sie können eine Aliaswarteschlange nicht löschen, wenn eine Anwendung momentan die Warteschlange geöffnet hat.

Zugehörige Konzepte

[Verteilerlisten](#)

Zugehörige Verweise

[ALTER QALIAS](#)

[DEFINE QALIAS](#)

[DELETE QALIAS](#)

Mit Modellwarteschlangen arbeiten

Modellwarteschlangen stellen eine komfortable Methode für Anwendungen zur Verfügung, um Warteschlangen nach Bedarf zu erstellen.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Informationen zu diesem Vorgang

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.

Prozedur

- Definieren Sie eine Modellwarteschlange.

Mit dem MQSC-Befehl **DEFINE QMODEL** können Sie eine Modellwarteschlange mit einer Gruppe von Attributen auf dieselbe Weise wie 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.) For example:

```
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.

- Verwenden Sie andere Befehle mit Modellwarteschlangen.

Sie können die entsprechenden MQSC-Befehle verwenden, um die Attribute einer Modellwarteschlange anzuzeigen oder zu ändern oder um das Modellwarteschlangenobjekt zu löschen. For example:

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 zu ändern und Einreihungen in jede dynamische Warteschlange zu ermöglichen, die aus diesem Modell erstellt wurde:

```
ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)
```

Verwenden Sie den Befehl **DELETE QMODEL** , um diese Modellwarteschlange zu löschen:

```
DELETE QMODEL (RED.MODEL.QUEUE)
```

Zugehörige Verweise

[ALTER QMODEL](#)

[QMODEL DEFINIER](#)

[DELETE QMODEL](#)

[ANZEIGEN QUEUE](#)

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.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Informationen zu diesem Vorgang

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.

Prozedur

- Informieren Sie den Warteschlangenmanager über die Warteschlange für nicht zustellbare Nachrichten.

To do this, specify a dead-letter queue name on the **crtmqm** command (**crtmqm -u DEAD.LETTER.QUEUE**, for example), or by using the **DEADQ** attribute on the **ALTER QMGR** command to specify one later. Sie müssen die Warteschlange für den dead-letter definieren, bevor Sie sie verwenden.

- Geben Sie an, wie Nachrichten in einer Warteschlange für nicht zustellbare Nachrichten verarbeitet werden sollen.

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](#).

Mit der Steuerroutine der IBM MQ -Warteschlange für nicht zustellbare Nachrichten können Sie angeben, wie Nachrichten, die in einer Warteschlange für nicht zustellbare Nachrichten gefunden werden, verarbeitet oder entfernt werden. Siehe [„Nachrichten in einer IBM MQ-Warteschlange für nicht zustellbare Nachrichten verarbeiten“](#) auf Seite 164.

Zugehörige Konzepte

[Warteschlangen für nicht zustellbare Nachrichten](#)

Zugehörige Tasks

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

Zugehörige Verweise

[ALTER QMGR](#)

[crtmqm \(WS-Manager erstellen\)](#)

Nachrichten in einer IBM MQ-Warteschlange für nicht zustellbare Nachrichten verarbeiten

Zur Verarbeitung von Nachrichten in einer Warteschlange für nicht zustellbare Nachrichten (Dead-Letter Queue, DLQ) verwenden Sie den Standard-DLQ-Handler, der von IBM MQ bereitgestellt wird. Der Handler gleicht Nachrichten in der DLQ mit Einträgen in einer Regeltabelle ab, die Sie definieren.

Informationen zu diesem Vorgang

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 *Steueroutine der Warteschlange für nicht zustellbare Nachrichten* (DLQ-Steueroutine) bezeichnet wird, die Sie mit dem MQSC-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 Tasks

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

Zugehörige Verweise

[Warteschlangen für nicht zustellbare Nachrichten](#)

Steueroutine der Warteschlange für nicht zustellbare Nachrichten aufrufen

Rufen Sie die Steueroutine der Warteschlange für nicht zustellbare Nachrichten (DLQ) mit dem Steuerbefehl **runmqdlq** auf. Sie können die DLQ, die Sie verarbeiten möchten, und den Warteschlangenmanager, den Sie auf zwei Arten verwenden möchten, benennen.

Vorbereitende Schritte

Um den DLQ-Handler ausführen zu können, müssen Sie 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 Nachrichten in Warteschlangen mit der Berechtigung der Benutzer-ID im Nachrichtenkontext einreicht, müssen Sie auch berechtigt sein, die Identität anderer Benutzer zu übernehmen.

Informationen zu diesem Vorgang

Die folgenden Beispiele gelten für die DLQ ABC1.DEAD.LETTER.QUEUE, deren Eigner der Warteschlangenmanager 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.

Weitere Informationen zum Befehl **runmqdlq** finden Sie im Abschnitt [runmqdlq](#).

Prozedur

- Sie können die DLQ und den Warteschlangenmanager als Parameter für den Befehl **runmqdlq** benennen.

Beispiel in der Eingabeaufforderung:

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrule.rul
```

- Sie können die DLQ und den Warteschlangenmanager in der Regeltabelle benennen.

For example:

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

Zugehörige Konzepte

[Warteschlangen für nicht zustellbare Nachrichten](#)

Zugehörige Tasks

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

  *Beispiel-DLQ-Handler **amqsd1q***

Zusätzlich zu dem mit dem Befehl **runmqdlq** aufgerufenen DLQ-Handler stellt IBM MQ die Quelle eines Beispiel-DLQ-Handlers namens **amqsd1q** mit einer Funktion zur Verfügung, die der von **runmqdlq** bereitgestellten Funktion ähnelt.

Sie können **amqsd1q** anpassen, um einen DLQ-Handler bereitzustellen, der Ihren Anforderungen entspricht. Sie können beispielsweise entscheiden, dass ein DLQ-Handler Nachrichten ohne Header von nicht zustellbaren Nachrichten verarbeiten kann. (Sowohl der Standard-DLQ-Handler als auch das Beispiel **amqsd1q** verarbeiten nur die Nachrichten in der DLQ, die mit einem Header für nicht zustellbare 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 **amqsd1q** 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 **amqsd1q** im folgenden Verzeichnis bereitgestellt:

```
MQ_INSTALLATION_PATH/samp/d1q
```

und die kompilierte Version wird im folgenden Verzeichnis bereitgestellt:

`MQ_INSTALLATION_PATH/samp/bin`

Eine erstellte Version des Beispielprogramms namens **amqsd1qc** ist enthalten. Damit können Sie im Clientmodus eine Verbindung zu einem fernen Warteschlangenmanager herstellen. Für die Verwendung von **amqsd1qc** müssen Sie eine der Umgebungsvariablen `MQSERVER` einrichten, **MQCHLLIB** oder **MQCHLTAB** festlegen, um anzugeben, wie eine Verbindung zum Warteschlangenmanager hergestellt werden soll. For example:

```
export MQSERVER="SYSTEM.DEF.SVRCONN/TCP/myappliance.co.uk(1414) "
```

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 Konzepte

[Warteschlangen für nicht zustellbare Nachrichten](#)

Zugehörige Tasks

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

DLQ-Steuerdaten

Sie können Schlüsselwörter in einen Steuerdateneintrag (control-data) in eine Regeltabelle für einen DLQ-Handler (Steueroutine 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* | " (Standardwert))

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* | " (Standardwert))

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 (Intervall| 60 (Standardwert))

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 (Standardwert) |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

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|* (Standardwert))

Der *ApplIdentityData*-Wert, der im Nachrichtendeskriptor MQMD der Nachricht in der Warteschlange für nicht zuorden_Nachrichten angegeben ist.

APPLNAME (PutAppName|* (Standardwert))

Der Name der Anwendung, die den Aufruf MQPUT oder MQPUT1 ausgegeben hat, wie im Feld *PutAppName* des Nachrichtendeskriptors, MQMD, der Nachricht in der DLQ angegeben ist.

APPLTYPE (PutApplTyp|* (Standardwert))

Der *PutApplType*-Wert, der im Nachrichtendeskriptor MQMD der Nachricht in der DLQ angegeben ist.

DESTQ (QueueName|* (Standardwert))

Der Name der Nachrichtenwarteschlange, für die die Nachricht bestimmt ist.

DESTQM (*QueueManagerName* | * (Standardwert))

Der Name des Warteschlangenmanagers für die Nachrichtenwarteschlange, für die die Nachricht bestimmt ist.

FEEDBACK (*Feedback* | * (Standardwert))

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* | * (Standardwert))

Der Name, den der Absender der Nachricht verwendet, um das Format der Nachrichtendaten zu beschreiben.

MSGTYPE (*MsgType* | * (Standardwert))

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* | * (Standardwert))

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* | * (Standardwert))

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* | * (Standardwert))

Der Name der Warteschlange für Antwortnachrichten, die im Nachrichtendeskriptor MQMD der Nachricht in der Warteschlange für DLQ angegeben ist.

REPLYQM (*QueueManagerName* | * (Standardwert))

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* | * (Standardwert))

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 Akti-

on 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 (*Warteschlangenname* | & DESTQ | & REPLYQ)

Der Name der Nachrichtenwarteschlange, an die die Nachricht weitergeleitet werden soll, wenn `ACTION` (FWD) angefordert wird.

QueueName

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 | " (Standardwert))

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 (Standardwert) | 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 (Standardwert) | 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.

RETRY (*Wiederholungszähler* | 1 (Standardwert))

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-Steuerroutinen; die Anzahl bleibt nicht über Neustarts hinweg bestehen. Wenn der DLQ-Handler erneut gestartet wird, wird die Anzahl der Versuche, eine Regel anzuwenden 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 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.

For example:

```
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 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

- 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_CONTEXT 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 Konzepte

Warteschlangen für nicht zustellbare Nachrichten

Zugehörige Tasks

Fehlerbehebung bei nicht zugestellten Nachrichten



Eine Beispieltabelle für eine Warteschlange für nicht zugestellte 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 Konzepte

[Warteschlangen für nicht zustellbare Nachrichten](#)

Zugehörige Tasks

[Fehlerbehebung bei nicht zugestellten Nachrichten](#)

Zugehörige Verweise

[runmqdlq \(Steerroutine der Warteschlange für nicht zustellbare Nachrichten ausführen\)](#)

Steerroutine der Warteschlange für nicht zustellbare Nachrichten unter IBM i aufrufen

Unter IBM MQ for IBM irufen Sie den DLQ-Handler mit dem Befehl **STRMQMDLQ** auf.

Vorbereitende Schritte

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.

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 163.

Informationen zu diesem Vorgang

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.

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 Anwendungen, die Nachrichten in die DLQ einreihen, müssen Sie einen MQDLH bereitstellen.

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, die als *Steerroutine der Warteschlange für nicht zustellbare Nachrichten* (DLQ-Steerroutine) bezeichnet wird, bereit, die Sie mit dem **STRMQMDLQ**-Befehl 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.

Prozedur

- 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. For example:

```
STRMQMDLQ UDLMSGQ(ABC1.DEAD.LETTER.QUEUE) SRCMBR(QRULE) SRCFILE(library/QXTSRC)
MQMNAME(MY.QUEUE.MANAGER)
```

- In der Regeltabelle. For example:

```
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.

Zugehörige Konzepte

[Warteschlangen für nicht zustellbare Nachrichten](#)

Zugehörige Tasks

[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* | " (Standardwert))

Der Name der DLQ, die verarbeitet werden soll:

1. Jeder UDLMMSGQ-Wert, den Sie als Parameter für den Befehl **STRMQMDLQ** angeben, überschreibt jeden INPUTQ-Wert 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* | " (Standardwert))

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 (*Intervall* | 60 (Standardwert))

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 (Standardwert) | 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.

NEIN

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** angeben, als Alternative zum Einschließen von Steuerdaten in die Regeltabelle. Wenn ein Wert sowohl in der Regeltabelle als auch bei 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. 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. 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).

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*|* (Standardwert))

Der *ApplIdentityData*-Wert der Nachricht in der DLQ, die im Nachrichtendeskriptor (MQMD) angegeben ist.

APPLNAME (*PutAppName*|* (Standardwert))

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 (*PutApplTyp*|* (Standardwert))

Der *PutApplType*-Wert, der im Nachrichtendeskriptor MQMD der Nachricht in der DLQ angegeben ist.

DESTQ (*QueueName*|* (Standardwert))

Der Name der Nachrichtenwarteschlange, für die die Nachricht bestimmt ist.

DESTQM (*QueueManagerName*|* (Standardwert))

Der Name des Warteschlangenmanagers für die Nachrichtenwarteschlange, für die die Nachricht bestimmt ist.

FEEDBACK (*Feedback*|* (Standardwert))

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*|* (Standardwert))

Der Name, den der Absender der Nachricht verwendet, um das Format der Nachrichtendaten zu beschreiben.

MSGTYPE (*MsgType*|* (Standardwert))

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*|* (Standardwert))

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*|* (Standardwert))

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 zugebenden_Nachrichten gestellt wurden, weil ihre Zielwarteschlangen voll waren.

REPLYQ (*QueueName* | * (Standardwert)

Der Name der Antwortwarteschlange, die im Nachrichtendeskriptor MQMD der Nachricht in der DLQ angegeben ist.

REPLYQM (*QueueManagerName* | * (Standardwert)

Der Name des Warteschlangenmanagers für die Warteschlange für Antwortwarteschlangen, die im Schlüsselwort REPLYQ angegeben ist.

USERID (*UserIdentifier* | * (Standardwert)

Die Benutzer-ID des Benutzers, der die Nachricht in der DLQ erstellt hat, wie im Nachrichtendeskriptor MQMD angegeben.

 **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.

QueueName

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 | " (Standardwert))

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 (Standardwert) |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 (Standardwert) | 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.

RETRY (Wiederholungszähler|1 (Standardwert))

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.

For example:

```
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 Bedeutung. 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 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

- 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.

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 Fehlernachrichten 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 Nachrichtendesktors 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 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

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)
```

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. Stellen Sie sicher, dass die DLQ möglichst wenige Nachrichten enthält.

Informationen zu diesem Vorgang

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.

Aus diesen Gründen sollten Sie sicherstellen, dass die DLQ möglichst wenige Nachrichten 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.

Gehen Sie wie folgt vor, um den DLQ-Handler zum Leeren der DLQ zu aktivieren:

Prozedur

- Verwenden Sie für Nachrichten, die Sie andernfalls ignorieren würden, eine Aktion, die die Nachrichten in eine andere Warteschlange verschiebt.

Versuchen Sie, den Befehl **ACTION (IGNORE)** nicht zu verwenden, der Nachrichten in der DLQ belässt- und denken Sie daran, dass **ACTION (IGNORE)** für Nachrichten angenommen wird, die nicht explizit von anderen Regeln in der Tabelle adressiert werden. Verwenden Sie stattdessen eine Aktion, die die Nachrichten in eine andere Warteschlange verschiebt. For example:

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

- Legen Sie die letzte Regel in der Tabelle als "catch-all" fest, um Nachrichten zu verarbeiten, die von früheren Regeln in der Tabelle nicht berücksichtigt 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.

Mit Verwaltungsthemen arbeiten

Verwenden Sie MQSC-Befehle, um Verwaltungsthemen zu verwalten.

Detaillierte Informationen zu diesen Befehlen finden Sie in den [MQSC-Befehlen](#) .

Zugehörige Konzepte

[Verwaltungsthemenobjekte](#)

Zugehörige Tasks

[„Verwaltungsthema definieren“ auf Seite 184](#)

Mit dem MQSC-Befehl **DEFINE TOPIC** können Sie ein Verwaltungsthema erstellen. Wenn Sie ein Verwaltungsthema definieren, können Sie optional jedes Themenattribut festlegen.

[„Attribute des Verwaltungsthemenobjekts anzeigen“ auf Seite 185](#)

Mit dem MQSC-Befehl **DISPLAY TOPIC** können Sie ein Verwaltungsthemenobjekt anzeigen.

[„Verwaltungsthemenattribute ändern“ auf Seite 186](#)

Sie können Themenattribute auf zwei Arten ändern: mit dem Befehl **ALTER TOPIC** oder mit dem Befehl **DEFINE TOPIC** und dem Attribut **REPLACE** .

[„Eine Verwaltungsthemendefinition kopieren“ auf Seite 186](#)

Sie können eine Themendefinition mit dem Attribut LIKE im Befehl **DEFINE** kopieren.

[„Löschen einer Verwaltungsthemendefinition“ auf Seite 187](#)

Mit dem MQSC-Befehl **DELETE TOPIC** können Sie ein Verwaltungsthema löschen.

Verwaltungsthema definieren

Mit dem MQSC-Befehl **DEFINE TOPIC** können Sie ein Verwaltungsthema erstellen. Wenn Sie ein Verwaltungsthema definieren, können Sie optional jedes Themenattribut festlegen.

Vorbereitende Schritte

Anmerkung: Das Beispiel in dieser Task erfordert die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Informationen zu diesem Vorgang

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.

Beispiel

Der folgende Befehl **DEFINE TOPIC** definiert beispielsweise ein Thema namens 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. Weitere Informationen hierzu finden Sie im Abschnitt „Attribute des Verwaltungsthemenobjekts anzeigen“ auf Seite 185.
- 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 186.

Zugehörige Verweise

[TOPIC DEFINI](#)

Attribute des Verwaltungsthemenobjekts anzeigen

Mit dem MQSC-Befehl **DISPLAY TOPIC** können Sie ein Verwaltungsthemenobjekt anzeigen.

Vorbereitende Schritte

Anmerkung: Die Beispiele in dieser Task erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Beispiel

Dieser Befehl zeigt alle Themen an:

```
DISPLAY TOPIC(ORANGE.TOPIC)
```

Sie können Attribute selektiv anzeigen, indem Sie sie einzeln mit dem Befehl **DISPLAY TOPIC** angeben. For example:

```
DISPLAY TOPIC(ORANGE.TOPIC) +  
TOPICSTR +  
DEFPRTY +  
NPMSGDLV
```

Dieser Befehl zeigt die drei angegebenen Attribute an:

```
AMQ8633: Display topic details.  
TOPIC(ORANGE.TOPIC)                                TYPE(LOCAL)  
TOPICSTR(ORANGE)                                    DEFPRTY(ASPARENT)  
NPMSGDLV(ASPARENT)
```

Verwenden Sie den Befehl **DISPLAY TPSTATUS**, um die ASPARENT-Werte des Themas anzuzeigen, wie sie zur 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 Verweise

[ANZEIGEN TOPIC](#)

[ANZEIGEN TPSTATUS](#)

Verwaltungsthemenattribute ändern

Sie können Themenattribute auf zwei Arten ändern: mit dem Befehl **ALTER TOPIC** oder mit dem Befehl **DEFINE TOPIC** und dem Attribut **REPLACE**.

Vorbereitende Schritte

Anmerkung: Die Beispiele in dieser Task erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Beispiel

Wenn Sie beispielsweise die Standardpriorität von Nachrichten ändern wollen, die einem Thema mit dem Namen ORANGE.TOPIC einen der folgenden Befehle, um 5 zu sein:

- 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 veröffentlichenden Anwendung bereitgestellt wird.

Zugehörige Verweise

[ALTER TOPIC](#)

[ANZEIGEN TOPIC](#)

Eine Verwaltungsthemendefinition kopieren

Sie können eine Themendefinition mit dem Attribut LIKE im Befehl **DEFINE** kopieren.

Vorbereitende Schritte

Anmerkung: Die Beispiele in dieser Task erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Beispiel

Mit dem folgenden Befehl wird das Thema MAGENTA.TOPIC mit denselben Attributen wie das ursprüngliche Thema, ORANGE.TOPIC anstelle des Standardverwaltungsthemas des Systems. 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.

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

Sie können diese Form des Befehls **DEFINE** auch verwenden, um eine Themendefinition zu kopieren, aber Änderungen an den Attributen des Originals vorzunehmen. For example:

```
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. For example:

```
DEFINE TOPIC (GREEN.TOPIC) +  
TOPICSTR (GREEN) +  
LIKE (BLUE.TOPIC) +  
USEDLQ (NO)
```

Zugehörige Verweise

[TOPIC DEFINI](#)

Löschen einer Verwaltungsthemendefinition

Mit dem MQSC-Befehl **DELETE TOPIC** können Sie ein Verwaltungsthema löschen.

Vorbereitende Schritte

Anmerkung: Das Beispiel in dieser Task erfordert die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

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 Verweise

[DELETE TOPIC](#)

Mit Subskriptionen arbeiten

Verwenden Sie MQSC-Befehle zum Verwalten von Subskriptionen.

Informationen zu diesem Vorgang

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](#).

Verwaltungssubskription definieren

Mit dem MQSC-Befehl **DEFINE SUB** können Sie eine Verwaltungssubskription 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.

Vorbereitende Schritte

Anmerkung: Die Beispiele in dieser Task erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Beispiel

Der folgende **DEFINE SUB** -Befehl definiert eine Subskription mit dem Namen ORANGE mit den folgenden Merkmalen:

- Dauerhafte Subskription, d. d. sie bleibt beim Neustart des Warteschlangenmanagers bestehen, mit unbegrenztem Ablaufdatum.
- Empfangen von Veröffentlichungen, die in der Themenzeichenfolge ORANGE erfolgen, mit den Nachrichtenprioritäten, die von den Veröffentlichungsanwendungen festgelegt werden.
- Veröffentlichungen, die für diese Subskription bereitgestellt werden, werden an die lokale Warteschlange SUBQ gesendet. Diese Warteschlange muss vor der Definition der Subskription definiert werden.

```
DEFINE SUB (ORANGE) +  
TOPICSTR (ORANGE) +  
DESTCLAS (PROVIDED) +  
DEST (SUBQ) +  
EXPIRY (UNLIMITED) +  
PUBPRTY (AS PUB)
```

Anmerkung:

- Die Subskription und der Name der Themenzeichenfolge müssen nicht übereinstimmen.
- Mit Ausnahme der Werte für die Ziel- und 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. Weitere Informationen hierzu finden Sie im Abschnitt „Attribute von Subskriptionen anzeigen“ auf Seite 189.
- Wenn bereits eine lokale Subskription für denselben WS-Manager mit dem Namen ORANGE vorhanden ist, schlägt dieser Befehl fehl. Verwenden Sie das Attribut **REPLACE**, wenn Sie die vorhandene Definition einer Warteschlange überschreiben wollen, aber siehe auch „Attribute für lokale Subskription ändern“ auf Seite 190.

- Wenn die Warteschlange SUBQ nicht vorhanden ist, schlägt dieser Befehl fehl.

Zugehörige Verweise

[SUB DEFINI](#)

Attribute von Subskriptionen anzeigen

Mit dem Befehl **DISPLAY SUB** können Sie konfigurierte Attribute jeder Subskription anzeigen, die dem Warteschlangenmanager bekannt ist.

Vorbereitende Schritte

Anmerkung: Die Beispiele in dieser Task erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Beispiel

```
DISPLAY SUB(ORANGE)
```

Sie können Attribute selektiv anzeigen, indem Sie sie einzeln angeben. For example:

```
DISPLAY SUB(ORANGE) +  
SUBID +  
TOPICSTR +  
DURABLE
```

Mit diesem Befehl werden die drei angegebenen Attribute wie folgt angezeigt:

```
AMQ8096: IBM MQ subscription inquired.  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
SUB(ORANGE) TOPICSTR(ORANGE)  
DURABLE(YES)
```

TOPICSTR ist die aufgelöste Themenzeichenfolge, auf der dieser Subskribent ausgeführt wird. Wenn eine Subskription definiert ist, um ein Topic-Objekt zu verwenden, wird die Themenzeichenfolge aus diesem Objekt als Präfix für die Themenzeichenfolge verwendet, die beim Herstellen der Subskription angegeben wird. SUBID ist eine eindeutige Kennung, die vom WS-Manager bei der Erstellung einer Subskription zugeordnet wird. Dies ist ein nützliches Attribut, das angezeigt werden kann, weil einige Subskriptionsnamen lang sein können oder in einem anderen Zeichensatz enthalten sind, für den sie möglicherweise unpraktisch werden.

Eine alternative Methode zum Anzeigen von Subskriptionen ist die Verwendung der SUBID:

```
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 Verweise

[ANZEIGEN SUB](#)

Attribute für lokale Subskription ändern

Sie können Subskriptionsattribute auf zwei Arten ändern: mit dem Befehl **ALTER SUB** oder dem Befehl **DEFINE SUB** mit dem Attribut **REPLACE**.

Vorbereitende Schritte

Anmerkung: Die Beispiele in dieser Task erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Beispiel

Wenn Sie die Priorität von Nachrichten, die einer Subskription mit dem Namen `ORANGE` zugestellt werden, in 5 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 Verweise

[ALTER SUB](#)

[SUB DEFINI](#)

Lokale Subskriptionsdefinition kopieren

Sie können eine Subskriptionsdefinition mit dem Attribut **LIKE** im Befehl **DEFINE** kopieren.

Vorbereitende Schritte

Anmerkung: Die Beispiele in dieser Task erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Beispiel

```
DEFINE SUB(BLUE) +  
      LIKE(ORANGE)
```

Sie können auch die Attribute der untergeordneten REAL in die untergeordnete THIRD.SUB kopieren und angeben, dass die **correlID** der bereitgestellten Veröffentlichungen DRITTE und nicht die Publisher **correlID** ist. For example:

```
DEFINE SUB(THIRD.SUB) +  
      LIKE(BLUE) +  
      DESTCURL(ORANGE)
```

Zugehörige Verweise

[SUB DEFINI](#)

Lokale Subskription löschen

Mit dem MQSC-Befehl **DELETE SUB** können Sie eine lokale Subskription löschen.

Vorbereitende Schritte

Anmerkung: Die Beispiele in dieser Task erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Beispiel

```
DELETE SUB(ORANGE)
```

Sie können eine Subskription auch mit der SUBID löschen:

```
DELETE SUB SUBID(414D5120414141202020202020202020EE921E4E20002A03)
```

Zugehörige Verweise

[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. Mit MQSC-Befehlen können Sie nach Nachrichten suchen, die sich derzeit für eine Subskription in der Warteschlange befinden.

Vorbereitende Schritte

Anmerkung: Für die Schritte in dieser Task müssen Sie MQSC-Befehle ausführen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Informationen zu diesem Vorgang

Die folgenden MQSC-Befehle zeigen nur die Subskriptionen an, die Nachrichten erhalten 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 zur Überprüfung auf Nachrichten, die für einen Subskriptionstyp `DISPLAY SBSTATUS(sub_name) NUMMSG` in die Warteschlange eingereicht wurden, enthält [„Attribute von Subskriptionen anzeigen“](#) auf Seite 189.
2. Wenn der Wert für `NUMMSG` größer als null ist, geben Sie die Warteschlange an, 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 mit dem Beispielprogramm durchsuchen“](#) auf Seite 151 beschriebenen Verfahren anzeigen.

Zugehörige Verweise

[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.

Informationen zu diesem Vorgang

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. Ein Server-Service-Objekt kann beispielsweise 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, aber mehrere Instanzen eines Befehlsserviceobjekts können 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 überprüft, ob das durch den MQSC-Befehl `START SERVICE` gestartete Programm noch aktiv ist.

Zugehörige Verweise

[SERVICE DEFINI](#)

[ANZEIGEN SVSTATUS](#)

[START SERVICE](#)

[STOP-SERVICE](#)

Serviceobjekt definieren

Sie definieren ein Serviceobjekt mit dem MQSC-Befehl `DEFINE SERVICE`.

Vorbereitende Schritte

Anmerkung: Für diese Task müssen MQSC-Befehle ausgeführt werden. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Prozedur

- Definieren Sie 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 von Serverserviceobjekten 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. Sie wird mithilfe 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 Tasks

„Services verwalten“ auf Seite 194

Eine Instanz eines Serviceobjekts kann vom Warteschlangenmanager automatisch gestartet und gestoppt oder mit den MQSC-Befehlen **START SERVICE** und **STOP SERVICE** gestartet bzw. gestoppt werden.

Zugehörige Verweise

[SERVICE DEFINI](#)

[ANZEIGEN SVSTATUS](#)

[START SERVICE](#)

[STOP-SERVICE](#)

Services verwalten

Eine Instanz eines Serviceobjekts kann vom Warteschlangenmanager automatisch gestartet und gestoppt oder mit den MQSC-Befehlen **START SERVICE** und **STOP SERVICE** gestartet bzw. gestoppt werden.

Vorbereitende Schritte

Anmerkung: Für diese Task müssen MQSC-Befehle ausgeführt werden. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#) .

Prozedur

- Setzen Sie den Parameter **CONTROL** auf dem Warteschlangenmanager, um eine Instanz eines Serviceobjekts automatisch zu starten bzw. zu stoppen, oder verwenden Sie die MQSC-Befehl **START SERVICE** und **STOP SERVICE** , um dies manuell zu tun.

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. Das folgende Beispiel zeigt einen Protokolleintrag für ein Serverserviceobjekt, das gestartet wird:

```
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.
```

Das folgende Beispiel zeigt einen Protokolleintrag für ein Befehlsserviceobjekt, das gestartet wird:

```
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. Das folgende Beispiel zeigt einen Protokolleintrag für ein Serverserviceobjekt, das gestoppt wird:

```
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 Tasks

„Zusätzliche Umgebungsvariablen in der Datei `service.env` definieren“ auf Seite 195

Wenn ein Service gestartet wird, wird die Umgebung, in der der Serviceprozess gestartet wird, aus der Umgebung des Warteschlangenmanagers übernommen. Sie können zusätzliche Umgebungsvariablen definieren, die in der Umgebung des Serviceprozesses festgelegt werden sollen, indem Sie die Variablen, die Sie definieren möchten, zu einer der `service.env`-Umgebungsüberschreibungsdateien hinzufügen.

Zugehörige Verweise

[STOP SERVICE \(Service stoppen\) unter Multiplatforms](#)

[START SERVICE \(Service starten\) unter Multiplatforms](#)

Zusätzliche Umgebungsvariablen in der Datei `service.env` definieren

Wenn ein Service gestartet wird, wird die Umgebung, in der der Serviceprozess gestartet wird, aus der Umgebung des Warteschlangenmanagers übernommen. Sie können zusätzliche Umgebungsvariablen definieren, die in der Umgebung des Serviceprozesses festgelegt werden sollen, indem Sie die Variablen, die Sie definieren möchten, zu einer der `service.env`-Umgebungsüberschreibungsdateien hinzufügen.

Informationen zu diesem Vorgang

Es gibt zwei mögliche Dateien, denen Sie Umgebungsvariablen hinzufügen können:

- Die Datei `service.env` im Maschinenbereich
- Die Datei des Queue Manager Bereichs `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.

Sie können eine beliebige Umgebungsvariable in der Datei `service.env` angeben. Wenn der IBM MQ-Service beispielsweise eine Reihe von Befehlen ausführt, kann es nützlich sein, die Benutzervariable **PATH** in der Datei `service.env` festzulegen.

Anmerkung: Die von Ihnen festgelegten Werte für die Variable können keine Umgebungsvariablen sein. Beispiel: `CLASSPATH= %CLASSPATH%` ist falsch. In ähnlicher Weise würde `Linux PATH= $PATH :/opt/mqm/bin` unerwartete Ergebnisse liefern.


CLASSPATH muss in Großbuchstaben angegeben werden und die Klassenpfadanweisung darf nur Literale enthalten. Einige Services (z. B. Telemetrie) legen ihren eigenen Klassenpfad fest. Die in `service.env` definierte **CLASSPATH** wird hinzugefügt.


Das Format der in der Datei `service.env` definierten Variablen ist eine Liste von Name/Wert-Variablenpaaren. Jede Variable muss in einer neuen Zeile definiert werden, und jede Variable wird so verwendet, wie sie explizit definiert ist, einschließlich des Leerraums.

Prozedur


- Fügen Sie Umgebungsvariablen zur Datei `service.env` im Maschinenbereich hinzu.


Diese Datei befindet sich in:

–  `/var/mqm` auf AIX and Linux-Systemen.

–  Datenverzeichnis, das während der Installation auf Windows-Systemen ausgewählt wurde.

- Fügen Sie Umgebungsvariablen zur Datei `service.env` im Warteschlangenmanagerbereich hinzu. Diese Datei befindet sich im Datenverzeichnis des Warteschlangenmanagers. Beispiel: Die Position der Umgebungsüberschreibungsdatei für einen Warteschlangenmanager mit dem Namen `QMNAME` lautet:

–  Auf AIX and Linux -Systemen, `/var/mqm/qmgrs/QMNAME/service.env`

-  Auf Windows -Systemen, C:\ProgramData \IBM \MQ\mqgrs\QMNAME\service.env

Beispiel einer service.env-Datei

```

#*****#
#*                                     *#
#* <N_OCO_COPYRIGHT>                 *#
#* Licensed Materials - Property of IBM *#
#*                                     *#
#* 63H9336                             *#
#* (C) Copyright IBM Corporation 2005, 2024. *#
#*                                     *#
#* <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 Tasks

„Austauschbare Einfügungen für Servicedefinitionen verwenden“ auf Seite 196

Sie können Tokens in der Definition eines Serviceobjekts ersetzen. Ersetzte Token werden automatisch durch ihren erweiterten Text ersetzt, wenn das Serviceprogramm ausgeführt wird.

Zugehörige Verweise

[Beschreibung der Umgebungsvariablen](#)

Austauschbare Einfügungen für Servicedefinitionen verwenden

Sie können Tokens in der Definition eines Serviceobjekts ersetzen. Ersetzte Token werden automatisch durch ihren erweiterten Text ersetzt, wenn das Serviceprogramm ausgeführt wird.

Informationen zu diesem Vorgang

Ersatztokens können aus der folgenden Liste allgemeiner Tokens oder aus allen Variablen, die in der Datei service.env definiert sind, entnommen werden.

Prozedur

- 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 in den Abschnitten „Serverserviceobjekt verwenden“ auf Seite 197 und „Befehlsserviceobjekt verwenden“ auf Seite 199.



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 AIX and Linux ist die IBM MQ-Datenverzeichnisposition /var/mqm/.

- **Windows** 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 Namenskonvertierung finden Sie im Artikel [IBM MQ -Dateinamen verstehen](#).

Serverserviceobjekt verwenden

Diese Beispiele zeigen, wie ein Serverserviceobjekt zum Starten eines Auslösemonitors oder eines anderen Programms definiert, verwendet und geändert wird.

Vorbereitende Schritte

Anmerkung: Diese Beispiele erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Diese Beispiele werden mit UNIX -Pfadtrennzeichen geschrieben, sofern nicht anders angegeben.

Vorgehensweise

1. Definieren Sie ein Server-Service-Objekt mit dem MQSC-Befehl DEFINE SERVICE :

```
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 Beispielprogramm, das mit IBM MQ bereitgestellt wird und 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 [„Austauschbare Einfügungen für Servicedefinitionen verwenden“](#) auf Seite 196.

2. Eine Instanz des Serverserviceobjekts wird ausgeführt, wenn der Warteschlangenmanager das nächste Mal gestartet wird. Sie können jedoch eine Instanz des Serverserviceobjekts sofort mit dem MQSC-Befehl START SERVICE starten:

```
START SERVICE(S1)
```

3. Zeigen Sie den Status des Serverserviceprozesses mit dem MQSC-Befehl **DISPLAY SVSTATUS** an:

```
DISPLAY SVSTATUS(S1)
```

4. Ändern Sie das Serverserviceobjekt und lassen Sie die Aktualisierungen abrufen, indem Sie den Serverserviceprozess mit dem MQSC-Befehl **ALTER SERVICE** manuell erneut starten.

Das Serverserviceobjekt wird so geändert, dass die Initialisierungswarteschlange als `JUPITER.INITIATION.QUEUE` angegeben wird.

```
ALTER SERVICE(S1) +  
STARTARG('-m +QMNAME+ -q JUPITER.INITIATION.QUEUE')
```

Anmerkung: Ein aktiver Service übernimmt keine Aktualisierungen seiner Servicedefinition, bis er erneut gestartet wird.

5. Starten Sie den Server-Service-Prozess mit den MQSC-Befehlen **STOP SERVICE** und **START SERVICE** erneut, damit die Änderung berücksichtigt wird.

```
STOP SERVICE(S1)
```

Gefolgt von:

```
START SERVICE(S1)
```

Der Serverserviceprozess wird erneut gestartet und nimmt die in „4“ auf Seite 198 vorgenommenen Änderungen wieder auf.

Anmerkung: Der MQSC-Befehl **STOP SERVICE** kann nur verwendet werden, wenn ein Argument **STOPCMD** in der Servicedefinition angegeben ist.

Weitere Beispiele für die Übergabe von Argumenten

- Definieren Sie ein Server-Service-Objekt, um ein Programm mit dem Namen `runserv` zu starten, wenn ein Warteschlangenmanager gestartet wird.

Verwenden Sie dazu den MQSC-Befehl **DEFINE SERVICE**.

Dieses Beispiel wird mit Windows-Pfadtrennzeichen geschrieben.

Eines der Argumente, die an das Startprogramm übergeben werden, ist eine Zeichenfolge mit einem Leerzeichen. Dieses Argument muss als einzelne Zeichenfolge übergeben werden. Um dies zu erreichen, werden Anführungszeichen wie im folgenden Befehl gezeigt verwendet, um das Befehlsserviceobjekt zu definieren.

```
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 Folgendes:

+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.

- **Definieren Sie ein Serverserviceobjekt, das verwendet werden kann, um den Auslösemonitor beim Start des Warteschlangenmanagers automatisch zu starten.**

Verwenden Sie dazu den MQSC-Befehl **DEFINE SERVICE** .

```
DEFINE SERVICE(TRIG_MON_START) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('runmqtim') +
STARTARG('-m +QMNAME+ -q +IQNAME+')
```

Dabei gilt Folgendes:

+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 Verweise

[ALTER SERVICE](#)

[SERVICE DEFINI](#)

[ANZEIGEN SVSTATUS](#)

[START SERVICE](#)

[STOP-SERVICE](#)

Befehlsserviceobjekt verwenden

Diese Beispiele zeigen, wie ein Befehlsserviceobjekt zum Starten eines Programms definiert wird, das Einträge in das Systemprotokoll des Betriebssystems schreibt, wenn ein Warteschlangenmanager gestartet oder gestoppt wird.

Vorbereitende Schritte

Anmerkung: Diese Beispiele erfordern die Ausführung des MQSC-Befehls **DEFINE SERVICE** . Wie Sie dies tun, hängt von der Plattform ab. Sehen [VerbreichungIBM MQ Verwenden von MQSC-Befehlen](#) .

Diese Beispiele werden mit UNIX -Pfadtrennzeichen geschrieben.

Informationen zu diesem Vorgang

In den folgenden Beispielen:

logger ist ein Beispielprogramm, das mit IBM MQ bereitgestellt wird und Einträge in das Systemprotokoll des Betriebssystems schreiben kann.

+QMNAME+ ist ein Token, das den Namen des Warteschlangenmanagers darstellt.

Prozedur

- Definieren Sie ein Befehlsserviceobjekt, um ein Programm zu starten, das Einträge in das Systemprotokoll des Betriebssystems schreibt, wenn ein Warteschlangenmanager gestartet oder gestoppt wird:

```
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')
```

- Definieren Sie ein Befehlsserviceobjekt, um ein Programm zu starten, das Einträge nur dann in das Systemprotokoll des Betriebssystems schreibt, wenn ein WS-Manager gestoppt wird:

```
DEFINE SERVICE(S3) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

Zugehörige Verweise

[SERVICE DEFINI](#)

Objekte zum Auslösen verwalten

Diese Beispiele zeigen, wie eine Anwendung automatisch gestartet wird, 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.

Vorbereitende Schritte

Anmerkung: Diese Beispiele erfordern die Ausführung von MQSC-Befehlen. Wie Sie dies tun, hängt von der Plattform ab. Sehen [Verabreichung IBM MQ Verwenden von MQSC-Befehlen](#).

Diese Beispiele werden mit UNIX -Pfadtrennzeichen geschrieben.

Informationen zu diesem Vorgang

Eine detaillierte Beschreibung des Auslösens finden Sie unter [IBM MQ -Anwendungen mit Triggern starten](#).

Prozedur

- Definieren Sie eine Anwendungswarteschlange zum Auslösen.

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, reiht 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.QUEUE 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:\appl\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. For example:

```
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.

Dienstprogramm `dmpmqmsg` zwischen zwei Systemen verwenden

Mit dem Dienstprogramm `dmpmqmsg` (früher `qload`) können Sie den Inhalt einer Warteschlange oder ihrer Nachrichten in eine Datei kopieren oder verschieben.

Übersicht

Die Datei, die Sie mit `dmpmqmsg` erstellen, kann nach Bedarf gespeichert und zu einem späteren Zeitpunkt zum erneuten Laden der Nachrichten in die Warteschlange verwendet werden.

Wichtig:

1. 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.
2. Das Dienstprogramm `dmpmqmsg` wird mit der Laufzeitdateigruppe für AIX, Linux, and Windowsausgeliefert und ist daher sowohl im IBM MQ -Server als auch im Client verfügbar.

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 `-1` für die Angabe der lokalen Bindung oder der Clientbindung. `-1` 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 der Beispielanwendung **amqspu**t (lokaler Warteschlangenmanager) oder **amqspu**tc (ferner Warteschlangenmanager) hinzugefügt wurden.

Auf der Linux-Maschine wird Folgendes angezeigt:

```
display q1(Q1) CURDEPTH
      2 : display q1(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLLOCAL)
      CURDEPTH(4)
```

Setzen Sie die Umgebungsvariable **MQSERVER** so, dass sie auf den Warteschlangenmanager in Linux verweist. For example:

```
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 die Ausgabe in `c:\temp\mqqload.txt` zu speichern.

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 in einem Format, das vom Dienstprogramm **dmpmqmsg** verstanden wird.

Setzen Sie auf der Windows -Maschine den **dmpmqmsg** -Befehl ab (mit der Option `-o` anstelle der Option `-i`), um die Warteschlange (*Q2*) auf der Linux -Maschine aus einer Datei auf der Windows -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 q1(Q2) CURDEPTH
      6 : display q1(Q2) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q2)
      TYPE(QLLOCAL)
      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(QLOCAL)
    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 204

Einfache Möglichkeiten zur Verwendung des Dienstprogramms **dmpmqmsg** (früher **qload**).

Beispiele für die Verwendung des Dienstprogramms dmpmqmsg

Einfache Möglichkeiten zur Verwendung des Dienstprogramms **dmpmqmsg** (früher **qload**).

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 204 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 204 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 204. 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 204), 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.

Mit fernen IBM MQ-Objekten arbeiten

Sie können IBM MQ-Objekte auf fernen Warteschlangenmanagern verwalten, indem Sie MQSC-Befehle, PCF-Befehle oder die administrative REST API verwenden. Bevor Sie eine dieser Methoden verwenden können, müssen Sie Übertragungswarteschlangen und Kanäle zwischen dem lokalen Warteschlangenmanager und dem fernen Warteschlangenmanager definieren, so dass Befehle an den fernen Warteschlangenmanager gesendet werden können, und die Antworten, die vom lokalen Warteschlangenmanager empfangen werden. Alternativ können Sie einen WS-Manager-Cluster konfigurieren und dann dieselben fernen Verwaltungsmethoden verwenden.

Informationen zu diesem Vorgang

Zur Vorbereitung von Warteschlangenmanagern für die Fernverwaltung müssen Sie die folgenden Objekte im lokalen Warteschlangenmanager konfigurieren:

- Ein Zuhörer.
- Eine Übertragungswarteschlange, die den Namen des fernen Warteschlangenmanagers hat.
- Ein Senderkanal, der die Verbindungsdetails für den fernen WS-Manager enthält.
- Ein Empfängerkanal, der denselben Namen wie der Senderkanal auf dem fernen Warteschlangenmanager hat.

Außerdem müssen Sie die folgenden Objekte auf dem fernen Warteschlangenmanager konfigurieren:

- Ein Zuhörer.
- Eine Übertragungswarteschlange, die den Namen des lokalen Warteschlangenmanagers hat.
- Ein Senderkanal, der die Verbindungsdetails für den lokalen WS-Manager enthält.
- Ein Empfängerkanal, der denselben Namen wie der Senderkanal auf dem lokalen WS-Manager hat.

Weitere Informationen zur Konfiguration dieser Objekte finden Sie im Abschnitt [„WS-Manager für die Fernverwaltung konfigurieren“](#) auf Seite 207.

Alternativ können Sie einen WS-Manager-Cluster konfigurieren. 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.

Weitere Informationen zum Konfigurieren eines Clusters finden Sie im Abschnitt [Cluster für Warteschlangenmanager konfigurieren](#).

Prozedur

- Informationen zum Verwalten von fernen IBM MQ-Objekten finden Sie in den folgenden Unterabschnitten:
 - „WS-Manager für die Fernverwaltung konfigurieren“ auf Seite 207
 - „Den Befehlsserver für die Fernverwaltung verwalten“ auf Seite 211
 - „MQSC-Befehle auf einem fernen WS-Manager absetzen“ auf Seite 212
 - „Datenkonvertierung zwischen codierten Zeichensätzen“ auf Seite 214

WS-Manager für die Fernverwaltung konfigurieren

Sie können einen fernen Warteschlangenmanager mithilfe der administrative REST API-, MQSC- oder PCF-Befehle von einem lokalen Warteschlangenmanager aus verwalten. Der ferne Warteschlangenmanager befindet sich möglicherweise auf demselben System, in einer anderen Installation oder auf einem anderen System mit derselben Umgebung oder mit einer anderen IBM MQ-Umgebung. Bevor Sie einen Warteschlangenmanager über Remotezugriff von einem lokalen Warteschlangenmanager aus verwalten können, müssen Sie einen Sender- und Empfängerkanal, einen Listener und eine Übertragungswarteschlange auf jedem WS-Manager erstellen. Diese Kanäle und Warteschlangen ermöglichen es, die Befehle an den fernen Warteschlangenmanager und die Antworten zu senden, die auf dem lokalen WS-Manager empfangen werden. Die Prozedur zum Erstellen dieser Warteschlangen und Kanäle ist identisch, unabhängig davon, ob Sie die administrative REST API-, MQSC- oder PCF-Befehle verwenden möchten.

Vorbereitende Schritte

- In der folgenden Prozedur werden Beispielwarteschlangenmanager `source.queue.manager` und `target.queue.manager` verwendet. Sie müssen diese WS-Manager auf Ihrem System erstellen und starten, um diese Schritte ausführen zu können, oder die Namen Ihrer eigenen Warteschlangenmanager in den entsprechenden Schritten zu ersetzen.
- In der folgenden Prozedur wird TCP/IP als Transporttyp verwendet. Sie müssen die IP-Adresse der beiden Systeme kennen, um diese Task ausführen zu können.
- Mit der folgenden Prozedur werden Empfangsprogramme erstellt, die die Netzports 1818 auf dem lokalen System und 1819 auf dem fernen System verwenden. Sie können andere Ports verwenden, müssen jedoch die Portwerte in den entsprechenden Schritten ersetzen.
- Sie müssen die Befehle in der Prozedur lokal oder über eine Netzfunktion wie Telnet ausführen.

Informationen zu diesem Vorgang

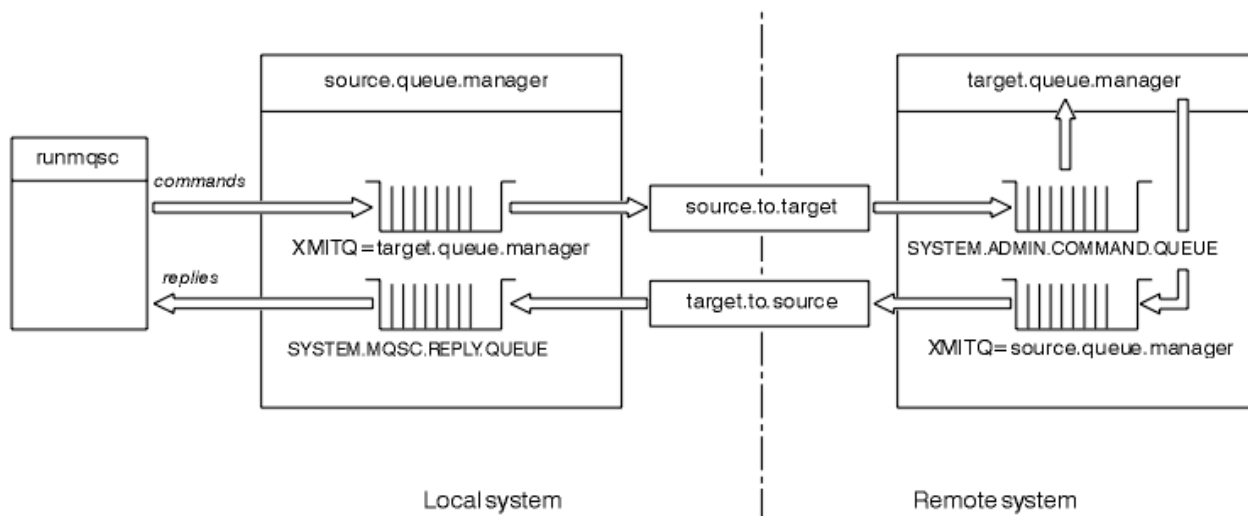


Abbildung 15. Kanäle und Warteschlangen für die Fernverwaltung konfigurieren

Abbildung 15 auf Seite 207 zeigt die Konfiguration von Warteschlangenmanagern, Warteschlangen und Kanälen, die Sie für die Fernverwaltung benötigen:

- Das Objekt `source.queue.manager` ist der Quellenwarteschlangenmanager, aus dem Sie administrative REST API-, MQSC- oder PCF-Befehle absetzen können und an den die Ergebnisse dieser Befehle zurückgegeben werden.
- Das Objekt `target.queue.manager` ist der Name des Zielwarteschlangenmanagers, der die Befehle verarbeitet und alle Bedienernachrichten generiert.
- Befehle werden in die Übertragungswarteschlange gestellt, die denselben Namen wie der ferne WS-Manager hat. In diesem Fall `target.queue.manager`. 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.
- Die Befehle werden vom `source.to.target`-Kanal an den `SYSTEM.ADMIN.COMMAND.QUEUE` auf dem fernen Warteschlangenmanager gesendet. Jedes Ende des Kanals verfügt über eine separate Definition. Ein Ende ist ein Sender und das andere Ende ist ein Empfänger. Die beiden Definitionen müssen denselben Namen haben und zusammen einen einzigen Nachrichtenkanal bilden.
- Die Befehlsausgabe wird in die ferne Übertragungswarteschlange gestellt, die denselben Namen wie der lokale WS-Manager hat, von dem aus der Befehl gesendet wurde. In diesem Fall `source.queue.manager`.
- Die Ausgabe wird vom `target.to.source`-Kanal an eine entsprechende Antwortwarteschlange gesendet, wo sie vom ursprünglichen Befehl ausgeführt und ausgegeben wird.

Vorgehensweise

1. Stellen Sie sicher, dass die Befehlswarteschlange `SYSTEM.ADMIN.COMMAND.QUEUE` auf dem fernen System-WS-Manager vorhanden ist. Diese Warteschlange wird standardmäßig bei der Erstellung eines Warteschlangenmanagers erstellt.
2. Überprüfen Sie auf dem fernen System, ob der Befehlsserver auf dem WS-Manager ausgeführt wird. Wenn der Befehlsserver nicht aktiv ist, ist die Fernverwaltung nicht möglich:
 - a) Starten Sie **runmqsc** für den Warteschlangenmanager. Geben Sie zum Beispiel für den Warteschlangenmanager `target.queue.manager` den folgenden Befehl ein:

```
runmqsc target.queue.manager
```

- b) Zeigen Sie den Status des Befehlsservers an, indem Sie den folgenden Befehl eingeben:

```
DISPLAY QMSTATUS CMDSERV
```

- c) Beenden Sie die **runmqsc**-Eingabeaufforderung, indem Sie den folgenden Befehl eingeben:

```
end
```

- d) Wenn der Befehlsserver nicht gestartet ist, starten Sie ihn. Geben Sie zum Beispiel für den Warteschlangenmanager `target.queue.manager` den folgenden Befehl ein:

```
strmqcsv target.queue.manager
```

3. Definieren Sie die Kanäle, den Listener und die Übertragungswarteschlange auf dem lokalen Warteschlangenmanager:

- a) Starten Sie **runmqsc** für den Warteschlangenmanager. Geben Sie zum Beispiel für den Warteschlangenmanager `source.queue.manager` den folgenden Befehl ein:

```
runmqsc source.queue.manager
```

- b) Definieren Sie den Senderkanal. Dieser Senderkanal muss denselben Namen wie der Empfängerkanal auf dem fernen Warteschlangenmanager haben. Geben Sie beispielsweise den folgenden

MQSC-Befehl ein, und ersetzen Sie dabei den Wert für **CONNAME** durch die IP-Adresse des fernen Warteschlangenmanagers und die Portnummer des Listeners:

```
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME (localhost:1819) +
  XMITQ ('target.queue.manager') +
  TRPTYPE(TCP)
```

- c) Definieren Sie den Empfängerkanal. Dieser Empfängerkanal muss denselben Namen wie der Senderkanal auf dem fernen Warteschlangenmanager haben. Geben Sie zum Beispiel den folgenden Befehl ein:

```
DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)
```

- d) Definieren Sie den Listener auf dem lokalen Warteschlangenmanager. Geben Sie zum Beispiel den folgenden Befehl ein:

```
DEFINE LISTENER ('source.queue.manager') +
  TRPTYPE (TCP) +
  PORT (1818)
```

- e) Definieren Sie die Übertragungswarteschlange auf dem lokalen WS-Manager. Diese Übertragungswarteschlange muss denselben Namen wie der ferne WS-Manager haben. Geben Sie zum Beispiel den folgenden Befehl ein:

```
DEFINE QLOCAL ('target.queue.manager') +
  USAGE (XMITQ)
```

- f) Starten Sie den Listener. Geben Sie zum Beispiel den folgenden Befehl ein:

```
START LISTENER ('source.queue.manager')
```

- g) Beenden Sie die **runmqsc** -Eingabeaufforderung, indem Sie den folgenden Befehl eingeben:

```
end
```

4. Definieren Sie die Kanäle, den Listener und die Übertragungswarteschlange auf dem fernen Warteschlangenmanager:

- a) Starten Sie **runmqsc** für den Warteschlangenmanager. Geben Sie zum Beispiel für den Warteschlangenmanager `target.queue.manager` den folgenden Befehl ein:

```
runmqsc target.queue.manager
```

- b) Definieren Sie den Senderkanal. Dieser Senderkanal muss denselben Namen wie der Empfängerkanal im lokalen WS-Manager haben. Geben Sie beispielsweise den folgenden MQSC-Befehl ein, und ersetzen Sie den Wert für **CONNAME** durch die IP-Adresse für den lokalen WS-Manager und die Portnummer für den Listener:

```
DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(SDR) +
  CONNAME (localhost:1818) +
  XMITQ ('source.queue.manager') +
  TRPTYPE(TCP)
```

- c) Definieren Sie den Empfängerkanal. Dieser Empfängerkanal muss denselben Namen wie der Senderkanal auf dem lokalen WS-Manager haben. Geben Sie beispielsweise den folgenden Befehl ein:

```
DEFINE CHANNEL ('source.to.target') +
```

```
CHLTYPE(RCVR) +  
TRPTYPE(TCP)
```

d) Definieren Sie den Listener. Geben Sie zum Beispiel den folgenden Befehl ein:

```
DEFINE LISTENER ('target.queue.manager') +  
TRPTYPE (TCP) +  
PORT (1819)
```

e) Definieren Sie die Übertragungswarteschlange. Diese Übertragungswarteschlange muss denselben Namen wie der lokale WS-Manager haben. Geben Sie zum Beispiel den folgenden Befehl ein:

```
DEFINE QLOCAL ('source.queue.manager') +  
USAGE (XMITQ)
```

f) Starten Sie den Listener. Geben Sie zum Beispiel den folgenden Befehl ein:

```
START LISTENER ('target.queue.manager')
```

g) Beenden Sie **runmqsc**, indem Sie folgenden Befehl eingeben:

```
end
```

5. Starten Sie den Senderkanal auf dem lokalen System:

a) Starten Sie **runmqsc** für den Warteschlangenmanager. Geben Sie zum Beispiel für den Warteschlangenmanager `source.queue.manager` den folgenden Befehl ein:

```
runmqsc source.queue.manager
```

b) Starten Sie den Senderkanal. Geben Sie zum Beispiel den folgenden Befehl ein:

```
START CHANNEL ('source.to.target')
```

c) Beenden Sie **runmqsc**, indem Sie folgenden Befehl eingeben:

```
end
```

6. Starten Sie den Senderkanal auf dem fernen System:

a) Starten Sie **runmqsc** für den WS-Manager. Geben Sie zum Beispiel für den Warteschlangenmanager `target.queue.manager` den folgenden Befehl ein:

```
runmqsc target.queue.manager
```

b) Starten Sie den Senderkanal. Geben Sie zum Beispiel den folgenden Befehl ein:

```
START CHANNEL ('target.to.source')
```

c) Beenden Sie **runmqsc**, indem Sie folgenden Befehl eingeben:

```
end
```

7. Testen Sie, ob die Konfiguration erfolgreich abgeschlossen wurde, indem Sie einen MQSC-Befehl vom lokalen System an den fernen WS-Manager senden:

a) Starten Sie die **runmqsc** -Eingabeaufforderung für den fernen Warteschlangenmanager vom lokalen System aus. Geben Sie zum Beispiel den folgenden Befehl ein:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

b) Zeigen Sie die Warteschlangen auf dem fernen WS-Manager an, indem Sie den folgenden Befehl eingeben:

```
DISPLAY QUEUE (*)
```

Bei einer erfolgreichen Konfiguration wird eine Liste der Warteschlangen vom fernen Warteschlangenmanager angezeigt.

- c) Wenn diese Schritte nicht funktionieren, überprüfen Sie, ob sich die Kanäle auf beiden Systemen in einem aktiven Status befinden. Wenn die Kanäle nicht aktiv sind und nicht gestartet werden, überprüfen Sie, ob die Kanäle und Übertragungswarteschlangen ordnungsgemäß konfiguriert sind und ob der Befehlsserver aktiv ist. Überprüfen Sie beispielsweise, ob der richtige CONNAME für die Senderkanäle angegeben ist und ob die Übertragungswarteschlangen die richtigen Namen haben. Überprüfen Sie außerdem die WS-Manager-Protokolle auf Sicherheitsausnahmen, die zur Lösung des Problems beitragen können.

Ergebnisse

Ihre Warteschlangenmanager sind so konfiguriert, dass der ferne WS-Manager über Remotezugriff vom lokalen System verwaltet wird.

Nächste Schritte

- Weitere Informationen zur Fernverwaltung mit MQSC-Befehlen finden Sie im Abschnitt „[MQSC-Befehle auf einem fernen WS-Manager absetzen](#)“ auf Seite 212.
- Weitere Informationen zum Schreiben von Verwaltungsprogrammen mithilfe von PCF-Befehlen finden Sie im Abschnitt „[IBM MQ Programmable Command Formats verwenden](#)“ auf Seite 28.
- Weitere Informationen zur Verwendung der administrative REST API für die Fernverwaltung finden Sie im Abschnitt „[Fernverwaltung über die REST API](#)“ auf Seite 82.

Den Befehlsserver für die Fernverwaltung verwalten

Jedem WS-Manager ist ein Befehlsserver zugeordnet. 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 einen Beendigungscode oder eine Bedienernachricht zurück. Sie können den Status des Befehlsservers starten, stoppen und anzeigen. Ein Befehlsserver ist für die gesamte Verwaltung mit PCF-Befehlen, der MQAI und auch für die Fernverwaltung obligatorisch.

Vorbereitende Schritte

Abhängig vom Wert des WS-Managerattributs **SCMDSERV** wird der Befehlsserver entweder automatisch gestartet, wenn der Warteschlangenmanager gestartet wird oder manuell gestartet werden muss. Wenn der Befehlsserver automatisch gestartet wird, können Sie die Befehle `strmqcsv` oder `endmqcsv` nicht verwenden, um den Befehlsserver zu starten und zu stoppen. Sie können den Wert des Attributs **SCMDSERV** mit dem MQSC-Befehl **ALTER QMGR** ändern. Standardmäßig wird der Befehlsserver automatisch gestartet.

Wenn Sie einen WS-Manager stoppen, wird auch der ihm zugeordnete Befehlsserver beendet.

Prozedur

- Zeigen Sie den Status des Befehlsservers an:
 - a) Starten Sie die **runmqsc** -Eingabeaufforderung für den entsprechenden Warteschlangenmanager, indem Sie den folgenden Befehl eingeben:

```
runmqsc target.queue.manager
```

Dabei steht `target.queue.manager` für den Warteschlangenmanager, für den der Befehlsserver angezeigt wird.

- b) Zeigen Sie den Status des Befehlsservers an, indem Sie den folgenden MQSC-Befehl eingeben:

```
DISPLAY QMSTATUS CMDSERV
```

c) Beenden Sie die **runmqsc** -Eingabeaufforderung, indem Sie den folgenden Befehl eingeben:

```
end
```

- Wenn der Befehlsserver nicht automatisch gestartet werden soll, starten Sie den Befehlsserver, indem Sie den folgenden Befehl eingeben:

```
strmqcsv target.queue.manager
```

Hierbei steht `target.queue.manager` für den Warteschlangenmanager, für den der Befehlsserver gestartet wird.

- Wenn der Befehlsserver nicht automatisch gestartet werden soll, stoppen Sie den Befehlsserver, indem Sie den folgenden Befehl eingeben:

```
endmqcsv target.queue.manager
```

Hierbei steht `target.queue.manager` für den Warteschlangenmanager, für den der Befehlsserver gestoppt wird.

Standardmäßig stoppt der Befehlsserver auf kontrollierte Weise. Sie können den Befehlsserver sofort stoppen, indem Sie das Flag `-i` zum Befehl hinzufügen.

MQSC-Befehle auf einem fernen WS-Manager absetzen

Nachdem Sie Warteschlangenmanager für die Fernverwaltung konfiguriert haben, können Sie eine bestimmte Form des Befehls **runmqsc** auf einem lokalen System verwenden, um MQSC-Befehle auf einem fernen Warteschlangenmanager auszuführen. Jeder Befehl wird als Escape-PCF an die Befehlswarteschlange, SYSTEM.ADMIN.COMMAND.QUEUE, des fernen Warteschlangenmanagers gesendet. Antworten werden in der Warteschlange SYSTEM.MQSC.REPLY.QUEUE empfangen.

Vorbereitende Schritte

Sie müssen die Schritte in „[WS-Manager für die Fernverwaltung konfigurieren](#)“ auf Seite 207 ausführen, um Kanäle, Übertragungswarteschlangen, Listener und den Befehlsserver zu konfigurieren, bevor Sie einen Warteschlangenmanager remote über MQSC-Befehle verwalten können.

Vorgehensweise

1. Stellen Sie sicher, dass der Befehlsserver auf dem fernen WS-Manager ausgeführt wird.
Informationen zum Starten des Befehlsservers auf einem Warteschlangenmanager finden Sie im Abschnitt „[Den Befehlsserver für die Fernverwaltung verwalten](#)“ auf Seite 211.
2. Auf dem Quellenwarteschlangenmanager können Sie dann MQSC-Befehle auf eine der beiden folgenden Arten ausführen:
 - Interaktiv, indem Sie **runmqsc** mit den folgenden Befehlen starten:
 - Wenn sich der ferne Warteschlangenmanager unter z/OS befindet, geben Sie folgenden Befehl ein:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager
```

- Wenn sich der ferne Warteschlangenmanager auf Multiplatforms befindet, geben Sie den folgenden Befehl ein:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- Aus einer Befehlsdatei:
 - a. Legen Sie die MQSC-Befehle, die auf dem fernen System ausgeführt werden sollen, in einer Textdatei, einem Befehl pro Zeile, ein.
 - b. Überprüfen Sie Ihre MQSC-Befehle auf dem lokalen Warteschlangenmanager mit dem Flag `-v` des Befehls **runmqsc**. Mit dem Flag `-v` wird überprüft, ob die Befehle gültig sind, aber nicht ausgeführt werden. Beachten Sie, dass einige Befehle fehlschlagen, wenn sie auf den fernen Warteschlangenmanager anwendbar sind, aber nicht auf den lokalen Warteschlangenmanager anwendbar sind:

```
runmqsc -v source.queue.manager < myCmdFile.in > results.out
```

Der `myCmdFile.in` enthält die zu überprüfenden MQSC-Befehle, und die Datei `results.out` enthält die Prüfergebnisse für die Befehle.

- c. Führen Sie die Befehlsdatei auf dem fernen WS-Manager aus, indem Sie einen der folgenden Befehle eingeben:
 - Wenn sich der ferne Warteschlangenmanager unter z/OS befindet, geben Sie folgenden Befehl ein:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager < myCmdFile.in > results.out
```

- Wenn sich der ferne Warteschlangenmanager auf Multiplatforms befindet, geben Sie den folgenden Befehl ein:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager < myCmdFile.in > results.out
```

Die verwendeten Parameter sind die folgenden Parameter:

-w seconds

Gibt an, dass die MQSC-Befehle im indirekten Modus ausgeführt werden, wobei Befehle in die Eingabewarteschlange des Befehlsservers gestellt und in der Reihenfolge ausgeführt werden.

Die Variable *seconds* gibt an, wie lange (in Sekunden) auf eine Antwort vom fernen Warteschlangenmanager gewartet werden soll. Alle Antworten, die nach dieser Zeit empfangen werden, werden gelöscht, die MQSC-Befehle werden jedoch weiterhin auf dem fernen Warteschlangenmanager ausgeführt. Die folgende Nachricht wird im lokalen WS-Manager generiert, wenn der Befehl das Zeitlimit für die Ausführung des Befehls hat:

```
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.

-x

Gibt an, dass der ferne Warteschlangenmanager ein z/OS-Warteschlangenmanager ist.

-m localQMgrName

Gibt den Namen des lokalen Warteschlangenmanagers an, der zum Übergeben von Befehlen an den fernen WS-Manager verwendet werden soll.

Nächste Schritte

Wenn Sie Schwierigkeiten beim Ausführen von MQSC-Befehlen über Remotezugriff haben:

- Überprüfen Sie, ob der ferne WS-Manager aktiv ist.
- Überprüfen Sie, ob der Befehlsserver auf dem fernen System ausgeführt wird.

- Überprüfen Sie, ob das Intervall für die Kanalunterbrechung abgelaufen ist. Beispiel: Wenn ein Kanal gestartet wurde, aber dann nach einiger Zeit heruntergefahren wurde. Dies ist besonders wichtig, wenn Sie die Kanäle manuell starten.
- Stellen Sie sicher, dass die Anforderungen, die vom lokalen WS-Manager gesendet werden, für den Zielwarteschlangenmanager sinnvoll sind. Beispiel: Anforderungen, die Parameter enthalten, die auf dem fernen Warteschlangenmanager nicht unterstützt werden.
- Siehe auch [Probleme mit MQSC-Befehlen beheben](#).

Datenkonvertierung zwischen codierten Zeichensätzen

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.

Informationen zur Konvertierung von EBCDIC-Zeilenvorschubzeichen (NL) in ASCII finden Sie im Abschnitt [Zeilengruppe für alle Warteschlangenmanager in der Datei mqs.ini](#) und in der Umgebungsvariablen **AMQ_CONVEBCDICNEWLINE**.

Unterstützte Konvertierungen werden in der [Datenkonvertierungsverarbeitung](#) definiert.

Die Konvertierung zwischen den CCSIDs 37 und 500 wird auf den IBM MQ Appliance, Windows, Linux und macOS unterstützt.

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.

Weitere Informationen zu Plattformen, für die `ccsid_part2.tbl` gilt, finden Sie unter [„Standarddatenkonvertierung angeben“](#) auf Seite 217 [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 215 beschrieben.

Erweiterte Unterstützung für Unicode-Datenkonvertierung

Das Produkt unterstützt alle Unicode-Zeichen, die im Unicode-Standard 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)


Eine zusätzliche Datei, `ccsid_part2.tbl`, wird bereitgestellt.


Die Datei `ccsid_part2.tbl` hat Vorrang vor der Datei `ccsid.tbl` und:

- Hinzufügung oder Änderung von CCSID-Einträgen
- Angabe einer Standarddatenkonvertierung
- Angabe von Daten für verschiedene Befehlsebenen

Die Datei `ccsid_part2.tbl` gilt nur für die folgenden Plattformen:

-  Linux - Alle Versionen
-  Windows

 Unter IBM MQ for Windows befindet sich `ccsid_part2.tbl` standardmäßig im Verzeichnis `MQDataRoot\conv\table`. Darüber hinaus werden in IBM MQ for Windows alle unterstützten codierten Zeichensätze aufgezeichnet.


 Unter IBM MQ for Linux befindet sich `ccsid_part2.tbl` im Verzeichnis `MQDataRoot/conv/table` und die unterstützten codierten Zeichensätze werden in den von IBM MQ bereitgestellten Konvertierungstabellen gespeichert.

Die Datei `ccsid_part2.tbl` ersetzt zwar die vorhandene Datei `ccsid.tbl`, die in früheren Versionen von IBM MQ verwendet wird, um zusätzliche CCSID-Informationen zu liefern, aber die Datei `ccsid.tbl` wird von IBM MQ weiterhin syntaktisch analysiert und darf deshalb nicht gelöscht werden.

Weitere Informationen finden Sie unter „Die `ccsid_part2.tbl`-Datei“ auf Seite 216.

ccsid.tbl-Datei

Auf anderen Plattformen als denen, auf die `ccsid_part2.tbl` angewendet wird, wird die Datei `ccsid.tbl` für die folgenden Zwecke verwendet:

-  Unter AIX werden die unterstützten codierten Zeichensätze intern vom Betriebssystem gespeichert.
- Sie gibt alle zusätzlichen codierten Zeichensätze an. Wenn Sie zusätzliche Codesets angeben möchten, müssen Sie `ccsid.tbl` bearbeiten (Anleitung dazu, wie dies in der Datei bereitgestellt wird).
- Sie gibt eine Standarddatenkonvertierung an.

Sie können die in `ccsid.tbl` aufgezeichneten Informationen aktualisieren. Dies kann sinnvoll sein, wenn z. B. ein zukünftiges Release Ihres Betriebssystems zusätzliche codierte Zeichensätze unterstützt.

Standarddatenkonvertierung

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.

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 Zeichendaten 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 Datenkonvertierungsexit 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 mit dem Attribut **CCSID** des Befehls [ALTER QMGR](#) die CCSID des Warteschlangenmanagers geändert haben, stoppen Sie den Warteschlangenmanager und starten Sie ihn erneut, um sicherzustellen, dass alle aktiven Anwendungen, einschließlich des Befehlsservers und der Kanalprogramme, 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 `ccsid_part2.tbl` -Datei

Die Datei `ccsid_part2.tbl` wird verwendet, um zusätzliche CCSID-Informationen zu liefern. Die Datei `ccsid_part2.tbl` ersetzt die Datei `ccsid.tbl`, die vor IBM MQ 9.0 verwendet wurde.

Anmerkung: Die `ccsid.tbl` -Datei, die vor IBM MQ 9.0 verwendet wurde, um zusätzliche CCSID-Informationen zu liefern, wird weiterhin von IBM MQ syntaktisch analysiert und sollte nicht gelöscht werden. Die Einträge in `ccsid_part2.tbl` haben jedoch Vorrang vor anderen Einträgen in `ccsid.tbl`.

Sie sollten `ccsid_part2.tbl` statt `ccsid.tbl` verwenden, da `ccsid_part2.tbl`:

- Enthält Unterstützung für die Unicode-Codierungswerte. Ab IBM MQ 9.0 unterstützt das Produkt alle Unicode-Zeichen, die in der Unicode 8.0-Norm in der Datenkonvertierung definiert sind, einschließlich der vollständigen Unterstützung für UTF-16. Weitere Informationen finden Sie unter [„Datenkonvertierung zwischen codierten Zeichensätzen“](#) auf Seite 214.
- Hier können Sie die Version der CCSID-Einträge angeben, so dass die Einträge nur auf ausgewählte Befehlsebenen angewendet werden können.

Sie können die Datei `ccsid_part2.tbl` wie folgt verwenden:



- CCSID-Einträge hinzufügen oder ändern
- Angabe einer Standarddatenkonvertierung
- Angabe von Daten für verschiedene Befehlsebenen

Die Datei `ccsid_part2.tbl` ist nur auf die folgenden Plattformen anwendbar:

-  Linux - Alle Versionen

-  Windows

Die Position der `ccsid_part2.tbl` -Datei hängt von Ihrer Plattform ab:

-  Das Verzeichnis `MQDataRoot/conv/table` in allen Versionen von Linux.
-  Das Verzeichnis `MQDataRoot\conv\table` unter Windows.

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.

Bei der ersten Syntaxanalyse von `ccsid_part2.tbl` behandelt IBM MQ alle vorgefundenen 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
```

Managed File Transfer verwalten

Managed File Transfer wird mithilfe von Managed File Transfer-Befehlen verwaltet. Für einige Verwaltungsaufgaben können Sie auch IBM MQ Explorer verwenden.

Übertragung durch Einreihen einer Nachricht in eine Agentenbefehlswarteschlange starten

Sie können eine Dateiübertragung auch starten, indem Sie eine Dateiübertragungsnachricht in die Befehlswarteschlange des Quellenagenten einreihen. Die Befehlswarteschlange könnte zum Beispiel SYS-

TEM.FTE.COMMAND.AGENT01 heißen. Sie müssen allerdings sicherstellen, dass die Nachricht die Befehlswarteschlange des richtigen Quellenagenten erreicht; falls die Nachricht bei einem Agenten eingeht, der nicht zu den Quelleninformationen der XML passt, wird die Nachricht zurückgewiesen.

Die Übertragungsanforderung-XML muss dem `FileTransfer.xsd`-Schema entsprechen und das Element `<request>` als Stammelement verwenden. Im Abschnitt [Nachrichtenformat für Dateiübertragungsanforderungen](#) finden Sie Informationen zu Struktur und Inhalt einer Übertragungsanforderungsnachricht. Wie Sie die Übertragungsanforderungsnachricht in die Befehlswarteschlange eines Agenten einreihen, hängt von der jeweiligen Task ab. Sie können beispielsweise die IBM MQ Java-API verwenden, um eine Nachricht programmgesteuert in die Warteschlange zu stellen.

Zugehörige Konzepte

[„Daten von Dateien in Nachrichten übertragen“](#) auf Seite 276

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

[„Daten von Nachrichten in Dateien übertragen“](#) auf Seite 292

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nachrichten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

[„Die Protokollbridge“](#) auf Seite 303

Die Protokollbridge ermöglicht in Ihrem Managed File Transfer- bzw. MFT-Netz den Zugriff auf Dateien, die in Ihrer lokalen Domäne oder remote auf einem Dateiserver außerhalb Ihres MFT-Netzes gespeichert sind. Dieser Dateiserver kann die FTP-, FTPS- oder SFTP-Netzprotokolle verwenden. Für jeden Dateiserver ist mindestens ein dedizierter Agent erforderlich. Der dedizierte Agent wird als Protokoll-Bridge-Agent bezeichnet. Ein Bridge-Agent kann mit mehreren Dateiservern interagieren.

[„Mit MFT über IBM Integration Bus arbeiten“](#) auf Seite 344

Unter Verwendung der FTEOutput- und FTEInput-Knoten können Sie mit Managed File Transfer über IBM Integration Bus arbeiten.

[„Wiederherstellung und Neustart von MFT“](#) auf Seite 344

Sollte aus irgendeinem Grund Ihr Agent oder Warteschlangenmanager z. B. aufgrund eines Strom- oder Netzausfalls nicht verfügbar sein, führt Managed File Transfer eine Wiederherstellung gemäß folgender Szenarios durch:

Zugehörige Tasks

[„MFT-Agenten starten“](#) auf Seite 220

Bevor Sie einen Managed File Transfer-Agenten für eine Dateiübertragung verwenden können, müssen Sie diesen aufrufen.

[„Neue Dateiübertragung starten“](#) auf Seite 228

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

[„Überwachen von Dateiübertragungen, die sich in Bearbeitung befinden“](#) auf Seite 235

Sie können eine laufende Dateiübertragung über die Registerkarte **Verwaltete Dateiübertragung-Aktueller Übertragungsfortschritt** in IBM MQ Explorer überwachen. Diese Dateiübertragung kann aus IBM MQ Explorer oder von der Befehlszeile aus gestartet worden sein. Auf der Registerkarte wird auch der Fortschritt geplanter Übertragungen an der Stelle angezeigt, an der die geplanten Übertragungen beginnen.

[„Status von Dateiübertragungen im Übertragungsprotokoll anzeigen“](#) auf Seite 237

Sie können im **Übertragungsprotokoll** in IBM MQ Explorer die Details zu Dateiübertragungen ansehen. Dabei spielt es keine Rolle, ob die Übertragungen über die Befehlszeile oder über IBM MQ Explorer gestartet wurden. Sie können den Inhalt des **Übertragungsprotokolls** anpassen.

[„MFT-Ressourcen überwachen“](#) auf Seite 239

Sie können Managed File Transfer-Ressourcen überwachen, beispielsweise eine Warteschlange oder ein Verzeichnis. Sobald für diese Ressource eine festgelegte Bedingung eintritt, startet die Ressourcenüber-

wachung eine bestimmte Task, beispielsweise die Übertragung einer Datei. Ressourcenüberwachungen können mit dem Befehl **fteCreateMonitor** oder über die Ansicht **Überwachungen** im Managed File Transfer-Plug-in für IBM MQ Explorer erstellt werden.

„Mit Dateiübertragungsvorlagen arbeiten“ auf Seite 273

Sie können Dateiübertragungsschablonen verwenden, um häufige Dateiübertragungseinstellungen für wiederholte oder komplexe Übertragungen zu speichern. Sie können entweder über die Befehlszeile mit dem Befehl **fteCreateTemplate** oder über IBM MQ Explorer mit dem Assistenten **Neue Vorlage für Managed File Transfer erstellen** eine Übertragungsvorlage erstellen oder bei der Erstellung einer Dateiübertragung eine Vorlage speichern, indem Sie das Kontrollkästchen **Übertragungseinstellungen als Vorlage sichern** aktivieren. Im Fenster **Übertragungsvorlagen** werden alle Übertragungsvorlagen angezeigt, die Sie in Ihrem Managed File Transfer-Netz erstellt haben.

„MFT-Agenten auflisten“ auf Seite 225

Sie können die bei einem bestimmten Warteschlangenmanager registrierten Managed File Transfer-Agenten über die Befehlszeile oder über die IBM MQ Explorerauflisten.

„MFT-Agenten stoppen“ auf Seite 226

Ein Managed File Transfer-Agent kann über die Befehlszeile gestoppt werden. Wenn Sie einen Agenten stoppen, legen Sie diesen still und ermöglichen dem Agenten gleichzeitig, seine aktuelle Dateiübertragung vor dem Stoppen abzuschließen. Zum sofortigen Stoppen eines Agenten können Sie auch den Parameter **-i** in der Befehlszeile angeben. Nach dem Stoppen des Agenten können Sie den Agenten erst dann wieder zum Übertragen von Dateien verwenden, wenn Sie diesen erneut starten.

MFT-Protokollfunktion konfigurieren

Zugehörige Verweise


Richtlinien für das Übertragen von Dateien


MFT-Agenten starten

Bevor Sie einen Managed File Transfer-Agenten für eine Dateiübertragung verwenden können, müssen Sie diesen aufrufen.

Informationen zu diesem Vorgang

Ein Managed File Transfer Agent kann über die Befehlszeile gestartet werden. In diesem Fall wird der Agentenprozess gestoppt, wenn Sie sich vom System abmelden.

 Unter AIX, Linux, and Windows können Sie einen Agenten so konfigurieren, dass er auch dann ausgeführt wird, wenn Sie sich vom System abmelden, und er weiterhin Dateiübertragungen empfängt.

 Unter z/OS können Sie den Agenten so konfigurieren, dass er über die Jobsteuersprache als gestartete Task startet, ohne dass Sie dafür eine interaktive Sitzung benötigen.

Wenn ein Agent bei seiner Ausführung einen nicht behebbaren Fehler feststellt, wird eine erste Fehlerdatenerfassung (FDC) generiert, und der Agent wird gestoppt.


Prozedur


- Verwenden Sie den Befehl **fteStartAgent**, um einen Agenten über die Befehlszeile zu starten.

Weitere Informationen finden Sie unter **fteStartAgent**.

- 

Gehen Sie wie folgt vor, um einen Agenten so zu konfigurieren, dass er weiterhin ausgeführt wird, wenn Sie sich vom System abmelden

-  Unter Windows konfigurieren Sie den Agenten für die Ausführung als Windows-Service. Weitere Informationen finden Sie unter „MFT-Agenten als Windows-Dienst starten“ auf Seite 221.

-  Konfigurieren Sie den Agenten auf AIX and Linux so, dass er während eines Neustarts mithilfe einer Skriptdatei automatisch gestartet wird. Weitere Informationen finden Sie unter „[Starten eines MFT-Agenten beim Start des AIX and Linux-Systems](#)“ auf Seite 223.

- 

Unter z/OS konfigurieren Sie den Agenten so, dass er über die Jobsteuersprache als gestartete Task startet, ohne dass Sie dafür eine interaktive Sitzung benötigen.

Weitere Informationen finden Sie unter „[Starting an MFT agent on z/OS](#)“ auf Seite 225.

Zugehörige Tasks

„MFT-Agenten auflisten“ auf Seite 225

Sie können die bei einem bestimmten Warteschlangenmanager registrierten Managed File Transfer -Agenten über die Befehlszeile oder über die IBM MQ Explorerauflisten.

„MFT-Agenten stoppen“ auf Seite 226

Ein Managed File Transfer-Agent kann über die Befehlszeile gestoppt werden. Wenn Sie einen Agenten stoppen, legen Sie diesen still und ermöglichen dem Agenten gleichzeitig, seine aktuelle Dateiübertragung vor dem Stoppen abzuschließen. Zum sofortigen Stoppen eines Agenten können Sie auch den Parameter **-i** in der Befehlszeile angeben. Nach dem Stoppen des Agenten können Sie den Agenten erst dann wieder zum Übertragen von Dateien verwenden, wenn Sie diesen erneut starten.

Zugehörige Verweise

Statusinformationen zum MFT-Agenten

fteStartAgent

MFT-Agenten als Windows-Dienst starten

Sie können einen Agenten als Windows-Dienst starten, sodass der Agent auch nach Ihrer Abmeldung von Windows weiterhin aktiv ist und Dateiübertragungen empfangen kann.

Informationen zu diesem Vorgang

Wenn Sie unter Windows einen Agenten über die Befehlszeile starten, wird der Agentenprozess unter dem Benutzernamen ausgeführt, mit dem Sie sich bei Windows angemeldet haben. Wenn Sie sich vom System abmelden, wird der Agentenprozess gestoppt. Um dies zu vermeiden, müssen Sie den Agenten so konfigurieren, dass er als Windows-Dienst ausgeführt wird. Als Windows-Dienst können Sie Agenten so konfigurieren, dass sie beim Starten oder Neustarten der Windows-Umgebung automatisch ebenfalls gestartet werden.

Führen Sie die nachfolgend beschriebenen Schritte aus, um einen Agenten zu starten, der als Windows-Dienst ausgeführt wird. Sie müssen Managed File Transfer in einer der unterstützten Windows -Versionen ausführen, um den Agenten als Windows -Dienst ausführen zu können. Eine Liste der unterstützten Umgebungen finden Sie im Abschnitt [System Requirements for IBM MQ](#).

Die genauen Schritte hängen davon ab, ob Sie bereits einen Agenten erstellt haben oder ob Sie einen Agenten erstellen. Beide Optionen werden in den folgenden Schritten beschrieben.

Vorgehensweise

1. Wenn Sie einen Managed File Transfer -Agenten erstellen, verwenden Sie den Befehl **fteCreateAgent**, **fteCreateCDAgent** oder **fteCreateBridgeAgent**. Geben Sie den Parameter **-s** an, damit der Agent als Windows-Dienst ausgeführt wird. Im folgenden Beispiel wird der Agent AGENT1 erstellt, der über einen Agenten-WS-Manager QMGR1 verfügt. Der Windows-Dienst wird unter dem Benutzernamen `fteuser` ausgeführt, dem das Kennwort `ftepassword` zugeordnet ist.

```
fteCreateAgent -agentName AGENT1 -agentQMGR QMGR1 -s -su fteuser -sp ftepassword
```

Sie können optional einen Namen für den Service nach dem Parameter **-s** angeben. Wenn Sie keinen Namen angeben, hat der Service den Namen `mqmftAgentAGENTQMGR`, wobei `AGENT` für den angege-

benen Agentennamen steht und *QMGR* für den Namen Ihres Agenten-WS-Managers steht. In diesem Beispiel lautet der Standardname für den Service `mqmftAgentAGENT1QMGR1`.

Anmerkung: Das Windows -Benutzerkonto, das Sie mit dem Parameter **-su** angeben, muss über die Berechtigung **Log on as a service** verfügen. Informationen zur Konfiguration finden Sie im Abschnitt [Fehlerbehebung für einen MFT -Agenten oder eine Protokollfunktion, der bzw. die als Windows -Dienst ausgeführt wird](#).

Weitere Informationen finden Sie unter [fteCreateAgent](#), [fteCreateCDAgent: Connect:Direct -Bridgeagenten erstellen](#) oder [fteCreateBridgeAgent \(MFT -Protokollbridgeagenten erstellen und konfigurieren\)](#).

2. Wenn Sie den vorherigen Schritt zum Erstellen eines Agenten ausgeführt haben, führen Sie die MQSC-Befehle aus, die vom Befehl **fteCreateAgent**, **fteCreateCDAgent** oder **fteCreateBridgeAgent** generiert werden. Mit diesen Befehlen werden die IBM MQ-Warteschlangen erstellt, die für den Agenten erforderlich sind.

Für einen Agenten mit dem Namen *AGENT1*, einem Agentenwarteschlangenmanager mit dem Namen *QMGR1* und einem Koordinations-WS-Manager mit dem Namen *COORDQMGR1* führen Sie beispielsweise den folgenden Befehl aus:

```
runmqsc QMGR1 MQ_DATA_PATH\mqft\config\COORDQMGR1\agents\AGENT1\AGENT1_create.mqsc
```

3. Wenn Sie keinen Agenten wie oben beschrieben erstellt haben, sondern einen bereits vorhandenen Agenten als Windows-Dienst konfigurieren möchten, müssen Sie den Agenten, falls er aktiv ist, stoppen und anschließend die Agentenkonfiguration ändern.

- a) Im folgenden Beispiel wird ein Agent mit dem Namen *AGENT1* verwendet. Führen Sie den folgenden Befehl aus:

```
fteStopAgent AGENT1
```

- b) Konfigurieren Sie den Agenten mit dem Befehl **fteModifyAgent** für die Ausführung als Windows-Dienst:

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

Weitere Informationen finden Sie im Abschnitt [fteModifyAgent: MFT-Agenten als Windows-Dienst ausführen](#).

4. Starten Sie Ihren Agenten mit dem Befehl **fteStartAgent** . Alternativ können Sie den Dienst auch über das Windows-Tool 'Dienste' starten, das unter 'Verwaltung' in der Systemsteuerung, die Sie im Desktop-Startmenü von Windows auswählen, verfügbar ist.

```
fteStartAgent AGENT1
```

Der Dienst wird auch nach der Windows-Abmeldung weiter ausgeführt. Um sicherzustellen, dass der Dienst nach einem Systemabschluss beim anschließenden Neustart von Windows ebenfalls erneut gestartet wird, muss das Feld **Autostarttyp** im Windows-Tool 'Dienste' auf den Standardwert **Automatisch** gesetzt sein. Setzen Sie dieses Feld auf **Manuell**, wenn der Dienst beim Neustart von Windows nicht gestartet werden soll.

5. Optional: Stoppen Sie den Agenten mit dem Befehl [fteStopAgent](#) oder mit dem Windows-Tool 'Dienste'. Führen Sie z. B. über die Befehlszeile den folgenden Befehl aus:

```
fteStopAgent AGENT1
```

- Wenn Sie den Befehl **fteStopAgent** als Dienst ausführen, wird der Befehl stets mit dem Parameter **-i** ausgeführt, unabhängig davon, ob Sie diesen Parameter angegeben haben. Mit dem Parameter **-i** wird der Agent sofort gestoppt, ohne die Übertragungen zu beenden, die sich in Bearbeitung befinden. Dies ist auf eine Einschränkung des Windows-Dienstes zurückzuführen.

Nächste Schritte

Wenn beim Starten des Windows -Service Probleme auftreten, lesen Sie den Abschnitt [Fehlerbehebung für einen MFT -Agenten oder eine Protokollfunktion, der bzw. die als Windows -Dienst ausgeführt wird](#). In diesem Abschnitt wird angegeben, wo sich die Protokolldateien des Windows-Diensts befinden.

Zugehörige Tasks

„MFT-Agenten auflisten“ auf Seite 225

Sie können die bei einem bestimmten Warteschlangenmanager registrierten Managed File Transfer -Agenten über die Befehlszeile oder über die IBM MQ Explorerauflisten.

„MFT-Agenten stoppen“ auf Seite 226

Ein Managed File Transfer-Agent kann über die Befehlszeile gestoppt werden. Wenn Sie einen Agenten stoppen, legen Sie diesen still und ermöglichen dem Agenten gleichzeitig, seine aktuelle Dateiübertragung vor dem Stoppen abzuschließen. Zum sofortigen Stoppen eines Agenten können Sie auch den Parameter **-i** in der Befehlszeile angeben. Nach dem Stoppen des Agenten können Sie den Agenten erst dann wieder zum Übertragen von Dateien verwenden, wenn Sie diesen erneut starten.

Zugehörige Verweise

[fteCreateAgent](#) (MFT-Agent erstellen)

[fteCreateCDAgent](#) (Connect:Direct-Bridgeagent erstellen)

[fteCreateBridgeAgent](#) (MFT-Protokollbridgeagenten erstellen und konfigurieren)

[fteModifyAgent](#) (MFT-Agenten als Windows-Service ausführen)

Zugehörige Informationen

Die MFT `agent.properties`-Datei

Linux

AIX

Starten eines MFT-Agenten beim Start des AIX and Linux-Systems

Ein Managed File Transfer Agent kann so konfiguriert werden, dass er beim Systemstart unter AIX and Linux ebenfalls gestartet wird. Wenn Sie sich abmelden, läuft Ihr Agent weiter und kann Dateiübertragungen empfangen.

Wenn Sie einen Agenten mit einem der Managed File Transfer -Befehle **fteCreateAgent**, **fteCreateCDAgent** oder **fteCreateBridgeAgent** erstellt und konfiguriert haben, können Sie ihn so konfigurieren, dass er automatisch während eines Warmstarts auf AIX and Linux -Maschinen gestartet wird, indem Sie eine Skriptdatei verwenden, die einfach den folgenden Befehl ausführt:

```
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

Dabei ist `mq_install_root` das Stammverzeichnis der erforderlichen Managed File Transfer -Installation. Der Standardwert ist `/opt/mqm` und `agent_name` ist der Name des zu startenden Managed File Transfer Agent s. Die Verwendung dieser Skriptdatei variiert je nach Betriebssystem. Unter Linux sind beispielsweise weitere Optionen verfügbar.

Linux

Linux

Auf Linux-Systemen gibt es mehrere Möglichkeiten, Anwendungen während des Systemboots zu starten. Im Allgemeinen wird die folgende Vorgehensweise empfohlen:

1. Erstellen Sie eine Datei mit dem Namen `/etc/rc.mqmft` mit dem Inhalt:

```
#!/bin/sh
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name"
```

Dabei ist `mqmft_user` die Benutzer-ID, unter der der Agentenprozess ausgeführt werden soll. Diese Benutzer-ID muss zur Gruppe 'mqm' gehören.

2. Sorgen Sie dafür, dass die Datei ausgeführt werden kann:

```
chmod 755 /etc/rc.mqmf
```

3. Fügen Sie anschließend die folgende Zeile zu `/etc/inittab` hinzu:

```
mqmf:5:boot:/etc/rc.mqmf
```

Weitere Möglichkeiten zum Starten eines Agenten während des Bootvorgangs in Linux sind das Hinzufügen der Scriptzeilen zur `/etc/rc.d/rc.local`-Datei oder Linux SuSe, das Hinzufügen der Scriptzeilen zu der Datei `/etc/init.d/boot.local`. Wählen Sie das für Ihre Umgebung geeignete Verfahren aus. Hier noch weitere Informationen zu weiteren Möglichkeiten, wie ein Agent unter bestimmten Linux-Distributionen beim Systemstart gestartet werden kann:

SLES 10 und 11

Unter SUSE Linux Enterprise Server (SLES) 10 und 11:

1. Erstellen Sie als Systemstammbenutzer-ID Ihre eigene `/etc/init.d/rc.rclocal`-Datei.
2. Fügen Sie die folgenden Zeilen zur Datei `rc.rclocal` hinzu:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: rc.rclocal
# Required-Start: $network $syslog
# Required-Stop: $network $syslog
# Default-Stop: 0 1 2 6
# Description: MQMFT agent startup
### END INIT INFO
su -l mqmf_user -c mq_install_root/bin/fteStartAgent agent_name"
```

3. Führen Sie folgende Befehle aus:

```
chmod 755 rc.rclocal
chkconfig --add rc.rclocal
```

Starten von Managed File Transfer-Agenten auf Linux mit systemd

Linux

Gehen Sie wie folgt vor:

1. Erstellen Sie im `/etc/systemd/system`-Ordner eine Datei mit dem Namen `<agentname>.service`. Fügen Sie den folgenden Inhalt hinzu, wobei `<agentname>` `MFT_AGT_LNX_0` ist

```
# vi /etc/systemd/system/MFT_AGT_LNX_0.service
[Unit]
Description=IBM MQ MFT MFT_AGT_LNX_0
[Service]
ExecStart=/opt/mqm/bin/fteStartAgent MFT_AGT_LNX_0
ExecStop=/opt/mqm/bin/fteStopAgent MFT_AGT_LNX_0
Type=forking
User=mqm
Group=mqm
KillMode=none
```

2. Um den Dienst zu aktivieren, führen Sie die folgenden Befehle aus:

```
# systemctl enable MFT_AGT_LNX_0
# systemctl daemon-reload
```

3. Um den Agenten zu starten und seinen Status zu überprüfen, führen Sie die folgenden Befehle aus:


```
# systemctl start MFT_AGT_LNX_0
# systemctl status MFT_AGT_LNX_0
```

Zugehörige Tasks

„MFT-Agenten stoppen“ auf Seite 226

Ein Managed File Transfer-Agent kann über die Befehlszeile gestoppt werden. Wenn Sie einen Agenten stoppen, legen Sie diesen still und ermöglichen dem Agenten gleichzeitig, seine aktuelle Dateiübertragung vor dem Stoppen abzuschließen. Zum sofortigen Stoppen eines Agenten können Sie auch den Parameter **-i** in der Befehlszeile angeben. Nach dem Stoppen des Agenten können Sie den Agenten erst dann wieder zum Übertragen von Dateien verwenden, wenn Sie diesen erneut starten.

Zugehörige Verweise

[fteCreateAgent](#)

[fteCreateCDAgent](#): Connect:Direct-Bridgeagenten erstellen

[fteCreateBridgeAgent](#) (MFT-Protokollbridgeagenten erstellen und konfigurieren)

Starting an MFT agent on z/OS

On z/OS, in addition to running the **fteStartAgent** command from a z/OS UNIX System Services session, you can start an agent as a started task from JCL without the need for an interactive session.

A started task is used because it runs under a specific user ID and is not affected by users logging off.

Note: Started tasks are typically run under an administrative user that might not have log-on privileges and so it is not possible to log on to the z/OS system as the user that the agent is running under. The **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, cannot be issued for that agent.

You can use the agent property **adminGroup** with Managed File Transfer agents on z/OS. You can define a security manager group, for example MFTADMIN and then add the started task userid and administrator TSO ids to this group. Edit the agent properties file and set the **adminGroup** property to be the name of this security manager group.

```
adminGroup=MFTADMIN
```

Members of this group can then issue the **fteStartAgent**, **fteStopAgent**, and **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, for the agent that is running as a started task.

For more information, see the **adminGroup** property in [The MFT agent.properties file](#).

As a Java application, an agent is a z/OS UNIX System Services application that you can run from JCL by using the BFGAGSTP member, from a generated Managed File Transfer command PDSE library data set for an agent. For more information about how to create an MFT command PDSE library data set, and customize it for the required agent, see [Creating an MFT Agent or Logger command data set](#).

Related concepts

[Enabling MFT agents to connect to remote z/OS queue managers](#)

Related reference

[“Stopping an MFT agent on z/OS” on page 227](#)

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

[The MFT agent.properties file](#)

MFT-Agenten auflisten

Sie können die bei einem bestimmten Warteschlangenmanager registrierten Managed File Transfer-Agenten über die Befehlszeile oder über die IBM MQ Explorerauflisten.

Informationen zu diesem Vorgang

Informationen zum Auflisten von Agenten über die Befehlszeile finden Sie im Abschnitt [Befehl 'fteListAgents'](#).

Um Agenten mithilfe von IBM MQ Explorer aufzulisten, klicken Sie in der Navigatoransicht unter dem Namen des Koordinationswarteschlangenmanagers auf **Agenten**.

Wenn ein Agent nicht mit dem Befehl **fteListAgents** aufgelistet oder nicht in IBM MQ Explorer angezeigt wird, verwenden Sie das Diagnoseablaufdiagramm im folgenden Abschnitt, um das Problem zu lokalisieren und zu beheben: [Was tun, wenn Ihr MFT-Agent nicht vom Befehl **fteListAgents** aufgelistet wird.](#)

Zugehörige Verweise

[fteListAgents: MFT-Agenten für einen Koordinationswarteschlangenmanager auflisten](#)

[MFT-Agentenstatuswerte](#)

[fteShowAgentDetails](#)

MFT-Agenten stoppen

Ein Managed File Transfer-Agent kann über die Befehlszeile gestoppt werden. Wenn Sie einen Agenten stoppen, legen Sie diesen still und ermöglichen dem Agenten gleichzeitig, seine aktuelle Dateiübertragung vor dem Stoppen abzuschließen. Zum sofortigen Stoppen eines Agenten können Sie auch den Parameter **-i** in der Befehlszeile angeben. Nach dem Stoppen des Agenten können Sie den Agenten erst dann wieder zum Übertragen von Dateien verwenden, wenn Sie diesen erneut starten.

Vorbereitende Schritte

Wenn die Namen der einem Warteschlangenmanager zugeordneten Agenten überprüft werden sollen, können Sie die Agenten über IBM MQ Explorer oder durch Eingabe des Befehls [fteListAgents](#) in der Befehlszeile auflisten.

Informationen zu diesem Vorgang

Hinweise zum Stoppen eines Agenten über die Befehlszeile finden Sie unter [fteStopAgent](#).

Wenn ein Agent über die **fteStopAgent** kontrolliert gestoppt wird, akzeptiert der Agent keine neuen verwalteten Übertragungsanforderungen und wartet, bis alle in Bearbeitung befindlichen Übertragungen abgeschlossen sind, bevor er sich selbst beendet. Um ab IBM MQ 9.3.0 anzuzeigen, dass sich der Agent noch in einem transienten Status befindet und daher noch nicht beendet und noch nicht erneut gestartet werden kann, wechselt der Agent in den Status STOPPING, bis alle laufenden Übertragungen abgeschlossen sind. Dieser Status wird in der Ausgabe der Befehle [fteListAgents](#) und [fteShowAgentDetails](#) sowie in [MFT REST API -Abfragen](#) und in der Ansicht **Agenten** des MFT -Plug-ins von IBM MQ Explorer angezeigt.

Windows Wenn Sie den Agenten für eine Ausführung als Windows-Dienst konfiguriert haben, wird bei Ausführung des Befehls **fteStopAgent** auch der Windows-Dienst beendet. Sie können den Agenten auch stoppen, indem Sie den Dienst über das Windows-Tool 'Dienste' stoppen. Weitere Informationen finden Sie im Abschnitt [„MFT-Agenten als Windows-Dienst starten“](#) auf Seite 221.

Zugehörige Tasks

[„MFT-Agenten starten“](#) auf Seite 220

Bevor Sie einen Managed File Transfer-Agenten für eine Dateiübertragung verwenden können, müssen Sie diesen aufrufen.

Zugehörige Verweise

[Statusinformationen zum MFT-Agenten](#)

[fteStopAgent](#)

[„Stopping an MFT agent on z/OS“](#) auf Seite 227

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

Stopping an MFT agent on z/OS

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

A started task is used because it runs under a specific user ID and is not affected by users logging off.

Note: Started tasks are typically run under an administrative user that might not have log-on privileges and so it is not possible to log on to the z/OS system as the user that the agent is running under. The **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, cannot be issued for that agent.

You can use the agent property **adminGroup** with Managed File Transfer agents on z/OS. You can define a security manager group, for example MFTADMIN and then add the started task userid and administrator TSO ids to this group. Edit the agent properties file and set the **adminGroup** property to be the name of this security manager group.

```
adminGroup=MFTADMIN
```

Members of this group can then issue the **fteStartAgent**, **fteStopAgent**, and **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, for the agent that is running as a started task.

For more information, see the **adminGroup** property in [The MFT agent.properties file](#).

Controlled agent shutdown by using the z/OS MODIFY command (F)

The **MODIFY** command allows you to stop an agent in a controlled way as an alternative to the **fteStopAgent** command. The agent completes any transfers currently in progress but the agent does not start any new transfers.

For example:

```
F job_name,APPL=STOP
```

where *job_name* is the job that the agent process is running under.

Immediate agent shutdown by using the z/OS STOP command (P)

The **STOP** command is equivalent to an immediate stop by using the **fteStopAgent** command with the **-i** parameter. The agent is stopped immediately even if the agent is currently transferring a file.

For example:

```
P job_name
```

where *job_name* is the job that the agent process is running under.

Related concepts

[Enabling MFT agents to connect to remote z/OS queue managers](#)

Related reference

“Starting an MFT agent on z/OS” on page 225

On z/OS, in addition to running the **fteStartAgent** command from a z/OS UNIX System Services session, you can start an agent as a started task from JCL without the need for an interactive session.

[The MFT agent.properties file](#)

Neue Dateiübertragung starten

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

Informationen zu diesem Vorgang

Informationen zum Starten einer neuen Dateiübertragung von der Befehlszeile aus finden Sie im Abschnitt zum [Befehl 'fteCreateTransfer'](#).

Gehen Sie wie folgt vor, um eine neue Dateiübertragung mit dem Assistenten **Neue verwaltete Dateiübertragung erstellen** in IBM MQ Explorer zu starten:

Vorgehensweise

1. Klicken Sie in der Navigatoransicht auf **Verwaltete Dateiübertragung**. In der Inhaltsansicht wird die **Verwaltete Dateiübertragungszentrale** angezeigt.
2. In der Navigatoransicht werden alle Ihre Koordinations-Warteschlangenmanager angezeigt. Erweitern Sie den Namen des Koordinationswarteschlangenmanagers, für den der Agent, den Sie für die Übertragung verwenden möchten, registriert ist. Wenn Sie derzeit mit einem anderen Koordinationswarteschlangenmanager verbunden sind als dem, den Sie für die Übertragung verwenden möchten, klicken Sie in der Navigatoransicht mit der rechten Maustaste auf den Namen des Koordinations-WS-Managers, und klicken Sie auf **Verbindung trennen**. Klicken Sie dann mit der rechten Maustaste auf den Namen des Koordinationswarteschlangenmanagers, den Sie verwenden möchten, und klicken Sie auf **Verbinden**.
3. Starten Sie den Assistenten **Neue verwaltete Dateiübertragung erstellen**; Sie haben zwei Möglichkeiten:
 - a) Klicken Sie mit der rechten Maustaste auf einen beliebigen der folgenden Knoten in der Navigatoransicht: den relevanten Koordinations-Warteschlangenmanager, **Übertragungsschablonen**, **Übertragungsprotokoll** oder **Anstehende Übertragungen**. Klicken Sie anschließend auf **Neue Übertragung**, um den Assistenten zu starten.
 - b) Klicken Sie auf **Datei > Neu > Sonstige > Assistenten für verwaltete Dateiübertragung > Assistent für neue Übertragung**
4. Befolgen Sie die Anweisungen in der Assistentenanzeige. Darüberhinaus steht für jede Anzeige eine kontextabhängige Hilfe zur Verfügung. Zum Zugriff auf die kontextabhängige Hilfe unter Windows drücken Sie F1. Unter Linux drücken Sie **Ctrl+F1** oder **Shift+F1**.

Zugehörige Konzepte

[„Definitionsdateien für die Übertragung verwenden“](#) auf Seite 229

Zur Erstellung einer Dateiübertragung können Sie eine Definitionsdatei für die Übertragung angeben. Bei der Definitionsdatei der Übertragung handelt es sich um eine XML-Datei, die einen Teil oder alle der Informationen definiert, die zur Erstellung der Übertragung erforderlich sind.

Zugehörige Tasks

[„Geplante Dateiübertragung erstellen“](#) auf Seite 231

Sie können eine neue Dateiübertragung entweder über IBM MQ Explorer oder über die Befehlszeile planen. Die geplante Übertragung kann einzelne Dateien oder mehrere Dateien in einer Gruppe enthalten. Sie können eine geplante Dateiübertragung einmal ausführen oder die Übertragung mehrmals wiederholen.

[„Dateiübertragung auslösen“](#) auf Seite 233

Sie können für eine Dateiübertragung bestimmte Auslöserbedingungen festlegen, die erfüllt sein müssen, bevor die Übertragung ausgeführt werden kann. Wenn die Auslöserbedingungen nicht erfüllt werden, wird die Dateiübertragungsanforderung nicht ausgeführt, und es wird eine Protokollnachricht übermittelt, um das Fehlschlagen der Übertragung zu protokollieren. Die Dateiübertragungsanforderung wird dann verworfen. Sie können beispielsweise eine Dateiübertragung erstellen, die nur erfolgt, wenn eine benannte Datei auf demselben System wie der Quellenagent eine bestimmte Größe überschreitet oder wenn sich

eine bestimmte benannte Datei auf demselben System wie der Quellenagent befindet. Eine ausgelöste Dateiübertragung kann entweder über IBM MQ Explorer oder über die Befehlszeile konfiguriert werden.

„Zeitlimit für die Wiederherstellung von blockierten Übertragungen festlegen“ auf Seite 345

Sie können ein Zeitlimit für die Übertragungswiederherstellung für blockierte Dateiübertragungen festlegen, die für alle Übertragungen eines Quellenagenten angewendet werden. Sie können auch ein Zeitlimit für die Übertragungswiederherstellung für eine einzelne Übertragung festlegen. Wenn Sie eine bestimmte Zeitdauer (in Sekunden) festlegen, während der ein Quellenagent versucht, eine blockierte Dateiübertragung wiederherzustellen, und die Übertragung noch nicht erfolgreich war, wenn der Agent dieses Zeitlimit erreicht, schlägt die Übertragung fehl.

Zugehörige Verweise

fteCreateTransfer: [Neue Dateiübertragung starten](#)
[Nachrichtenformat für Dateiübertragungsanforderungen](#)
[Richtlinien für das Übertragen von Dateien](#)

Definitionsdateien für die Übertragung verwenden

Zur Erstellung einer Dateiübertragung können Sie eine Definitionsdatei für die Übertragung angeben. Bei der Definitionsdatei der Übertragung handelt es sich um eine XML-Datei, die einen Teil oder alle der Informationen definiert, die zur Erstellung der Übertragung erforderlich sind.

Definitionsdateien für Übertragungen sind hilfreich, wenn Sie in einer einzelnen Übertragungsoperation mehrere Quellendateien und Zieldateien angeben möchten. Damit können Sie mit diesen Dateien komplexe Dateiübertragungen durchführen. Die Definitionsdatei für die Übertragung kann wiederverwendet und für die gemeinsame Nutzung freigegeben werden.

Sie können zwei Formate für eine Übertragungsdefinitionsdatei verwenden, und während diese Formate geringfügig variieren, müssen beide Formate dem `FileTransfer.xsd`-Schema entsprechen. Dieses Schema finden Sie im Verzeichnis `samples\schema` der Installation von Managed File Transfer .

Die folgenden beiden Formate von Definitionsdateien für die Übertragung werden unterstützt:

- Eine Definition der Quellen- und Zieldateien für eine Übertragung. Diese Definition verwendet ein Element vom Typ **transferSpecifications** als Stammverzeichnis.
- Eine Definition der gesamten Übertragung, einschließlich der Quellen- und Zieldateien sowie der Quellen- und Zielagenten. Diese Definition verwendet ein Element vom Typ **request** als Stammverzeichnis.
 - Dateien dieses Formats können unter Angabe des Parameters **-gt** mit dem Befehl **fteCreateTransfer** generiert werden.

Das folgende Beispiel veranschaulicht das Format einer Definitionsdatei für Übertragungen, in der nur die Quellen- und Zieldateien einer Übertragung angegeben sind:

```
<?xml version="1.0" encoding="UTF-8"?>
<transferSpecifications xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <item checksumMethod="MD5" mode="text">
    <source recursive="false" disposition="leave">
      <file>textTransferTest.txt</file>
    </source>
    <destination type="directory" exist="overwrite">
      <file>c:\targetfiles</file>
    </destination>
  </item>
</transferSpecifications>
```

Zur Übergabe dieses Formats der Definitionsdatei für die Übertragung müssen Sie die Quellen- und Zielagenten in der Befehlszeile angeben:

```
fteCreateTransfer -sa AGENT1 -sm agent1qm -da AGENT2 -dm agent2qm -td
c:\definitions\example1.xml
```

Im folgenden Beispiel wird das Format einer Definitionsdatei für die Übertragung gezeigt, in dem alle für eine Übertragung erforderlichen Informationen angegeben werden:

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="3.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>fteuser</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="agent1qm"/>
    <destinationAgent agent="AGENT2" QMgr="agent2qm"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:\sourcefiles\*.jpg</file>
        </source>
        <destination type="directory" exist="error">
          <file>/targetfiles/images</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Sie können eine Datei dieses Formats unter Angabe des Parameters **-gt** im Befehl **fteCreateTransfer** generieren. Bei der Übergabe einer Definitionsdatei für die Übertragung mit diesem Format müssen Sie in der Befehlszeile keine weiteren Angaben machen:

```
fteCreateTransfer -td c:\definitions\example2.xml
```

Sie können die Angaben zum Quellen- und Zielagenten über die Befehlszeile ändern, indem Sie neben der Definitionsdatei für die Übertragung noch die normalen Parameter übergeben. For example:

```
fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml
```

In diesem Beispiel wird der in der Übertragungsdefinitionsdatei definierte Zielagent mithilfe der Befehlszeilenoptionen in **AGENT9**, der Zielwarteschlangenmanager in **agent9qm** geändert.

Beide der beschriebenen Formate können ein oder mehrere <item>-Elemente enthalten. Weitere Informationen zu dem Element <item>finden Sie unter Nachrichtenformat für Dateiübertragungsanforderung. All diese Übertragungselemente definieren ein Paar aus Quellen- und Zieldatei mit zusätzlichen Attributen für die Steuerung des Übertragungsverhaltens. Sie können beispielsweise folgendes Verhalten festlegen:

- Ob bei der Übertragung eine Kontrollsumme verwendet wird
- Ob die Übertragung im Text- oder Binärmodus erfolgt
- Ob die Quelldatei nach Abschluss der Übertragung gelöscht werden soll
- Ob die Zieldatei (sofern vorhanden) überschrieben werden soll

Die Nutzung von Definitionsdateien für die Übertragung hat den Vorteil, dass Sie zusätzliche Optionen angeben können, die bei einer Ausführung über die Befehlszeile nicht zur Verfügung stehen. Bei Übertragungen von Nachrichten in eine Datei können Sie das Attribut 'groupId' mithilfe einer Übertragungsdefinitionsdatei angeben. Dieses Attribut gibt die IBM MQ-Gruppen-ID der Nachrichten an, die aus der Warteschlange gelesen werden. Ein weiterer Vorteil von Übertragungsdefinitionsdateien besteht darin, dass für jedes Dateipaar andere Optionen angegeben werden können. So können Sie beispielsweise für die einzelnen Dateien angeben, ob eine Prüfsumme verwendet oder die Datei im Text- oder Binärmodus übertragen werden soll. Wenn Sie die Befehlszeile verwenden, gelten die Optionen für alle Dateien in einer Übertragung.

For example:

```
<item checksumMethod="none" mode="binary">
  <source disposition="leave">
    <file>c:\sourcefiles\source1.doc</file>
  </source>
  <destination type="file" exist="error">
    <file>c:\destinationfiles\destination1.doc</file>
  </destination>
</item>

<item checksumMethod="MD5" mode="text">
  <source disposition="delete">
    <file>c:\sourcefiles\source2.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file encoding="UTF8" EOL="CRLF">c:\destinationfiles\destination2.txt</file>
  </destination>
</item>

<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>c:\originfiles\source3.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file>c:\targetfiles\destination3.txt</file>
  </destination>
</item>
```

z/OS Mithilfe von Elementen können Sie eine Datei aus einem verteilten System in ein z/OS-System übertragen:

```
z/OS
<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>textTransferTest.txt</file>
  </source>
  <destination type="dataset" exist="overwrite">
    <file encoding="IBM-1047">//TEXT.TRANS.TEST</file>
  </destination>
</item>
```

z/OS In diesem Beispiel wird die Datei `textTransferTest.txt` von dem Quellenagenten in den Datensatz `//TEXT.TRANS.TEST` auf dem Zielagenten im Textmodus übertragen. Bei der Übertragung werden die Quelldaten von der Standardcodierung des Quellenagenten (es ist kein Quellencodierungsattribut angegeben) in Codepage IBM-1047 konvertiert.

Geplante Dateiübertragung erstellen

Sie können eine neue Dateiübertragung entweder über IBM MQ Explorer oder über die Befehlszeile planen. Die geplante Übertragung kann einzelne Dateien oder mehrere Dateien in einer Gruppe enthalten. Sie können eine geplante Dateiübertragung einmal ausführen oder die Übertragung mehrmals wiederholen.

Informationen zu diesem Vorgang

Sie können einen Zeitplan für die Dateiübertragung einrichten, der einmal ausgeführt werden soll, oder in den folgenden Intervallen auftreten:

- Jede Minute
- Stündlich
- Täglich
- Wöchentlich
- Monatlich
- Jähr.

Anschließend können Sie die Vorkommen angeben, die an den folgenden Punkten gestoppt werden sollen:

- Zu einem definierten Zeitpunkt und zu einem definierten Datum
- Nach einer definierten Anzahl von Vorkommen

Alternativ können Sie angeben, dass die Vorkommen für immer fortgesetzt werden sollen.

Wenn eine geplante Übertragung jeden Tag zur gleichen Zeit ausgeführt wird, verwenden Sie das Attribut **adjustScheduleTimeForDaylightSaving** in der Agenteneigenschaftendatei, um die Zeit anzupassen, zu der der Zeitplan stattfindet, wenn sich die Systemzeit ändert. Weitere Informationen finden Sie unter [Die MFT agent.properties-Datei](#).

Um eine neue geplante Dateiübertragung über die Befehlszeile zu erstellen, verwenden Sie die Planungsparameter (**-tb**, **-ss**, **-oi**, **-of**, **-oc** und **-es**) für den Befehl "[fteCreateTransfer](#)".

So erstellen Sie eine neue geplante Dateiübertragung mithilfe des Assistenten **Neuen Managed File Transfer erstellen** in IBM MQ Explorer:

Vorgehensweise

1. Klicken Sie in der Navigatoransicht auf **Verwaltete Dateiübertragung**. In der Inhaltsansicht wird die **Verwaltete Dateiübertragungszentrale** angezeigt.
2. In der Navigatoransicht werden alle Ihre Koordinations-Warteschlangenmanager angezeigt. Erweitern Sie den Namen des Koordinationswarteschlangenmanagers, für den der Agent, den Sie für die Übertragung verwenden möchten, registriert ist. Wenn Sie derzeit mit einem anderen Koordinationswarteschlangenmanager verbunden sind als dem, den Sie für die Übertragung verwenden möchten, klicken Sie in der Navigatoransicht mit der rechten Maustaste auf den Namen des Koordinations-WS-Managers, und klicken Sie auf **Verbindung trennen**. Klicken Sie dann mit der rechten Maustaste auf den Namen des Koordinationswarteschlangenmanagers, den Sie verwenden möchten, und klicken Sie auf **Verbinden**.
3. Starten Sie den Assistenten **Neue verwaltete Dateiübertragung erstellen** mit einer der folgenden Methoden:
 - a) Klicken Sie mit der rechten Maustaste auf einen beliebigen der folgenden Knoten in der Navigatoransicht: den relevanten Koordinations-Warteschlangenmanager, **Übertragungsschablonen**, **Übertragungsprotokoll** oder **Anstehende Übertragungen**. Klicken Sie anschließend auf **Neue Übertragung**, um den Assistenten zu starten.
 - b) Klicken Sie auf **Datei > Neu > Sonstige > Assistenten für verwaltete Dateiübertragung > Assistent für neue Übertragung**
4. Befolgen Sie die Anweisungen in der Assistentenanzeige. Stellen Sie sicher, dass Sie das Kontrollkästchen **Zeitplanübertragung aktivieren** ausgewählt haben, und geben Sie Ihre Zeitplandetails auf der Registerkarte **Zeitplan** ein. Geplante Dateiübertragungen beginnen innerhalb einer Minute nach der Startzeit des Zeitplans, wenn es keine Probleme gibt, die sich auf die Übertragung auswirken könnten. Es kann zum Beispiel Probleme mit Ihrem Netz oder Agenten geben, die die geplante Übertragung verhindern. Für jede Anzeige steht eine kontextabhängige Hilfe zur Verfügung. Zum Zugriff auf die kontextabhängige Hilfe unter Windows drücken Sie F1. Unter Linux drücken Sie Ctrl+F1 oder Shift+F1.

Ergebnisse

Informationen zu den Nachrichten, die an geplanten Dateiübertragungen beteiligt sind, finden Sie in [Geplante Dateiübertragungsprotokollnachrichtensformate](#).

Arbeiten mit anstehenden Dateiübertragungen

In IBM MQ Explorer können Sie geplante anstehende Dateiübertragungen anzeigen. Das Fenster **Anstehende Übertragungen** zeigt alle anstehenden Übertragungen an, die für den Koordinations-Warteschlangenmanager registriert sind, mit dem Sie aktuell verbunden sind.


Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um den Status einer geplanten, aber noch nicht gestarteten Dateiübertragung anzuzeigen:

Vorgehensweise

1. Erweitern Sie den Knoten **Verwaltete Dateiübertragung** in der Navigatoransicht. In der Inhaltsansicht wird die **Verwaltete Dateiübertragungszentrale** angezeigt.
2. In der Navigatoransicht werden alle Ihre Koordinations-Warteschlangenmanager angezeigt. Erweitern Sie den Namen des Koordinations-Warteschlangenmanagers, den Sie für die geplante Übertragung verwendet haben. Wenn Sie den Koordinations-Warteschlangenmanager ändern möchten, mit dem Sie verbunden sind, klicken Sie mit der rechten Maustaste auf den Namen des Koordinations-Warteschlangenmanagers, den Sie in der Navigatoransicht verwenden möchten, und klicken Sie auf **Verbinden**.
3. Klicken Sie auf **Anstehende Übertragungen**. Das Fenster **Anstehende Übertragungen** wird in der Inhaltsansicht angezeigt.
4. Das Fenster **Anstehende Übertragungen** zeigt folgende Details Ihrer geplanten Dateiübertragungen an:
 - a) **Name** Die Nummer der geplanten Dateiübertragung. Diese Nummer wird automatisch zugewiesen.
 - b) **Quelle** Der Name des Quellenagenten.
 - c) **Quellendatei** Der Name der auf ihr Hostsystem zu übertragenden Datei.
 - d) **Ziel** Der Name des Zielagenten.
 - e) **Zieldatei** Der Name der Datei, nachdem diese auf das Zielsystem übertragen wurde.
 - f) **Geplanter Start (ausgewählte Zeitzone)** Das Datum und die Uhrzeit, zu dem/der die Dateiübertragung geplant ist, entsprechend der vom Administrator ausgewählten Zeitzone. Klicken Sie zum Ändern der angezeigten Zeitzone auf **Fenster > Benutzervorgaben > IBM MQ Explorer > Managed File Transfer** und wählen Sie eine alternative Zeitzone in der Liste **Zeitzone:** aus. Klicken Sie auf **OK**.
 - g) **Wiederholen alle** Wenn die geplante Übertragung wiederholt werden soll, ist dies das Intervall, in dem die Übertragung wiederholt wird, ausgedrückt in einer Zahl.
 - h) **Wiederholungstyp** Wenn Sie ausgewählt haben, dass die geplante Übertragung wiederholt werden soll: der Typ des Wiederholungsintervalls, das Sie für die Dateiübertragung angegeben haben. Der Typ kann einer der folgenden Werte sein: Minuten, Stunden, Tage, Wochen, Monate oder Jahre.
 - i) **Wiederholen bis** Wenn Sie ausgewählt haben, dass die geplante Übertragung wiederholt werden soll, finden Sie hier die Details zu dem Zeitpunkt, an dem die wiederholte Dateiübertragung gestoppt werden soll. Dies kann z. B. ein bestimmtes Datum mit Uhrzeit sein oder eine bestimmte Häufigkeit.

Ergebnisse

Sie können den Inhalt des Fensters **Anstehende Übertragungen** aktualisieren, indem Sie in der Symbolleiste der Inhaltsansicht auf 'Aktualisieren'  klicken.

Zum Abbrechen einer anstehenden Dateiübertragung, klicken Sie mit der rechten Maustaste auf die entsprechende Übertragung, und klicken Sie auf **Abbrechen**. Wenn Sie eine Übertragung abbrechen, wird die Dateiübertragungsanforderung vollständig verworfen.

Dateiübertragung auslösen

Sie können für eine Dateiübertragung bestimmte Auslöserbedingungen festlegen, die erfüllt sein müssen, bevor die Übertragung ausgeführt werden kann. Wenn die Auslöserbedingungen nicht erfüllt werden, wird die Dateiübertragungsanforderung nicht ausgeführt, und es wird eine Protokollnachricht übermittelt, um das Fehlschlagen der Übertragung zu protokollieren. Die Dateiübertragungsanforderung wird dann

verworfen. Sie können beispielsweise eine Dateiübertragung erstellen, die nur erfolgt, wenn eine benannte Datei auf demselben System wie der Quellenagent eine bestimmte Größe überschreitet oder wenn sich eine bestimmte benannte Datei auf demselben System wie der Quellenagent befindet. Eine ausgelöste Dateiübertragung kann entweder über IBM MQ Explorer oder über die Befehlszeile konfiguriert werden.

Informationen zu diesem Vorgang

Sie können eine Ressource kontinuierlich auf die Erfüllung einer Auslöserbedingung überwachen. Weitere Informationen zur Ressourcenüberwachung finden Sie im Abschnitt [„MFT-Ressourcen überwachen“](#) auf Seite 239.

Sie können drei unterschiedliche Auslöserbedingungen festlegen. Die Bedingungen lauten wie folgt:

- Eine bestimmte Datei befindet sich auf demselben System wie der Quellenagent.
- Eine bestimmte Datei befindet sich nicht auf demselben System wie der Quellenagent.
- Eine bestimmte Datei, die sich auf demselben System wie der Quellenagent befindet, überschreitet eine vorgegebene Größe (diese kann in Byte, KB, MB oder GB angegeben werden). Diese Maßeinheiten verwenden die 2¹⁰-Konvention, sodass beispielsweise 1 KB 1024 Byte entspricht und 1 MB 1024 KB.

Es gibt zwei Möglichkeiten, die oben aufgeführten Auslösertypen zu kombinieren:

- Bei Angabe nur einer Bedingung können Sie mehrere Dateien angeben, die sich auf demselben System wie der Quellenagent befinden. Dadurch wird die Übertragung ausgelöst, sobald eine der angegebenen Dateien die Bedingung erfüllt (boolescher Operator ODER).
- Sie können mehrere Bedingungen angeben. Die Übertragung wird in diesem Fall nur dann ausgelöst, wenn alle Bedingungen erfüllt sind (boolescher Operator UND).

Sie können eine ausgelöste Übertragung auch mit einer geplanten Übertragung kombinieren. Weitere Informationen finden Sie unter [Erstellen einer geplanten Dateiübertragung](#). In diesem Fall werden die Auslöserbedingungen zu dem Zeitpunkt ausgewertet, zu dem der Plan starten soll. Bei einer wiederholten Übertragung werden die Auslöserbedingungen zu jedem Zeitpunkt ausgewertet, zu dem der Plan starten soll.

Ausgelöste Übertragungen werden auf Protokollbridge-Agenten nicht unterstützt.

Um eine ausgelöste Dateiübertragung über die Befehlszeile zu erstellen, verwenden Sie den Parameter **-tr** im Befehl `fteCreateTransfer`.

So erstellen Sie eine geplante Dateiübertragung mithilfe des Assistenten **Neuen Managed File Transfer erstellen** in IBM MQ Explorer:

Vorgehensweise

1. Klicken Sie in der Navigatoransicht auf **Verwaltete Dateiübertragung**. In der Inhaltsansicht wird die **Verwaltete Dateiübertragungszentrale** angezeigt.
2. In der Navigatoransicht werden alle Ihre Koordinations-Warteschlangenmanager angezeigt. Erweitern Sie den Namen des Koordinations-Warteschlangenmanagers, den Sie für die geplante Übertragung verwendet haben. Wenn Sie den Koordinations-Warteschlangenmanager ändern möchten, mit dem Sie verbunden sind, klicken Sie mit der rechten Maustaste auf den Namen des Koordinations-Warteschlangenmanagers, den Sie in der Navigatoransicht verwenden möchten, und klicken Sie auf **Verbinden**.
3. Starten Sie den Assistenten **Neue verwaltete Dateiübertragung erstellen**; Sie haben zwei Möglichkeiten:
 - a) Klicken Sie mit der rechten Maustaste auf einen beliebigen der folgenden Knoten in der Navigatoransicht: den relevanten Koordinations-Warteschlangenmanager, **Übertragungsschablonen**, **Übertragungsprotokoll** oder **Anstehende Übertragungen**. Klicken Sie anschließend auf **Neue Übertragung**, um den Assistenten zu öffnen.
 - b) Klicken Sie auf **Datei > Neu > Sonstige > Assistenten für verwaltete Dateiübertragung > Assistent für neue Übertragung**


4. Befolgen Sie die Anweisungen in der Assistentenanzeige. Stellen Sie sicher, dass das Kontrollkästchen **Ausgelöste Übertragung aktivieren** auf der Registerkarte **Auslöser** ausgewählt wurde, um die Auslöserfunktion zu definieren. Für jede Anzeige steht eine kontextabhängige Hilfe zur Verfügung. Zum Zugriff auf die kontextabhängige Hilfe unter Windows drücken Sie F1. Drücken Sie unter Linux die Tastenkombination **Ctrl+F1** oder Shift+F1.

Überwachen von Dateiübertragungen, die sich in Bearbeitung befinden

Sie können eine laufende Dateiübertragung über die Registerkarte **Verwaltete Dateiübertragung-Aktueller Übertragungsfortschritt** in IBM MQ Explorer überwachen. Diese Dateiübertragung kann aus IBM MQ Explorer oder von der Befehlszeile aus gestartet worden sein. Auf der Registerkarte wird auch der Fortschritt geplanter Übertragungen an der Stelle angezeigt, an der die geplanten Übertragungen beginnen.

Informationen zu diesem Vorgang

Wenn Sie mit IBM MQ Explorer Übertragungen überwachen möchten, die einem Koordinationswarteschlangenmanager auf einem fernen System zugeordnet sind, befolgen Sie die Anweisungen im Abschnitt [„IBM MQ Explorer für die Überwachung eines fernen Koordinationswarteschlangenmanagers konfigurieren“](#) auf Seite 236.

Wenn Sie IBM MQ Explorer stoppen und erneut starten, bleiben vorherige Dateiübertragungsinformationen nicht erhalten. Beim Neustart werden die Informationen zu vergangenen Übertragungen über die Registerkarte **Aktueller Übertragungsfortschritt** gelöscht. Sie können abgeschlossene Übertragungen jederzeit über **Abgeschlossene Übertragungen entfernen**  löschen, wenn IBM MQ Explorer geöffnet ist.


Vorgehensweise


Wenn Sie über IBM MQ Explorer oder die Befehlszeile eine neue Dateiübertragung gestartet haben, können Sie den Fortschritt Ihrer Übertragung auf der Registerkarte **Aktueller Übertragungsfortschritt** überwachen. Die folgenden Informationen werden für jede Übertragung in Bearbeitung angezeigt:

- a) **Quelle**. Der Name des Agenten, der zum Übertragen der Datei aus dem Quellsystem verwendet wird.
- b) **Ziel**. Der Name des Agenten, der zum Empfangen der Datei auf dem Zielsystem verwendet wird.
- c) **Aktuelle Datei**. Der Name der Datei, die gerade übertragen wird. Der Teil der bereits übertragenen Einzeldatei wird in B, KiB, MiB angezeigt. GiB oder TiB zusammen mit der Gesamtgröße der Datei in runden Klammern. Die Maßeinheit, die angezeigt wird, hängt von der Größe der Datei ab.
B ist Byte pro Sekunde. KiB/s sind Kibibyte pro Sekunde, wobei 1 Kibibyte 1024 Byte entspricht. MiB/s ist Mebibyte pro Sekunde, wobei 1 Mebibyte gleich 1 048 576 Byte ist. GiB/s sind Gibibyte pro Sekunde, wobei 1 Gibibyte 1 073 741 824 Byte entspricht. TiB/s ist Tebibyte pro Sekunde, wobei 1 Tebibyte gleich 1 099 511 627 776 Byte ist.
- d) **Dateinummer**. Wenn Sie mehr als eine Datei übertragen, stellt diese Zahl dar, wie weit die gesamte Gruppe von Dateien die Übertragung ist.
- e) **Fortschritt**. In der Fortschrittsleiste wird angezeigt, wie die aktuelle Dateiübertragung als Prozentsatz abgeschlossen ist.
- f) **Rate**. Die Rate, mit der die Datei in KiB/s übertragen wird (Kibibyte pro Sekunde, wobei 1 Kibibyte 1024 Byte entspricht.)
- g) **Gestartet (ausgewählte Zeitzone)**. Die Zeit, zu der die Dateiübertragung gestartet wurde, die in der ausgewählten Zeitzone des Administrators dargestellt wurde. Klicken Sie zum Ändern der angezeigten Zeitzone auf **Fenster > Benutzervorgaben > IBM MQ Explorer > Managed File Transfer** und wählen Sie eine alternative Zeitzone in der Liste **Zeitzone:** aus. Klicken Sie auf **OK**.

Wenn die Übertragung einen Wiederherstellungsstatus beim Übertragen der Datei eingibt, wird die Uhrzeit der gestarteten Zeit aktualisiert, um die Zeit wiederzugeben, die die Dateiübertragung wieder aufgenommen hat.

Ergebnisse

Diese Registerkarte aktualisiert ihre Informationen regelmäßig automatisch, um jedoch eine aktualisierte Ansicht zu erzwingen, was auf der Registerkarte **Current Transfer Progress** (Aktueller Übertragungsfortschritt) angezeigt wird, klicken Sie in der Symbolleiste der Inhaltsansicht auf **Refresh** (Aktualisieren) .

Wenn Sie Dateiübertragungen von der Registerkarte **Aktueller Übertragungsfortschritt** löschen möchten, klicken Sie in der Symbolleiste der Inhaltsansicht auf **Abgeschlossenes Umlagern entfernen** . Wenn Sie auf diese Schaltfläche klicken, werden die Dateiübertragungsdetails nur von der Registerkarte entfernt. Es wird keine aktuelle oder geplante Übertragung gestoppt oder abgebrochen.

Wenn Sie nach dem Schließen zur Registerkarte **Aktueller Übertragungsfortschritt** zurückkehren möchten, können Sie die Registerkarte durch Klicken auf **Fenster > Ansicht anzeigen > Sonstige > Sonstige > Verwaltete Dateiübertragung - Aktueller Übertragungsfortschritt** erneut aufrufen. Klicken Sie auf **OK**.

Nächste Schritte

Darüber hinaus ist es möglich, Anwendungen für die Überwachung angepasster Dateiübertragungen zu entwickeln. Dies kann erreicht werden, indem Sie eine Subskription für das entsprechende Managed File Transfer-Verwaltungsthema erstellen (entweder über das Programm oder über die Verwaltung), und die Überwachungsanwendung kann dann Veröffentlichungen zu Managed File Transfer-Dateiübertragungsaktivitäten zu dem Thema empfangen. Weitere Informationen zum Subskriptionsthema und zum Publizierungsnachrichtenformat finden Sie im Abschnitt [Beispiele für Fortschrittsnachrichten zu Dateiübertragungen](#).

Zugehörige Tasks

[„IBM MQ Explorer für die Überwachung eines fernen Koordinationswarteschlangenmanagers konfigurieren“ auf Seite 236](#)

Mit IBM MQ Explorer können Sie Dateiübertragungen überwachen, die einem Koordinationswarteschlangenmanager, der auf einem fernen System aktiv ist, zugeordnet sind. Sie benötigen ein System, auf dem IBM MQ Explorerausgeführt werden kann. Die Komponente IBM MQ Explorer muss installiert werden, damit eine Verbindung zum fernen Koordinationswarteschlangenmanager hergestellt werden kann.

[„Status von Dateiübertragungen im Übertragungsprotokoll anzeigen“ auf Seite 237](#)

Sie können im **Übertragungsprotokoll** in IBM MQ Explorer die Details zu Dateiübertragungen ansehen. Dabei spielt es keine Rolle, ob die Übertragungen über die Befehlszeile oder über IBM MQ Explorer gestartet wurden. Sie können den Inhalt des **Übertragungsprotokolls** anpassen.

IBM MQ Explorer für die Überwachung eines fernen Koordinationswarteschlangenmanagers konfigurieren

Mit IBM MQ Explorer können Sie Dateiübertragungen überwachen, die einem Koordinationswarteschlangenmanager, der auf einem fernen System aktiv ist, zugeordnet sind. Sie benötigen ein System, auf dem IBM MQ Explorerausgeführt werden kann. Die Komponente IBM MQ Explorer muss installiert werden, damit eine Verbindung zum fernen Koordinationswarteschlangenmanager hergestellt werden kann.

Informationen zu diesem Vorgang

Voraussetzungen: Berechtigung zum Herstellen einer Verbindung zum fernen Koordinations-WS-Manager durch Konfigurieren des Warteschlangenmanagers, um Fernverbindungen zu ermöglichen.

Weitere Informationen über die entsprechende Konfiguration finden Sie in den Abschnitten [Verbindung zu einem Warteschlangenmanager im Clientmodus mit Kanalauthentifizierung herstellen](#) und [Berechtigungen für MFT-spezifische Ressourcen verwalten](#).

Damit Warteschlangenmanager und die Dateiübertragungen zwischen Agenten auf einem System überwacht werden können, bei dem es sich nicht um ein Windows- oder Linux-System handelt, müssen Sie IBM MQ Explorer wie folgt konfigurieren, damit eine Verbindung zum fernen System hergestellt werden kann.

Vorgehensweise

1. Starten Sie den lokalen IBM MQ Explorer.
2. Klicken Sie in IBM MQ Explorer mit der rechten Maustaste auf den Ordner **Managed File Transfer** und wählen Sie **New configuration** (Neue Konfiguration) aus.
3. Fahren Sie mit dem Assistenten fort, und wählen Sie den Warteschlangenmanager "Koordination" und "Befehle" aus. Definieren Sie dann einen Namen für die Konfiguration.
4. Klicken Sie auf **Fertigstellen**, um die Definition abzuschließen.
5. Wenn die Definition abgeschlossen ist, klicken Sie mit der rechten Maustaste auf die Definition, und wählen Sie **Verbinden** aus.

Ergebnisse

Starten Sie jetzt IBM MQ Explorer und überwachen Sie damit Übertragungsaktivitäten im Managed File Transfer-Netz, das dem Koordinationswarteschlangenmanager zugeordnet ist.

Zugehörige Tasks

„Überwachen von Dateiübertragungen, die sich in Bearbeitung befinden“ auf Seite 235

Sie können eine laufende Dateiübertragung über die Registerkarte **Verwaltete Dateiübertragung-Aktueller Übertragungsfortschritt** in IBM MQ Explorer überwachen. Diese Dateiübertragung kann aus IBM MQ Explorer oder von der Befehlszeile aus gestartet worden sein. Auf der Registerkarte wird auch der Fortschritt geplanter Übertragungen an der Stelle angezeigt, an der die geplanten Übertragungen beginnen.


„Status von Dateiübertragungen im Übertragungsprotokoll anzeigen“ auf Seite 237

Sie können im **Übertragungsprotokoll** in IBM MQ Explorer die Details zu Dateiübertragungen ansehen. Dabei spielt es keine Rolle, ob die Übertragungen über die Befehlszeile oder über IBM MQ Explorer gestartet wurden. Sie können den Inhalt des **Übertragungsprotokolls** anpassen.

Status von Dateiübertragungen im Übertragungsprotokoll anzeigen

Sie können im **Übertragungsprotokoll** in IBM MQ Explorer die Details zu Dateiübertragungen ansehen. Dabei spielt es keine Rolle, ob die Übertragungen über die Befehlszeile oder über IBM MQ Explorer gestartet wurden. Sie können den Inhalt des **Übertragungsprotokolls** anpassen.

Vorgehensweise

1. Erweitern Sie den Knoten **Verwaltete Dateiübertragung** in der Navigatoransicht und erweitern Sie dann den Namen des Koordinations-Warteschlangenmanagers, dessen Übertragungsprotokoll Sie sich anzeigen lassen wollen.
2. Klicken Sie in der Navigatoransicht auf **Übertragungsprotokoll**. Das **Übertragungsprotokoll** wird in der Inhaltsansicht angezeigt.
3. Das Fenster **Übertragungsprotokoll** enthält die folgenden Details zu Ihrer Dateiübertragung:
 - a) **Quelle** Der Name des Agenten auf dem System, auf dem sich die Datei befindet.
 - b) **Ziel** Der Name des Agenten auf dem System, auf das Sie die Datei übertragen möchten.
 - c) **Beendigungsstatus** Der Status der Dateiübertragung. Der Status kann einer der folgenden Werte sein: "Gestartet", "In Bearbeitung", "Erfolgreich", "Teilweise erfolgreich", "Abgebrochen", oder "Fehlgeschlagen".
 - d) **Eigner** Die Benutzer-ID auf dem Host, von dem die Übertragungsanforderung abgesendet wurde.
 - e) **Gestartet (ausgewählte Zeitzone)** Der Zeitpunkt (Datum und Uhrzeit), zu dem die Dateiübertragung vom Managed File Transfer-Agent akzeptiert wurde, dargestellt in der ausgewählten Zeitzone des Administrators. Klicken Sie zum Ändern der angezeigten Zeitzone auf **Fenster > Benutzervorgaben > IBM MQ Explorer > Managed File Transfer** und wählen Sie eine alternative Zeitzone in der Liste **Zeitzone:** aus. Klicken Sie auf **OK**.
 - f) **Statusaufzeichnung (ausgewählte Zeitzone)** (Diese Spalte wird standardmäßig nicht angezeigt. Sie können sie im Fenster **Spalten des Übertragungsprotokolls konfigurieren** ) anzeigen. Der



Zeitpunkt (Datum und Uhrzeit) in der vom Administrator ausgewählten Zeitzone, zu dem der Beendigungsstatus aufgezeichnet wurde.

- g) **Jobname** Eine vom Benutzer über den Parameter **-jn** von **fteCreateTransfer** oder in einem Ant-Script angegebene Kennung.
- h) **Übertragungs-ID** Die eindeutige Kennung für die Dateiübertragung.
- i) **Connect: Direct** (Verbinden: Direkt) Details zu **Process Number** (Prozessnummer), **Process Name** (Prozessname), **Primary Node** (Primärknoten), **Secondary Node** (Sekundärknoten), **Source Type** (Quellentyp) und **Destination Type** (Zieltyp) werden aufgelistet.

Ergebnisse

Anmerkung: Das interne Format des Übertragungsprotokolls wurde für APAR IC99545 in IBM MQ 8.0.0 Fix Pack 1 geändert. Wenn daher für IBM MQ Explorer ein Upgrade auf V8.0.0.1 oder höher vorgenommen wurde, anschließend aber V8.0.0.0 wiederhergestellt wurde, wird für Übertragungen, die stattfanden, solange IBM MQ Explorer die Version V8.0.0.1 hatte, keine Prüfungs-XML-Datei angezeigt. Die XML-Anzeige im Fenster **Eigenschaften** für diese Übertragung enthält ein leeres Textfeld.

Um weitere Details zu einer abgeschlossenen Übertragung anzuzeigen, erweitern Sie die Übertragung, an der Sie interessiert sind, indem Sie auf das Pluszeichen (+) klicken. Anschließend können Sie alle Namen der Quellen- und Zieldateien anzeigen, die in dieser Übertragung enthalten sind. Ist die Übertragung allerdings gerade aktiv und werden viele Dateien übertragen, können Sie nur die Dateien anzeigen, die bereits übertragen wurden.

Um zu aktualisieren, was im **Übertragungsprotokoll** angezeigt wird, klicken Sie in der Symbolleiste der Inhaltsansicht auf die Schaltfläche **Refresh** (Aktualisieren) . Die Informationen zur Dateiübertragung im Übertragungsprotokoll sind auch nach dem Stoppen und einem anschließenden Neustart von IBM MQ Explorer noch vorhanden. Sollen alle abgeschlossenen Dateiübertragungen aus dem Protokoll gelöscht werden, klicken Sie in der Symbolleiste der Inhaltsansicht auf **Abgeschlossene Übertragungen entfernen** .

Klicken Sie zum Löschen einzelner abgeschlossener Dateiübertragungen aus dem Protokoll mit der rechten Maustaste auf die Übertragung, und klicken Sie auf **Löschen**. Wenn Sie eine Übertragung löschen, werden nur die gespeicherten historischen Daten gelöscht; aktive oder geplante Übertragungen werden nicht beendet bzw. abgebrochen.

Um die eindeutige Kennung einer Übertragung in die Zwischenablage zu kopieren, klicken Sie mit der rechten Maustaste auf die Übertragung und klicken Sie auf **ID kopieren**.

Die Metadaten und die vollständigen XML-Überwachungsdaten der Übertragung stehen im Popup-Menü unter der Aktion **Eigenschaften** zur Verfügung.

Zugehörige Tasks

„Überwachen von Dateiübertragungen, die sich in Bearbeitung befinden“ auf Seite 235

Sie können eine laufende Dateiübertragung über die Registerkarte **Verwaltete Dateiübertragung-Aktueller Übertragungsfortschritt** in IBM MQ Explorer überwachen. Diese Dateiübertragung kann aus IBM MQ Explorer oder von der Befehlszeile aus gestartet worden sein. Auf der Registerkarte wird auch der Fortschritt geplanter Übertragungen an der Stelle angezeigt, an der die geplanten Übertragungen beginnen.

„Übertragungsprotokoll konfigurieren“ auf Seite 239

Sie können angeben, welche Informationen im **Übertragungsprotokoll** in IBM MQ Explorer angezeigt werden sollen und wie die Anzeige erfolgen soll.

„Zeitlimit für die Wiederherstellung von blockierten Übertragungen festlegen“ auf Seite 345

Sie können ein Zeitlimit für die Übertragungswiederherstellung für blockierte Dateiübertragungen festlegen, die für alle Übertragungen eines Quellenagenten angewendet werden. Sie können auch ein Zeitlimit für die Übertragungswiederherstellung für eine einzelne Übertragung festlegen. Wenn Sie eine bestimmte Zeitdauer (in Sekunden) festlegen, während der ein Quellenagent versucht, eine blockierte Dateiübertragung wiederherzustellen, und die Übertragung noch nicht erfolgreich war, wenn der Agent dieses Zeitlimit erreicht, schlägt die Übertragung fehl.


Übertragungsprotokoll konfigurieren

Sie können angeben, welche Informationen im **Übertragungsprotokoll** in IBM MQ Explorer angezeigt werden sollen und wie die Anzeige erfolgen soll.


Informationen zu diesem Vorgang

Wenn Sie die Reihenfolge der Spalten im **Übertragungsprotokoll** ändern möchten, klicken Sie auf den Titel der Spalte, die Sie verschieben möchten, und ziehen Sie die Spalte in die neue Position. Die neue Spaltenanordnung wird beibehalten, bis IBM MQ Explorer gestoppt und erneut gestartet wird.

Um Einträge im **Übertragungsprotokoll** zu filtern, geben Sie eine Zeichenfolge in das Feld **Anzeige der angezeigten Protokolleinträge filtern** ein. Wenn Sie alle Einträge in das Protokoll zurückschreiben möchten, löschen Sie die Zeichenfolge, die Sie aus dem Feld eingegeben haben. In diesem Feld können Sie einen beliebigen regulären Java-Ausdruck angeben. Weitere Informationen finden Sie im Abschnitt [Von MFT verwendete reguläre Ausdrücke](#).

Wenn Sie anpassen möchten, welche Spalten im Übertragungsprotokoll angezeigt werden, verwenden Sie die Option **Spalten des Übertragungsprotokolls konfigurieren** . Führen Sie die folgenden Schritte aus, um das Fenster **Spalten des Übertragungsprotokolls konfigurieren** zu starten und zu verwenden.

Vorgehensweise

1. Stellen Sie sicher, dass das **Übertragungsprotokoll** in der Inhaltsansicht geöffnet ist. Klicken Sie in der Symbolleiste der Inhaltsansicht auf **Spalten des Übertragungsprotokolls konfigurieren** . Das Fenster **Spalten des Übertragungsprotokolls konfigurieren** wird geöffnet.
2. Wenn Sie Ihre Sicht des **Übertragungsprotokolls** anpassen möchten, wählen Sie einzelne Markierungsfelder für die Spalten aus, die angezeigt oder ausgeblendet werden sollen, oder wählen Sie sie ab. Sie können auf **Alle auswählen** und dann auf **OK** klicken, um alle Markierungsfelder auszuwählen, oder **Alles abwählen** und anschließend **OK**, um alle Markierungsfelder zu löschen.

Zugehörige Tasks

[„Überwachen von Dateiübertragungen, die sich in Bearbeitung befinden“](#) auf Seite 235

Sie können eine laufende Dateiübertragung über die Registerkarte **Verwaltete Dateiübertragung-Aktueller Übertragungsfortschritt** in IBM MQ Explorer überwachen. Diese Dateiübertragung kann aus IBM MQ Explorer oder von der Befehlszeile aus gestartet worden sein. Auf der Registerkarte wird auch der Fortschritt geplanter Übertragungen an der Stelle angezeigt, an der die geplanten Übertragungen beginnen.

[„Status von Dateiübertragungen im Übertragungsprotokoll anzeigen“](#) auf Seite 237

Sie können im **Übertragungsprotokoll** in IBM MQ Explorer die Details zu Dateiübertragungen ansehen. Dabei spielt es keine Rolle, ob die Übertragungen über die Befehlszeile oder über IBM MQ Explorer gestartet wurden. Sie können den Inhalt des **Übertragungsprotokolls** anpassen.

MFT-Ressourcen überwachen

Sie können Managed File Transfer-Ressourcen überwachen, beispielsweise eine Warteschlange oder ein Verzeichnis. Sobald für diese Ressource eine festgelegte Bedingung eintritt, startet die Ressourcenüberwachung eine bestimmte Task, beispielsweise die Übertragung einer Datei. Ressourcenüberwachungen können mit dem Befehl **fteCreateMonitor** oder über die Ansicht **Überwachungen** im Managed File Transfer-Plug-in für IBM MQ Explorer erstellt werden.

Informationen zu diesem Vorgang

In Zusammenhang mit der Managed File Transfer-Ressourcenüberwachung werden folgende Begriffe verwendet:

Ressourcenüberwachung

Bei der Ressourcenüberwachung handelt es sich um einen Prozess, der eine Ressource (z. B. ein Verzeichnis oder eine Warteschlange) in regelmäßigen Abständen abfragt, um festzustellen, ob sich der Inhalt der Ressource geändert hat. Ist dies der Fall, wird der Inhalt mit den für diesen Über-

wachungsprozess definierten Bedingungen verglichen. Werden diese Bedingungen erfüllt, wird die diesem Überwachungsprozess zugeordnete Task gestartet.

Resource

Die Systemressource, die von der Ressourcenüberwachung in jedem Abfrageintervall überprüft und mit den Auslöserbedingungen verglichen wird. Bei der überwachten Ressource kann es sich um Warteschlangen, Verzeichnisse oder verschachtelte Verzeichnisstrukturen handeln.

Bedingung und Auslöserbedingung

Bei einer Bedingung handelt es sich um einen Ausdruck, der ausgewertet wird (in der Regel durch einen Vergleich mit dem Inhalt der überwachten Ressource). Ergibt der Ausdruck 'true', trägt diese Bedingung zur (endgültigen) Auslöserbedingung bei.

Die Auslöserbedingung ist die endgültige Bedingung, die erfüllt ist, wenn alle anderen Bedingungen erfüllt sind. Wird die Auslöserbedingung erfüllt, kann die Task fortgesetzt werden.

Task

Eine Task ist der Vorgang, der bei Erfüllung der Auslöserbedingung oder mehrerer Bedingungen ausgeführt wird. Unterstützte Tasks sind Dateiübertragungen und Befehlsaufrufe.

Auslöserdatei

Bei einer Auslöserdatei handelt es sich um eine Datei, die in ein überwachtes Verzeichnis eingefügt wird und anzeigt, dass eine Task (in der Regel eine Übertragung) gestartet werden kann. Die Auslöserdatei kann beispielsweise anzeigen, dass alle Dateien, die verarbeitet werden sollen, in einem bekannten Verzeichnis vorhanden sind und übertragen oder anderweitig verarbeitet werden können. Außerdem können durch den Namen der Auslöserdatei mittels Variablensubstitution auch die zu übertragenden Dateien angegeben werden. Weitere Informationen finden Sie unter „Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“ auf Seite 252.

Die Auslöserdatei wird gelegentlich auch als Ready- oder Go-Datei bezeichnet. In dieser Dokumentation wird sie jedoch in der Regel als Auslöserdatei bezeichnet.

Die Ressourcenüberwachung wird auf Protokollbridgeagenten oder Connect:Direct -Bridgeagenten nicht unterstützt.

Zugehörige Konzepte

[Anleitung zur Konfiguration eines MFT-Ressourcenmonitors zur Vermeidung der Überlastung eines Agenten](#)

Zugehörige Verweise

[fteCreateMonitor: MFT-Ressourcenüberwachung erstellen](#)

[fteListMonitors: MFT-Ressourcenüberwachungen auflisten](#)

[fteDeleteMonitor: MFT-Ressourcenüberwachung löschen](#)

[Formate der MFT-Überwachungsanforderungsnachrichten](#)

Konzepte der MFT-Ressourcenüberwachung

Dieser Abschnitt enthält eine Übersicht über die wichtigsten Begriffe im Zusammenhang mit der Managed File Transfer-Ressourcenüberwachung.

Ressourcenüberwachung

Sie erstellen eine Ressourcenüberwachung, indem Sie den Befehl **fteCreateMonitor** verwenden, der eine neue Ressourcenüberwachung über die Befehlszeile erstellt und startet. Der Ressourcenmonitor ist einem Managed File Transfer-Agenten zugeordnet und nur aktiv, wenn dieser Agent ausgeführt wird. Wenn der Überwachungsagent stoppt, wird auch die Ressourcenüberwachung gestoppt. Wenn der Agent beim Erstellen des Ressourcenmonitors bereits ausgeführt wird, wird der Ressourcenmonitor sofort gestartet. Der Überwachungsagent muss auch der Quellenagent der Task sein, die durch die Ressourcenüberwachung eingeleitet wird.

Die Namen der Ressourcenüberwachung müssen innerhalb ihres Agenten eindeutig sein. Der Name der Ressourcenüberwachung muss mindestens ein Zeichen lang sein und darf keinen Stern (*), Prozentzeichen (%) oder Fragezeichen (?) enthalten. Es wird ignoriert, ob der Name einer Ressourcenüberwa-

chung in Groß- oder Kleinschreibung bereitgestellt wird und der Ressourcenüberwachungsname wird in Großschreibung umgewandelt. Wenn Sie versuchen, eine Ressourcenüberwachung mit einem Namen zu erstellen, der bereits vorhanden ist, wird die Anforderung ignoriert und der Versuch wird in das Überwachungsprotokollthema protokolliert.

Anmerkung: Sie können keine Ressourcenüberwachung mit einer Taskdefinition erstellen, die geplante Übertragungen enthält.

Vor IBM MQ 9.3.0 besteht die einzige Möglichkeit zum Stoppen einer Ressourcenüberwachung darin, den Agenten zu stoppen, der die Überwachungsoperation ausführt. Wenn Sie eine Ressourcenüberwachung erneut starten möchten, müssen Sie den Agenten vollständig erneut starten. Von IBM MQ 9.3.0 können Sie Ressourcenmonitore starten und stoppen, ohne einen Agenten anhalten oder neu starten zu müssen. Weitere Informationen finden Sie unter [„Ressourcenüberwachung starten und stoppen“](#) auf Seite 243.

Es gibt keine Einschränkung für die Anzahl der Ressourcenüberwachungen, die auf einem Agenten erstellt und die alle mit derselben Priorität ausgeführt werden können. Berücksichtigen Sie die Auswirkungen von überlappenden überwachten Ressourcen, Konflikte verursachenden Auslöserbedingungen und wie oft die Ressourcen abgefragt werden.

Überlappende Ressourcenmonitore können Folgendes verursachen:

- Mögliche Konflikte in Bezug auf den Quellort/die Elemente.
- Mögliche doppelte Übertragungsanforderungen für dieselben Quelleinheiten.
- Unerwartete Fehler oder Ausfälle bei Übertragungen aufgrund von Konflikten zwischen Quelleinheiten.

Wenn mehrere Monitore denselben Standort scannen und auf dieselben Elemente triggern können, kommt es möglicherweise zu dem Problem, dass zwei verschiedene Monitore verwaltete Übertragungsanforderungen für dieselbe Einheit senden.

Ressourcenüberwachungen prüfen den Inhalt der Ressourcen nach jedem Abfrageintervall. Der Inhalt der Ressource wird mit den Auslöserbedingungen verglichen, und wenn diese Bedingungen erfüllt sind, wird die Task aufgerufen, die der Ressourcenüberwachung zugeordnet ist.

Die Task wird asynchron gestartet. Wenn eine Bedingung übereinstimmt und die Task gestartet wird, fragt die Ressourcenüberwachung weiterhin nach weiteren Änderungen am Ressourceninhalt ab. Wenn z. B. eine Übereinstimmung aufgetreten ist, weil eine Datei mit dem Namen `reports.go` in einem überwachten Verzeichnis eingetroffen ist, wird die Task einmal gestartet. Im nächsten Abfrageintervall, selbst wenn die Datei noch vorhanden ist, wird die Task nicht erneut gestartet. Wenn die Datei jedoch gelöscht und dann erneut in das Verzeichnis gestellt wird oder die Datei aktualisiert wird (so dass das letzte geänderte Datumsattribut geändert wird), führt die nächste Auslöserbedingung dazu, dass die Task erneut aufgerufen wird.

Wenn eine Ressourcenüberwachung vor IBM MQ 9.1.5 eine Abfrage durchführt, die länger als das Abfrageintervall dauert, wird die nächste Abfrage gestartet, sobald die aktuelle Abfrage beendet ist. Es gibt dazwischen keine Lücke, was sich darauf auswirken kann, wie schnell Ressourcenüberwachungen die Arbeit an einen Agenten übergeben. Dies kann zu Leistungsproblemen führen, wenn die während der ersten Abfrage gefundenen Elemente noch immer vorhanden sind, wenn die zweite Abfrage ausgeführt wird.

Die Ressourcenüberwachung verwendet den Service `ScheduledExecutor` und leitet die nächste Abfrage nur nach Abschluss der vorherigen Abfrage plus der konfigurierten Abfrageintervallzeit ein. Es gibt also immer eine Lücke zwischen den Abfrageintervallen und eine weitere Abfrage startet nicht direkt nach der vorherigen Abfrage, wenn die Dauer der Abfrage das Abfrageintervall übersteigt.

Wenn die Übertragung einer Datei fehlgeschlagen ist, können Sie das Ressourcenüberwachungsprotokoll löschen. Dadurch kann eine weitere Übertragungsanforderung übergeben werden, ohne dass die Datei gelöscht und erneut in das Verzeichnis gestellt werden muss, oder die Datei aktualisieren, um das Attribut für das Datum der letzten Änderung zu ändern. Das Löschen des Protokolls ist beispielsweise dann hilfreich, wenn die Datei übertragen werden muss, das Ändern der Datei aber nicht möglich ist. Weitere Informationen finden Sie unter [„Protokoll der Ressourcenüberwachung löschen“](#) auf Seite 271.

Ressourcen

Ressourcenüberwachungen in Managed File Transfer können die Inhalte der folgenden beiden Ressourcentypen abfragen:

Verzeichnisse oder verschachtelte Verzeichnisstrukturen

Ein gängiges Szenario ist beispielsweise die Überwachung eines Verzeichnisses auf das Vorhandensein einer Auslöserdatei. Eine externe Anwendung beispielsweise verarbeitet mehrere Dateien und stellt sie in ein bekanntes Quellenverzeichnis. Nach Abschluss der Verarbeitung zeigt die Anwendung an, dass die Dateien nun übertragen oder auf andere Weise verarbeitet werden können, indem sie eine Auslöserdatei in das überwachte Verzeichnis einfügt. Die Auslöserdatei kann von einer Managed File Transfer-Ressourcenüberwachung erkannt werden und die Übertragung dieser Dateien aus dem Quellenverzeichnis an einen anderen Managed File Transfer Agent wird eingeleitet.

Standardmäßig wird das angegebene Verzeichnis überwacht. Wenn Sie auch Unterverzeichnisse prüfen möchten, legen Sie die Rekursionsebene im Befehl **fteCreateTransfer** fest.

Nachfolgend finden Sie zwei Beispiele für die Überwachung eines Verzeichnisses:

- Überwachen Sie eine Auslöserdatei (z. B. `trigger.file`), und übertragen Sie anschließend ein Platzhalterzeichen (z. B. `*.zip`).
- Überwachen Sie `*.zip` und übertragen Sie anschließend `${FilePath}` (z. B. die Datei, die die Übertragung ausgelöst hat). Weitere Informationen zur Variablensubstitution finden Sie im Abschnitt „Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“ auf Seite 252.

Anmerkung: Erstellen Sie keinen Monitor, der auf `*.zip` überwacht, und übertragen Sie dann `*.zip`. Der Monitor versucht, eine Übertragung von `*.zip` für jede ZIP-Datei auf Ihrem System zu starten. Das heißt, der Monitor generiert * die Anzahl der Übertragungen für `*.zip`.

Ein Beispiel für die Erstellung einer Ressourcenüberwachung zur Überwachung eines Verzeichnisses finden Sie im Abschnitt „Verzeichnis überwachen und Variablensubstitution verwenden“ auf Seite 249.

IBM MQ-Warteschlangen

Ein Beispiel für die Überwachung einer Warteschlange ist beispielsweise, wenn eine externe Anwendung Nachrichten generiert und diese in eine bekannte Warteschlange mit der gleichen Gruppen-ID stellt. Nachdem die Anwendung alle Nachrichten in die Warteschlange gestellt hat, meldet sie, dass die Gruppe vollständig ist. Eine Managed File Transfer-Ressourcenüberwachung kann erkennen, dass die Nachrichtengruppe vollständig ist, und die Übertragung der Nachrichtengruppe aus der Quellenwarteschlange in eine Datei wird eingeleitet. Ein Beispiel für die Erstellung einer Ressourcenüberwachung zur Überwachung einer Warteschlange finden Sie im Abschnitt „Beispiel: MFT-Ressource konfigurieren“ auf Seite 251.

Anmerkung: Sie können pro Warteschlange nur einen Monitor angeben. Falls Sie angeben, dass eine IBM MQ-Warteschlange von mehreren Überwachungsprozessen abgefragt werden soll, hat dies ein unvorhersehbares Verhalten zur Folge.

Überwachungsdatengruppen werden nicht unterstützt.

Bedingungen und Auslöserbedingungen

Die Bedingung wird erfüllt, wenn die Ressource einen Wert enthält, der mit einer anderen Zeichenfolge oder einem anderen Muster übereinstimmt. Bedingungen können einer der folgenden Bedingungen sein:

- Übereinstimmung mit Dateiname (Muster)
- Keine Übereinstimmung mit Dateiname (Muster)
- Dateigröße
- Übereinstimmung, wenn die Dateigröße für eine Reihe von Abfragen identisch bleibt

Der Dateinamensabgleich kann wie folgt ausgedrückt werden:

- Exakte Übereinstimmung der Zeichenfolge

- Einfache Platzhalterübereinstimmung, wie in [Platzhalterzeichen mit MFT verwenden](#) beschrieben
- Übereinstimmung mit regulären Ausdrücken

Es ist auch möglich, Dateinamen vom Dateinamensabgleich auszuschließen. Zu diesem Zweck werden Platzhalterzeichen oder reguläre Java-Ausdrücke zur Angabe der Dateinamen verwendet, die nicht abgeglichen werden sollen.

Wenn eine übereinstimmende Datei erkannt wird, wird die Zeitmarke des letzten geänderten Zeitstempels beibehalten. Wenn nachfolgende Umfragen feststellen, dass die Datei geändert wurde, wird die Auslöserbedingung erneut erfüllt, und die Task wird gestartet. Wenn die Bedingung feststellt, dass eine Datei nicht vorhanden ist, wenn keine Datei im überwachten Verzeichnis mit dem Dateinamensmuster übereinstimmt, wird die Task gestartet. Wenn eine Datei dann dem Verzeichnis hinzugefügt wird, das mit dem Dateinamensmuster übereinstimmt, wird die Task nur gestartet, wenn die Datei dann gelöscht wird.

Aufgaben

In Managed File Transfer können nur die folgenden beiden Tasktypen so konfiguriert werden, dass sie von Ressourcenüberwachungsprozessen gestartet werden können:

Dateiübertragungstasks

Dateiübertragungstasks werden in der gleichen Weise definiert wie jede andere Dateiübertragung. Eine hilfreiche Methode zum Generieren der Task-XML, die für eine Überwachung erforderlich ist, ist die Ausführung des Befehls `fteCreateTransfer` mit dem Parameter `-gt`. Dieser Befehl generiert eine Taskdefinition als XML-Dokument, einschließlich der Übertragungsspezifikation. Anschließend übergeben Sie den Namen des Task-XML-Dokuments als Wert für den Parameter `-mt` im Befehl `fteCreateMonitor`. Wenn der `fteCreateMonitor` ausgeführt wird, liest er das Task-XML-Dokument. Nach der Ausführung von `fteCreateMonitor` werden Änderungen, die an der Task-XML-Datei vorgenommen werden, nicht vom Monitor verwendet.

Wenn Sie eine Dateiübertragungstask verwenden, können Sie auswählen, wie viele Auslöserbedingungen als Stapel in einer Task zusammengefasst werden. Der Standardwert ist eine Auslöserbedingung, um eine Task zu starten. Sie können den Befehl `fteCreateMonitor` mit der Option `-bs` ausführen, um die Anzahl der Auslöserbedingungen auszuwählen, die zusammen zu einer Task zusammengefasst werden.

Befehlstasks

Befehlstasks können Ant-Scripts ausführen, ausführbare Programme aufrufen oder JCL-Jobs ausführen. Weitere Informationen finden Sie unter [„MFT-Überwachungstasks für den Start von Befehlen und Scripts konfigurieren“](#) auf Seite 245.

Auslöserdateien

Sie können den Inhalt einer Auslöserdatei in einem Ressourcenmonitor verwenden, um eine Gruppe von Dateien zu definieren, die in einer einzigen Übertragungsanforderung übertragen werden sollen. Jedes Mal, wenn eine übereinstimmende Auslöserdatei gefunden wird, wird ihr Inhalt für Quellendateipfade und optional für Zieldateipfade geparkt. Diese Dateipfade werden dann verwendet, um die Dateielemente in der von Ihnen angegebenen Task-Transfer-XML-Datei zu definieren, die als einzelne Übertragungsanforderung an den Agenten übergeben wird. Die Definition des Ressourcenmonitors bestimmt, ob der Auslöserinhalt aktiviert ist.

Das Format jeder Auslöserdatei ist ein einzelner Dateipfad für die Übertragung in jeder Textzeile. Das Standardformat für die Zeile ist entweder ein einzelner Quellendateipfad oder ein Pfad der Quellen- und Zieldatei, die durch ein Komma voneinander getrennt sind.

Weitere Informationen und Beispiele finden Sie im Abschnitt [„Auslöserdatei verwenden“](#) auf Seite 261.

Ressourcenüberwachung starten und stoppen

Vor IBM MQ 9.3.0 besteht die einzige Möglichkeit zum Stoppen einer Ressourcenüberwachung darin, den Agenten zu stoppen, der die Überwachungsoperation ausführt. Wenn Sie ein Ressourcenüberwachung

erneut starten möchten, müssen Sie den Agenten vollständig erneut starten. Weitere Informationen hierzu finden Sie unter [„MFT-Agenten starten“](#) auf Seite 220 und [„MFT-Agenten stoppen“](#) auf Seite 226.

Ab IBM MQ 9.3.0 können Sie Ressourcenüberwachungen starten und stoppen, ohne einen Agenten stoppen oder erneut starten zu müssen, indem Sie die Befehle **fteStartMonitor** und **fteStopMonitor** verwenden. Dies ist z. B. in den folgenden Situationen nützlich:

- Wenn ein Agent über mehrere Ressourcenüberwachungen verfügt und nur einige von ihnen Fehler festgestellt haben, aber die übrigen Ressourcenüberwachungen fehlerfrei funktionieren, möchten Sie vielleicht nur die fehlgeschlagenen Ressourcenüberwachungen erneut starten.
- Wenn Sie eine Ressourcenüberwachung stoppen möchten, um Wartungsarbeiten durchzuführen, oder wenn die Ressourcenüberwachung eine Zeit lang nicht erforderlich ist und Sie nicht möchten, dass sie unnötig ausgeführt wird und dabei wertvolle Systemressourcen verbraucht.

Weitere Informationen hierzu finden Sie unter [„MFT-Ressourcenüberwachung starten“](#) auf Seite 267 und [„MFT-Ressourcenüberwachung stoppen“](#) auf Seite 268.

<i>Tabelle 9. Verhalten einer Ressourcenüberwachung abhängig vom ausgeführten Befehl</i>	
Befehl	Verhalten der Ressourcenüberwachung
fteStartMonitor	Wenn der Agent aktiv ist, wird die Ressourcenüberwachung gestartet, wenn sie derzeit gestoppt ist.
fteStopMonitor	Wenn der Agent aktiv ist, wird die Ressourcenüberwachung gestoppt, wenn sie gerade gestartet wird.
fteStartAgent	Die Ressourcenüberwachung wird als Teil des Agentenstartsgestartet, unabhängig von früheren Aufrufen von fteStopMonitor .
fteStopAgent	Alle aktiven Ressourcenüberwachungen werden gestoppt.

Ressourcenüberwachung sichern und wiederherstellen

Sie können die bereits definierten Ressourcenüberwachungen sichern, damit Sie sie in Zukunft erneut verwenden können. Es gibt verschiedene Optionen, die Sie wie folgt verwenden können:

- Verwenden Sie den Befehl **fteCreateMonitor** mit dem Parameter **-ox**, um eine Ressourcenüberwachungskonfiguration in eine XML-Datei zu exportieren, und den Parameter **-ix**, um eine Ressourcenüberwachung wiederherzustellen, indem Sie die Ressourcenüberwachungskonfiguration aus einer XML-Datei importieren.
- Verwenden Sie den Befehl **fteListMonitors** mit dem **-ox**, um die Definition für einen einzelnen Ressourcenmonitor in eine XML-Datei zu exportieren.
- Verwenden Sie den Befehl **fteListMonitors** mit dem **-od**, um mehrere Ressourcenmonitordefinitionen in ein angegebenes Verzeichnis zu exportieren. Jede Ressourcenmonitordefinition wird in einer separaten XML-Datei gespeichert. Sie können auch die Option **-od** verwenden, um eine einzelne Ressourcenüberwachungsdefinition in ein angegebenes Verzeichnis zu exportieren.

Weitere Informationen finden Sie unter [„MFT-Ressourcenüberwachungen sichern und wiederherstellen“](#) auf Seite 269.

Protokollierung der Ressourcenüberwachung

Managed File Transfer umfasst die Ressourcenmonitorprotokollierung. Weitere Informationen finden Sie unter [„MFT-Ressourcenmonitore protokollieren“](#) auf Seite 264.

Zugehörige Konzepte

[„Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“](#) auf Seite 252

Wenn die Auslöserbedingungen für eine aktive Ressourcenüberwachung erfüllt sind, wird die definierte Task aufgerufen. Zusätzlich zum Aufrufen der Übertragungs- oder Befehlstask mit demselben Zielagenten oder demselben Zieldateinamen können Sie die Taskdefinition auch zur Ausführungszeit ändern. Dies tun Sie, indem Sie Variablennamen in die Taskdefinitions-XML einfügen. Wenn der Monitor feststellt, dass die Auslöserbedingungen erfüllt sind, und dass die Taskdefinition Variablennamen enthält, ersetzt er die Variablennamen mit den Variablenwerten und ruft dann die Task auf.

Zugehörige Tasks

„MFT-Überwachungstasks für den Start von Befehlen und Scripts konfigurieren“ auf Seite 245

Ressourcenmonitore sind nicht auf die Ausführung von Dateiübertragungen als ihre zugeordnete Task beschränkt. Sie können den Monitor auch so konfigurieren, dass er andere Befehle vom Überwachungsagenten, einschließlich ausführbarer Programme, Ant-Scripts oder JCL-Jobs, aufruft. Wenn Sie Befehle aufrufen möchten, bearbeiten Sie die Monitor-Taskdefinitions-XML so, dass sie ein oder mehrere Befehls-elemente mit entsprechenden Befehlsaufrufparametern, wie z. B. Argumente und Eigenschaften, enthält.

„Beispiel: MFT-Ressource konfigurieren“ auf Seite 251

Sie können eine IBM MQ -Warteschlange als Ressource angeben, die von einer Ressourcenüberwachung überwacht werden soll, indem Sie den Parameter **-mq** mit dem Befehl **fteCreateMonitor** verwenden.

„Warteschlange überwachen und Variablensubstitution verwenden“ auf Seite 258

Mit dem Befehl **fteCreateMonitor** können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

Zugehörige Verweise

fteCreateMonitor: [MFT-Ressourcenüberwachung erstellen](#)

fteListMonitors: [MFT-Ressourcenüberwachungen auflisten](#)

fteDeleteMonitor: [MFT-Ressourcenüberwachung löschen](#)

MFT-Überwachungstasks für den Start von Befehlen und Scripts konfigurieren

Ressourcenmonitore sind nicht auf die Ausführung von Dateiübertragungen als ihre zugeordnete Task beschränkt. Sie können den Monitor auch so konfigurieren, dass er andere Befehle vom Überwachungsagenten, einschließlich ausführbarer Programme, Ant-Scripts oder JCL-Jobs, aufruft. Wenn Sie Befehle aufrufen möchten, bearbeiten Sie die Monitor-Taskdefinitions-XML so, dass sie ein oder mehrere Befehls-elemente mit entsprechenden Befehlsaufrufparametern, wie z. B. Argumente und Eigenschaften, enthält.

Informationen zu diesem Vorgang

Der Dateipfad zum ausführbaren Programm, zum Ant-Script oder zum JCL-Job, den der Überwachungsagent aufrufen soll, muss in den Befehlspfad (commandPath) des Überwachungsagenten eingeschlossen werden. Informationen zur Eigenschaft des Befehlspfads finden Sie im Abschnitt [MFT-Eigenschaft commandPath](#).

Sie können das XML-Dokument der Taskdefinition auf eine der folgenden Arten erstellen:

- Erstellen Sie das XML-Dokument für die Taskdefinition manuell entsprechend dem `FileTransfer.xsd`-Schema.
- Verwenden Sie ein generiertes XML-Dokument als Basis für Ihre Taskdefinition.

Unabhängig davon, ob Sie eine Übertragungsaufgabe oder eine Befehlstask wünschen, muss die Taskdefinition mit einem `<request>`-Stammelement beginnen. Das untergeordnete Element von `<request>` muss entweder `<managedTransfer>` oder `<managedCall>` sein. In der Regel wählen Sie `<managedCall>` aus, wenn ein einzelner Befehl oder ein einzelnes Script ausgeführt werden soll, und `<managedTransfer>`, wenn die Task eine Dateiübertragung und optional bis zu vier Befehlsaufrufe enthalten soll.

Prozedur

- Informationen zum manuellen Erstellen des XML-Dokuments für die Taskdefinition gemäß dem FileTransfer.xsd -Schema finden Sie im Artikel „Taskdefinition manuell entsprechend dem Schema erstellen“ auf Seite 246.
- Wenn Sie eine Taskdefinition erstellen, indem Sie ein generiertes Dokument ändern, bearbeiten Sie das vom **fteCreateTransfer**-Parameter **-gt** generierte XML-Dokument. Weitere Informationen finden Sie unter „Erstellen eines Taskdefinitionsdokuments durch Ändern eines generierten Dokuments“ auf Seite 248.

Taskdefinition manuell entsprechend dem Schema erstellen

Sie können eine XML-Taskdefinitionsdatei entsprechend dem Schema FileTransfer.xsd manuell erstellen.

Informationen zu diesem Vorgang

Das Schema FileTransfer.xsd kann in der `MQ_INSTALLATION_PATH/mqft/samples/schemagefunden` werden. Weitere Informationen zu diesem Schema finden Sie im Abschnitt Nachrichtenformat für Dateiübertragungsanforderung.

Beispiel

Das folgende Beispiel zeigt ein als `cleanup.xml`, gespeichertes XML-Beispieltaskdefinitionsdokument, das das Element `<managedCall>` verwendet, um ein Script Ant mit dem Namen `RunCleanup.xml` aufzurufen. Das Script `RunCleanup.xml` Ant muss sich auf dem Befehlspfad des Überwachungsagenten befinden.

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>hostName</hostName>
      <userID>userID</userID>
      <mqmdUserID>mqmdUserID</mqmdUserID>
    </originator>
    <agent QMgr="QM1" agent="AGENT1"/>
    <reply QMGR="QM1">reply</reply>
    <transferSet priority="1">
      <metaDataSet>
        <metaData key="name1">value1</metaData>
      </metaDataSet>
      <call>
        <command name="RunCleanup.xml" type="antscript" retryCount="2"
          retryWait="30" successRC="0">
          <target>check_exists</target>
          <target>copy_to_archive</target>
          <target>rename_temps</target>
          <target>delete_files</target>
          <property name="trigger.filename" value="{FileName}"/>
          <property name="trigger.path" value="{FilePath}"/>
        </command>
      </call>
    </transferSet>
  </job>
  <name>JOBCLEAN1</name>
</managedCall>
</request>
```

Das Element `<agent>` gibt den Managed File Transfer Agent an, der mit dem benannten Ant-Script auf seinem `commandPath` konfiguriert ist.

Die Struktur `<call><command> . . .` definiert die ausführbare Datei oder das Script, die bzw. das ausgeführt wird. Der Befehl verwendet ein optionales Attribut `type`, das einen der folgenden Werte haben kann:

antscript

Führen Sie ein Ant-Script in einer separaten JVM aus.

executable

Rufen Sie ein ausführbares Programm auf.

jcl

Rufen Sie einen JCL-Job auf.

Wenn Sie das Attribut `type` weglassen, wird der Standardwert ausführbare Datei verwendet.

Das Attribut `name` gibt den Namen des Ant-Scripts, der ausführbaren Datei oder des JCL-Jobs, den Sie ausführen möchten, ohne Pfadinformationen an. Der Agent sucht nach dem Script oder Programm an den Positionen, die durch die Eigenschaft `CommandPath` in der `agent.properties`-Datei des Agenten angegeben sind.

Das Attribut `retrycount` gibt an, wie oft das Programm erneut aufgerufen werden soll, wenn das Programm keinen Erfolgsrückkehrcode zurückgibt. Der Wert, der diesem Attribut zugeordnet wird, darf nicht negativ sein. Wenn Sie das Attribut `retrycount` nicht angeben, wird der Standardwert null verwendet.

Das Attribut `retrywait` gibt die Wartezeit in Sekunden an, bevor der Programmaufruf wiederholt wird. Der Wert, der diesem Attribut zugeordnet wird, darf nicht negativ sein. Wenn Sie das Attribut `retrywait` nicht angeben, wird der Standardwert null verwendet.

Das Attribut `successrc` ist ein Ausdruck, mit dessen Hilfe ermittelt wird, wann der Programmaufruf erfolgreich ausgeführt wird. Der Prozessrückgabecode für den Befehl wird mithilfe dieses Ausdrucks ausgewertet. Der Wert kann aus einem oder mehreren Ausdrücken kombiniert werden, die mit einem vertikalen Balken (`|`) kombiniert werden, um booleschen OR oder ein Et-Zeichen (`&`) als Boolesches UND zu bezeichnen. Folgende Ausdruckstypen sind möglich:

- Eine Zahl, um einen Gleichheitstest zwischen dem Prozessrückgabecode und der Zahl anzugeben.
- Eine Zahl, die mit einem Größer-als-Zeichen (`>`) als Präfix versehen ist, um einen Größer-als-Test zwischen der Zahl und dem Prozessrückgabecode anzugeben.
- Eine Zahl, die mit einem Kleiner-als-Zeichen (`<`) als Präfix versehen ist, um einen Kleiner-als-Test zwischen der Zahl und dem Prozessrückgabecode anzugeben.
- Eine Zahl, die mit einem Ausrufezeichen (`!`) als Präfix versehen ist, um einen Nicht-gleich-Test zwischen der Zahl und dem Prozessrückgabecode anzugeben. Beispiel: `>2&<7&!5|0|14` wird so interpretiert, als wären die folgenden Rückgabecodes erfolgreich: 0, 3, 4, 6, 14. Alle anderen Rückgabecode werden als nicht erfolgreich interpretiert.

Wenn Sie das Attribut `successrc` nicht angeben, wird der Standardwert null verwendet. Dies bedeutet, dass die Ausführung des Befehls nur als erfolgreich gewertet wird, wenn der Code null zurückgegeben wird.

Für ein Ant-Script geben Sie normalerweise die Elemente `<target>` und `<property>` an. Die `<target>`-Elementwerte müssen mit den Zielnamen im Script Ant übereinstimmen.

Für ausführbare Programme können Sie `<argument>`-Elemente angeben. Verschachtelte Argumentelemente geben Argumente an, die an das Programm übergeben werden sollen, das als Teil des Programmaufrufs aufgerufen wird. Die Programmargumente werden aus den Werten erstellt, die durch die Argumentelemente in der Reihenfolge angegeben sind, in der die Argumentelemente gefunden werden. Sie können null oder mehr Argumentelemente als verschachtelte Elemente eines Programmaufrufs angeben.

Der Administrator definiert und startet die Überwachung wie üblich mithilfe des XML-Dokuments zur Taskdefinition, das das Element `<managedCall>` enthält. For example:

```
fteCreateMonitor -ma AGENT1 -mm QM1 -md /monitored -mn MONITOR01 -mt
/tasks/cleanuptask.xml -pi 30 -pu seconds -tr match,*.go
```

Der Pfad zum XML-Dokument mit der Übertragungsdefinition muss sich in dem lokalen Dateisystem befinden, in dem Sie den Befehl **fteCreateMonitor** ausführen (in diesem Beispiel `/tasks/cleanuptask.xml`). Das `cleanuptask.xml`-Dokument wird nur zum Erstellen des Ressourcenmonitors verwendet. Alle Tasks, auf die sich die `cleanuptask.xml`-Dokumentreferenzen (Ant-Scripts oder JCL-Jobs) befinden, müssen sich im Befehlspfad des Überwachungsagenten befinden. Wenn die Auslöserbedingung für den Monitor erfüllt ist, werden alle Variablen in der Taskdefinitions-XML durch tatsächliche

Werte aus dem Monitor ersetzt. So wird beispielsweise `${FilePath}` in der Anforderungsnachricht ersetzt, die an den Agenten mit `/monitored/cleanup.gogesendet` wird. Die Anforderungsnachricht wird in die Befehlswarteschlange des Agenten gestellt. Der Befehlsprozessor erkennt, dass die Anforderung für einen Programmaufruf ist, und startet das angegebene Programm. Wenn ein Befehl des Typs `antscript` aufgerufen wird, dann wird eine neue JVM gestartet und die Task `Ant` wird unter der neuen JVM ausgeführt. Weitere Informationen zur Verwendung der Variablensubstitution finden Sie im Abschnitt [Tasks mit Variablensubstitution anpassen](#).

Zugehörige Konzepte

„Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“ auf Seite 252

Wenn die Auslöserbedingungen für eine aktive Ressourcenüberwachung erfüllt sind, wird die definierte Task aufgerufen. Zusätzlich zum Aufrufen der Übertragungs- oder Befehlstask mit demselben Zielagenten oder demselben Zieldateinamen können Sie die Taskdefinition auch zur Ausführungszeit ändern. Dies tun Sie, indem Sie Variablennamen in die Taskdefinitions-XML einfügen. Wenn der Monitor feststellt, dass die Auslöserbedingungen erfüllt sind, und dass die Taskdefinition Variablennamen enthält, ersetzt er die Variablennamen mit den Variablenwerten und ruft dann die Task auf.

Zugehörige Verweise

[Nachrichtenformat für Dateiübertragungsanforderungen](#)

[MFT-Eigenschaft 'commandPath'](#)

Erstellen eines Taskdefinitionsdokuments durch Ändern eines generierten Dokuments

Sie können das Dokument mit der Definition der Überwachungstask erstellen, indem Sie das XML-Dokument ändern, das von der Option `-gt` von `fteCreateTransfer` generiert wird.

Informationen zu diesem Vorgang

Das generierte Dokument enthält ein Element `<request>` gefolgt von einem Element `<managedTransfer>`. Gehen Sie wie folgt vor, um diese Taskdefinition in eine gültige `<managedCall>`-Struktur zu konvertieren:

Vorgehensweise

1. Ersetzen Sie die Start- und Endtags von `<managedTransfer>` durch `<managedCall>`-Tags.
2. Entfernen Sie alle `<schedule>`-Elemente und untergeordneten Knoten.
3. Ersetzen Sie die `<sourceAgent>`-Start- und Endtags durch `<agent>`, damit sie den Konfigurationsdetails des Überwachungsagenten entsprechen.
4. Entfernen Sie die Elemente `<destinationAgent>` und `<trigger>`.
5. Entfernen Sie `<item>`-Elemente.
6. Entfernen Sie alle Elemente `preSourceCall`, `postSourceCall`, `preDestinationCall` oder `postDestinationCall`.
7. Fügen Sie eine neue `<call> . . . </call>`-Struktur im Element `<transferSet>` ein. Diese Struktur enthält die Befehlsdefinition wie im folgenden Beispiel gezeigt:

```
<call>
  <command name="RunCleanup.xml" type="antscript" retryCount="2"
  retryWait="30" successRC="0">
    <target>check_exists</target>
    <target>copy_to_archive</target>
    <target>rename_temps</target>
    <target>delete_files</target>
    <property name="trigger.filename" value="${FileName}"/>
    <property name="trigger.path" value="${FilePath}"/>
  </command>
</call>
```


Beispiel

Sie können auch das Element `<managedTransfer>` einschließlich aller Dateiübertragungsdetails beibehalten und bis zu vier Befehlsaufrufe einfügen. In diesem Fall fügen Sie eine Auswahl der folgenden Aufrufelemente zwischen den Elementen `<metaDataSet>` und `<item>` ein:

preSourceCall

Rufen Sie ein Programm auf dem Quellenagenten auf, bevor Sie die Übertragung starten.

postSourceCall

Rufen Sie ein Programm auf dem Quellenagenten auf, nachdem Sie die Übertragung abgeschlossen haben.

Aufruf 'preDestinationCall'

Rufen Sie ein Programm auf dem Zielagenten auf, bevor Sie die Übertragung starten.

postDestinationCall

Rufen Sie ein Programm auf dem Zielagenten auf, nachdem Sie die Übertragung abgeschlossen haben.

Jedes dieser Elemente hat die Elementstruktur `<command>`, wie im vorherigen Beispiel beschrieben. Das Schema `FileTransfer.xsd` definiert die Typen, die von den verschiedenen Aufrufelementen verwendet werden.

Das folgende Beispiel zeigt 'preSourceCall', 'postSourceCall', 'preDestinationCall' und 'postDestinationCall' in einem Taskdefinitionsdokument:

```
:
<transferSet priority="1">
  <metaDataSet>
    <metaData key="key1">value1</metaData>
  </metaDataSet>
  <preSourceCall>
    <command name="send.exe" retryCount="0" retryWait="0" successRC="0"
      type="executable">
      <argument>report1.pdf</argument>
      <argument>true</argument>
    </command>
  </preSourceCall>
  <postSourceCall>
    <command name="//DO_IT.JCL" retryCount="0" retryWait="0" successRC="0"
      type="jcl">
      <argument>argument</argument>
    </command>
  </postSourceCall>
  <preDestinationCall>
    <command name="ant_script.xml" retryCount="0" retryWait="0" successRC="0"
      type="antscript">
      <target>step1</target>
      <property name="name" value="value"/>
    </command>
  </preDestinationCall>
  <postDestinationCall>
    <command name="runit.cmd" retryCount="0" retryWait="0" successRC="0" />
  </postDestinationCall>
  <item checksumMethod="none" mode="binary">
:

```

Sie können verschiedene Arten von Befehlen in die Übertragung mischen. Argument-, Ziel- und Eigenschaftselemente sind optional.

Verzeichnis überwachen und Variablensubstitution verwenden

Sie können mit dem Befehl **fteCreateMonitor** ein Verzeichnis überwachen. Der Wert einer Substitutionsvariablen kann in der XML-Definition der Task ersetzt und dazu verwendet werden, das Verhalten von Übertragungen zu steuern.

Informationen zu diesem Vorgang

In diesem Beispiel ist `AGENT_HOP` der Quellenagent. Das Verzeichnis, das `AGENT_HOP` überwacht, wird als `/test/monitored` bezeichnet. Der Agent überprüft das Verzeichnis alle fünf Minuten.

Nachdem eine .zip -Datei in das Verzeichnis geschrieben wurde, schreibt die Anwendung, die die Datei in das Verzeichnis schreibt, eine Auslöserdatei in das gleiche Verzeichnis. Der Name der Auslöserdatei ist mit dem Namen der .zip -Datei identisch, weist jedoch eine andere Dateierweiterung auf. Nachdem die Datei file1.zip zum Beispiel in das Verzeichnis geschrieben wurde, wird die Datei file1.go in das Verzeichnis geschrieben. Die Ressourcenüberwachung überwacht das Verzeichnis für Dateien, die mit dem Muster *.go übereinstimmen, und verwendet dann die Variablensubstitution, um eine Übertragung der zugehörigen .zip -Datei anzufordern.

Vorgehensweise

1. Erstellen Sie die Task-XML, die die Task definiert, die die Überwachung bei ihrer Auslösung ausführen soll.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/${fileName}{token=1}{separator=.}.zip</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/out/${fileName}{token=1}{separator=.}.zip</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Die Variablen, die durch die der Auslöserdatei zugeordneten Werte ersetzt werden, sind **fett** hervorgehoben. Diese Task-XML wird in die Datei /home/USER1/task.xml gespeichert.

2. Erstellen Sie eine Ressourcenüberwachung, um das Verzeichnis /test/monitored zu überwachen. Führen Sie folgenden Befehl aus:

```
fteCreateMonitor -ma AGENT_HOP -mm QM_HOP -md /test/monitored
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr match,*.go -pi 5 -pu minutes
```

3. Ein Benutzer oder ein Programm schreibt die Datei jump.zip in das Verzeichnis /test/monitored und schreibt dann die Datei jump.go in das Verzeichnis.
4. Der Monitor wird durch die Existenz der Datei jump.go ausgelöst. Der Agent ersetzt die Informationen zur Auslöserdatei in den XML-Code der Task.

Die Task-XML wird entsprechend wie folgt umgesetzt:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/jump.zip</file>
```

```

</source>
<destination type="file" exist="overwrite">
  <file>/out/jump.zip</file>
</destination>
</item>
</transferSet>
</managedTransfer>
</request>

```

Ergebnisse

Die von der Task-XML definierte Übertragung wird ausgeführt. Die Datei `jump.zip` wird von `AGENT_HOP` aus dem `/test/monitored` -Verzeichnis gelesen und in eine Datei mit dem Namen `/out/jump.zip` übertragen, die sich auf dem System befindet, auf dem `AGENT_SKIP` ausgeführt wird.

Zugehörige Konzepte

„Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“ auf Seite 252

Wenn die Auslöserbedingungen für eine aktive Ressourcenüberwachung erfüllt sind, wird die definierte Task aufgerufen. Zusätzlich zum Aufrufen der Übertragungs- oder Befehlstask mit demselben Zielagenten oder demselben Zieldateinamen können Sie die Taskdefinition auch zur Ausführungszeit ändern. Dies tun Sie, indem Sie Variablennamen in die Taskdefinitions-XML einfügen. Wenn der Monitor feststellt, dass die Auslöserbedingungen erfüllt sind, und dass die Taskdefinition Variablennamen enthält, ersetzt er die Variablennamen mit den Variablenwerten und ruft dann die Task auf.

Zugehörige Tasks

„MFT-Überwachungstasks für den Start von Befehlen und Scripts konfigurieren“ auf Seite 245

Ressourcenmonitore sind nicht auf die Ausführung von Dateiübertragungen als ihre zugeordnete Task beschränkt. Sie können den Monitor auch so konfigurieren, dass er andere Befehle vom Überwachungsagenten, einschließlich ausführbarer Programme, Ant-Scripts oder JCL-Jobs, aufruft. Wenn Sie Befehle aufrufen möchten, bearbeiten Sie die Monitor-Taskdefinitions-XML so, dass sie ein oder mehrere Befehls-elemente mit entsprechenden Befehlsaufrufparametern, wie z. B. Argumente und Eigenschaften, enthält.

Zugehörige Verweise

fteCreateMonitor: MFT-Ressourcenüberwachung erstellen

Beispiel: MFT-Ressource konfigurieren

Sie können eine IBM MQ -Warteschlange als Ressource angeben, die von einer Ressourcenüberwachung überwacht werden soll, indem Sie den Parameter `-mq` mit dem Befehl **fteCreateMonitor** verwenden.

Informationen zu diesem Vorgang

In diesem Beispiel handelt es sich bei der zu überwachenden Ressource um die Warteschlange `MONITORED_QUEUE`. Diese Warteschlange muss sich im Warteschlangenmanager `QM_NEPTUNE` des Überwachungsagenten befinden. Die Bedingung für die Überwachung der Warteschlange ist das Vorhandensein einer vollständigen Nachrichtengruppe. Die Task, die ausgeführt werden soll, wenn die Bedingung erfüllt ist, ist in der Datei `task.xml` definiert.

Anmerkung: Pro Warteschlange sollte nicht mehr als eine Ressourcenüberwachung erstellt werden. Andernfalls hat dies ein unvorhersehbares Verhalten zur Folge.

Vorgehensweise

Geben Sie den folgenden Befehl ein:

```

fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE
-mt task.xml -tr completeGroups -pi 5 -pu minutes

```

Der Monitor überprüft die Warteschlange alle fünf Minuten, um festzustellen, ob die Bedingung `completeGroups` wahr ist. Wenn eine oder mehrere vollständige Gruppen in der Warteschlange vorhanden sind, führt der Monitor die Task aus, die in der `task.xml` -Datei einmal für jede vollständige Gruppe definiert ist.

Zugehörige Konzepte

„Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“ auf Seite 252

Wenn die Auslöserbedingungen für eine aktive Ressourcenüberwachung erfüllt sind, wird die definierte Task aufgerufen. Zusätzlich zum Aufrufen der Übertragungs- oder Befehlstask mit demselben Zielagenten oder demselben Zieldateinamen können Sie die Taskdefinition auch zur Ausführungszeit ändern. Dies tun Sie, indem Sie Variablennamen in die Taskdefinitions-XML einfügen. Wenn der Monitor feststellt, dass die Auslöserbedingungen erfüllt sind, und dass die Taskdefinition Variablennamen enthält, ersetzt er die Variablennamen mit den Variablenwerten und ruft dann die Task auf.

Zugehörige Tasks

„MFT-Überwachungstasks für den Start von Befehlen und Scripts konfigurieren“ auf Seite 245

Ressourcenmonitore sind nicht auf die Ausführung von Dateiübertragungen als ihre zugeordnete Task beschränkt. Sie können den Monitor auch so konfigurieren, dass er andere Befehle vom Überwachungsagenten, einschließlich ausführbarer Programme, Ant-Scripts oder JCL-Jobs, aufruft. Wenn Sie Befehle aufrufen möchten, bearbeiten Sie die Monitor-Taskdefinitions-XML so, dass sie ein oder mehrere Befehls-elemente mit entsprechenden Befehlsaufrufparametern, wie z. B. Argumente und Eigenschaften, enthält.

„Warteschlange überwachen und Variablensubstitution verwenden“ auf Seite 258

Mit dem Befehl **fteCreateMonitor** können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

Zugehörige Verweise

fteCreateMonitor: [MFT-Ressourcenüberwachung erstellen](#)

Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution

Wenn die Auslöserbedingungen für eine aktive Ressourcenüberwachung erfüllt sind, wird die definierte Task aufgerufen. Zusätzlich zum Aufrufen der Übertragungs- oder Befehlstask mit demselben Zielagenten oder demselben Zieldateinamen können Sie die Taskdefinition auch zur Ausführungszeit ändern. Dies tun Sie, indem Sie Variablennamen in die Taskdefinitions-XML einfügen. Wenn der Monitor feststellt, dass die Auslöserbedingungen erfüllt sind, und dass die Taskdefinition Variablennamen enthält, ersetzt er die Variablennamen mit den Variablenwerten und ruft dann die Task auf.



Achtung: Bei Variablennamen wird die Groß-/Kleinschreibung nicht beachtet.

Die für die Substitution verwendeten Variablen sind nur für positive Auslöserbedingungen verfügbar. Nur bei Erfüllung der Auslöserbedingungen `match` und `fileSize` werden Variablen ersetzt. Wenn eine `NoMatch`-Bedingung verwendet wird und in der Taskdefinition Substitutionsvariablennamen vorhanden sind, wird die Task nicht aufgerufen, und die Überwachung hebt den Rückkehrcode 110 und die Fehlermeldung `BFGDM0060Eauf`.

Wenn die überwachte Ressource eine Warteschlange ist

Die Werte der IBM MQ-Nachrichteneigenschaften der ersten Nachricht, die aus der überwachten Warteschlange gelesen wird, können in die XML-Datei mit der Taskdefinition eingesetzt werden.

Benutzerdefinierte Nachrichteneigenschaften sind mit dem Präfix `usr.` versehen, das jedoch nicht im Variablennamen enthalten sein darf. Variablennamen müssen ein Dollarzeichen (\$) vorangestellt und in geschweifte Klammern { } eingeschlossen werden.

Beispiel: `${destFileName}` wird durch den Wert der Nachrichteneigenschaft `usr.destFileName` der ersten Nachricht ersetzt, die aus der Quellenwarteschlange gelesen werden soll. Weitere Informationen finden Sie in den Abschnitten [Von MFT gelesene MQ-Nachrichteneigenschaften aus Nachrichten in Quellenwarteschlangen](#) und [„Warteschlange überwachen und Variablensubstitution verwenden“](#) auf Seite 258.

Wenn eine Variable nicht als Nachrichteneigenschaft definiert ist, meldet der Monitor einen BFGDM0060E-Fehler und gibt den Rückkehrcode 110 zurück (die Überwachung der Monitor-Taskvariablen ist fehlgeschlagen). Darüber hinaus schreibt der Agent die folgende Fehlernachricht in das zugehörige Ereignisprotokoll (outputN.log):

```
BFGDM0113W: Trigger failure for <monitor name> for reason BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

Wenn eine moderate oder ausführliche Ressourcenüberwachungsprotokollierung für den Monitor aktiviert ist, schreibt der Monitor die folgende Nachricht in das Ereignisprotokoll des Agenten (resmoneventN.log), um das Ressourcenmonitorereignisprotokoll zu überwachen:

```
BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

Weitere Informationen zur Protokollierung von Ressourcenmonitoren finden Sie unter „[MFT-Ressourcenmonitore protokollieren](#)“ auf Seite 264.

In der folgenden Tabelle ist dargestellt, welche Substitutionsvariablen standardmäßig bereitgestellt werden. `#{AGENTNAME}` wird z. B. durch den Namen des Ressourcenmonitoragenten ersetzt.

Variable	Beschreibung
AGENTNAME	Der Name des Ressourcenmonitoragenten.
QUEUENAME	Der Name der Warteschlange, die überwacht wird.
ENCODING	Die Zeichencodierung für die erste Nachricht in der Warteschlange oder die erste Nachricht in einer Gruppe.
MESSAGEID	Die IBM MQ-Nachrichten-ID der ersten Nachricht in der Warteschlange oder Gruppe.
GROUPID	Die IBM MQ-Gruppen-ID der Gruppe bzw. bei nur einer Nachricht die Nachrichten-ID. Diese Variable wird nur festgelegt, wenn Sie die Überwachung für vollständige Gruppen überwachen.
Aktuellzeitmarke	Eine Zeitmarke, die auf der lokalen Zeit basiert, zu der die Überwachung ausgelöst wurde. Der Zeitmarkenwert ist für den Agenten eindeutig.
currentTimeStampUTC	Eine Zeitmarke, die auf der Zeit in der UTC-Zeitzone basiert, die der Monitor ausgelöst hat. Der Zeitmarkenwert ist für den Agenten eindeutig.

Wenn die überwachte Ressource ein Verzeichnis ist

In der folgenden Tabelle ist die Gruppe der Variablennamen dargestellt, die in der Task-XML-Definition ersetzt werden können.

Variable	Beschreibung
FilePath	Der vollständige Pfadname der Auslöserdatei.
FileName	Der Dateinamensabschnitt des Auslösers.
LetzteModifiedTime	Die Uhrzeit, zu der die Auslöserdatei zuletzt geändert wurde. Diese Zeit wird als Ortszeit der Zeitzone ausgedrückt, in der der Agent ausgeführt wird, und wird als ISO 8601-Zeit formatiert.
LetzteÄnderungsdatum	Das Datum, an dem die Auslöserdatei zuletzt geändert wurde. Dieses Datum wird als das lokale Datum der Zeitzone ausgedrückt, in der der Agent ausgeführt wird, und wird als ISO 8601-Datum formatiert.

Tabelle 11. Variables, das ersetzt werden kann (Forts.)

Variable	Beschreibung
LastModified-TimeUTC	Die Uhrzeit, zu der die Auslöserdatei zuletzt geändert wurde. Diese Zeit wird als die lokale Zeit ausgedrückt, die in die UTC-Zeitzone konvertiert wurde und als ISO 8601-Zeit formatiert ist.
LastModified-DateUTC	Das Datum, an dem die Auslöserdatei zuletzt geändert wurde. Dieses Datum wird als das lokale Datum ausgedrückt, das in die UTC-Zeitzone konvertiert wurde und als ISO 8601-Datum formatiert ist.
AgentName	Der Name des Ressourcenmonitoragenten.
Aktuellezeitmarke	Eine Zeitmarke, die auf der lokalen Zeit basiert, zu der die Überwachung ausgelöst wurde. Der Zeitmarkenwert ist für den Agenten eindeutig.
CurrentTimeStampUTC	Eine Zeitmarke, die auf der Zeit in der UTC-Zeitzone basiert, die der Monitor ausgelöst hat. Der Zeitmarkenwert ist für den Agenten eindeutig.

Wenn es sich bei der überwachten Ressource um eine Auslöserdatei handelt

In der folgenden Tabelle wird die Gruppe der Variablennamen angezeigt, die ersetzt werden können, wenn eine Ressourcenüberwachung den Inhalt einer Auslöserdatei verwendet, um die Dateien zu bestimmen, die übertragen werden müssen.

Tabelle 12. Variablen, die bei Verwendung einer Auslöserdatei ersetzt werden können

Variable	Beschreibung
contentSource	Der vollständige Pfadname der Quelldatei.
contentDestination	Der vollständige Pfadname der Zieldatei.

Den Variablennamen muss ein Dollarzeichen (\$) vorangestellt werden, das in geschweifte Klammern ({}), eingeschlossen ist. Beispiel: `${FilePath}` wird durch den vollständig qualifizierten Dateipfad der entsprechenden Auslöserdatei ersetzt.

Es gibt zwei spezielle Schlüsselwörter, die auf Variablennamen angewendet werden können, um eine weitere Verbesserung zu ermöglichen. Diese sind:

token

Der Tokenindex, der ersetzt werden soll (beginnend bei 1 von links und beginnend bei -1 von rechts).

separator

Ein einzelnes Zeichen, um den Variablenwert zu tokenisieren. Der Standardwert ist der Schrägstrich (/) auf AIX and Linux -Plattformen oder der umgekehrte Schrägstrich (\) auf Windows -Plattformen. Das Trennzeichen kann jedoch ein beliebiges gültiges Zeichen sein, das im Variablenwert enthalten sein kann.

Wenn das Trennschlüsselwort in einem Variablennamen angegeben wird, wird der Variablenwert in Token entsprechend dem Trennzeichen aufgeteilt.

Der Wert, der dem Schlüsselwort token zugeordnet ist, wird als Index verwendet, um auszuwählen, welches Token verwendet werden soll, um den Variablennamen zu ersetzen. Der Tokenindex ist relativ zum ersten Zeichen in der Variablen und beginnt bei 1. Wenn das Schlüsselwort 'token' nicht angegeben ist, wird die gesamte Variable eingefügt.

Alle Werte, die in einem Agentennamen in der Nachrichten-XML ersetzt werden, werden in einer nicht-case-sensitiven Weise behandelt. Alle Managed File Transfer Agent-Namen werden in Großbuchstaben geschrieben. Wenn der Wert Paris in einem Agentenattribut in der Nachrichten-XML ersetzt wird, wird dieser Wert als Verweis auf den Agenten PARIS interpretiert.

Zugehörige Konzepte

„Beispiele: Variablenersetzung für Ressourcenmonitordefinitionen“ auf Seite 255

Beispiele für die Variablenersetzung für Ressourcenmonitordefinitionen unter Verwendung von XML und IBM MQ Explorer.

Zugehörige Tasks

Vorgehensweise, wenn durch die Variablensubstitution mehrere Dateien zu einem einzigen Dateinamen führen

Beispiele: Variablenersetzung für Ressourcenmonitordefinitionen

Beispiele für die Variablenersetzung für Ressourcenmonitordefinitionen unter Verwendung von XML und IBM MQ Explorer.

Beispiele für die Funktionsweise der Variablensubstitution

Wenn der Dateipfad zur entsprechenden Auslöserdatei `c:\MONITOR\REPORTS\Paris\Report2009.doc` auf Windows -und `/MONITOR/REPORTS/Paris/Report2009.doc` auf AIX and Linux -Plattformen ist, werden die Variablen wie in der folgenden Tabelle dargestellt ersetzt.

<i>Tabelle 13. Wie die Variablen ersetzt werden</i>	
Variablenspezifikation	Nach der Variablensubstitution
<code>\${FilePath}</code>	Windows : <code>c:\MONITOR\REPORTS\Paris\Report2009.doc</code> AIX and Linux : <code>/MONITOR/REPORTS/Paris/Report2009.doc</code>
<code>\${FilePath{token=1}{separator=.}}</code>	Windows : <code>c:\MONITOR\REPORTS\Paris\Report2009</code> AIX and Linux : <code>/MONITOR/REPORTS/Paris/Report2009</code>
<code>\${FilePath{token=2}{separator=.}}</code>	Windows: Dokument AIX and Linux : Dokument
<code>\${FilePath{token=3}}</code>	Windows : BERICHTE AIX and Linux : Paris

Sie können auch einen negativen Tokenindex angeben, um Token vom letzten Zeichen der Variablen aus gerechnet auszuwählen, wie in der folgenden Tabelle gezeigt. Die Beispiele in der Tabelle verwenden denselben Variablenwert, `c:\MONITOR\REPORTS\Paris\Report2009.doc` unter Windows und `/MONITOR/REPORTS/Paris/Report2009.doc` unter AIX and Linux.

<i>Tabelle 14. Beispiele für die Verwendung eines negativen Tokenindex</i>	
Variablenspezifikation	Nach der Variablensubstitution
<code>\${FilePath}</code>	Windows : <code>c:\MONITOR\REPORTS\Paris\Report2009.doc</code> AIX and Linux : <code>/MONITOR/REPORTS/Paris/Report2009.doc</code>

Tabelle 14. Beispiele für die Verwendung eines negativen Tokenindex (Forts.)

Variablenspezifikation	Nach der Variablensubstitution
<code>\${FilePath{token=-2}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux : /MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=-2}{separator=\\}}</code>	Windows : Paris AIX and Linux : Paris
<code>\${FilePath{token=-4}}</code>	Windows : MONITOR AIX and Linux : MONITOR

Die Variablen, die für die Substitution verwendet werden, sind nur für die folgenden positiven Auslöserbedingungen und die Option `noSizeChange` verfügbar. Dies ist eine Ausnahme von der Regel für positive Auslöserbedingungen:

- `match`
- `fileSize`
- `noSizeChange`

Wenn eine `NoMatch` -Bedingung verwendet wird und in der Taskdefinition Substitutionsvariablenamen vorhanden sind, wird die Task nicht aufgerufen, und die Überwachung hebt den Rückkehrcode 110 und die Fehlermeldung `BFGDM0060Eauf`.

Beispiel für die Verwendung von XML

In dem folgenden Beispiel einer Taskdefinitions-XML wird der Name des Überwachungsagenten als Quellenagent für die Übertragung (Paris) und der vorletzte Verzeichnisname im Dateipfad als Name des Zielagenten für die Übertragung (Report2009) verwendet; außerdem wird die übertragene Datei umbenannt in das Stammelement des Auslöserdateinamens mit der Erweiterung `.rpt`.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="${AgentName}" QMgr="QM1" />
    <destinationAgent agent="${FilePath{token=-2}}" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${FileName{token=1}{separator=.}}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Die Task-XML wird entsprechend wie folgt umgesetzt:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
```



```

</originator>
<sourceAgent agent="AGENT1" QMgr="QM1" />
<destinationAgent agent="Paris" QMgr="QMD" />
<transferSet>
  <item mode="binary" checksumMethod="MD5">
    <source recursive="false" disposition="leave">
      <file>c:/incoming/reports/summary/report.doc</file>
    </source>
    <destination type="file" exist="overwrite">
      <file>/reports/Report2009.rpt</file>
    </destination>
  </item>
</transferSet>
</managedTransfer>
</request>

```

Die Variable `#{FilePath}{token=-2}` im Attribut `agent` des Elements `<destinationAgent>` wird durch den Wert `Paris` ersetzt. Bei diesem Wert wird die Groß-/Kleinschreibung nicht beachtet; er wird als Verweis auf den Agenten `PARIS` interpretiert.

Beispiele für die Verwendung von IBM MQ Explorer

Wenn Sie eine Ressourcenüberwachung unter Verwendung von IBM MQ Explorer erstellen und die Überwachungseigenschaften und Auslöserbedingungen angegeben wurden, wird die Option zum Hinzufügen von Übertragungselementen zur Überwachung angegeben. Die folgenden Beispiele zeigen, wie die Variablen `#{FilePath}` und `#{FileName}` in der Anzeige **"Add a transfer item"** (Ein Übertragungselement hinzufügen) verwendet werden können, um Übertragungen anzupassen, die sich aus einem Abgleich der Ressourcenüberwachung ergeben.

Beispiel 1

Um die Quellendatei an eine andere Position zu übertragen, wenn eine Auslöserbedingung erfüllt ist, kann die Variable `#{FilePath}` verwendet werden:

- Setzen Sie den Quellendateinamen **Dateiname** auf `#{FilePath}`.
- Wählen Sie im Dropdown-Menü **Typ** für das Ziel **Verzeichnis** aus.
- Legen Sie als Ziel **Dateiname** die Position fest, an die die Quellendatei übertragen werden soll. Dies könnte beispielsweise `C:\MFT\out\sein`.

Beispiel 2

Um die Quellendatei an eine andere Position zu übertragen und die Erweiterung der Datei zu ändern, kann die Variable `#{FileName}` in Verbindung mit der Variablen `#{FilePath}` verwendet werden:

Im folgenden Beispiel wird angenommen, dass der Dateipfad der Quellendatei gleich `C:\MONITOR\REPORTS\Paris\Report2009.doc` ist:

- Setzen Sie den Quellendateinamen **Dateiname** auf `#{FilePath}`.
- Setzen Sie das Ziel **Dateiname** auf die Position, an die die Quellendatei übertragen werden soll, gefolgt von `#{FileName}{token=1}{separator=.}`, gefolgt von der neuen Erweiterung der Datei. Dies könnte zum Beispiel `C:\MFT\out\#{FileName}{token=1}{separator=.}.rpt` sein, was `C:\MFT\out\Report2009.rpt` mit dem Namen der Quellendatei gleichkäme würde.

Beispiel 3

Um einen Teil des Dateipfads der Quellendatei zu verwenden, um die Zieladresse der Übertragung zu bestimmen, kann die Variable `#{FilePath}` in Verbindung mit Token- und Trennzeichenspezifikationen verwendet werden.

Im folgenden Beispiel wird angenommen, dass der Dateipfad der Quellendatei gleich `C:\MONITOR\REPORTS\Paris\Report2009.doc` ist.

Es ist möglich, einen Teil des Quellendateipfads zu verwenden, um die Zieladresse der Datei zu bestimmen. Wenn Sie das Dateipfadbeispiel von `C:\MONITOR\REPORTS\Paris\Report2009.doc` verwenden,

wenn die Datei abhängig von der Position der Quelldatei in einen Ordner übertragen werden soll, d. h. Paris in diesem Beispiel, könnte Folgendes ausgeführt werden:

- Setzen Sie den Quelldateinamen **Dateiname** auf `${FilePath}`.
- Geben Sie für den Zieldateinamen **Dateiname** die Zieladresse an, in der sich die Ordner für die einzelnen Positionen befinden, und hängen Sie dann den Zielteil des Dateipfads und den Dateinamen an. Dies könnte zum Beispiel `C:\MFT\out\${FilePath}{token=-2}{separator=\}\${FileName}` sein, was `C:\MFT\out\Paris\Report2009.doc` mit dem Namen der Quelldatei gleichkäme würde.

Zugehörige Konzepte

„Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“ auf Seite 252

Wenn die Auslöserbedingungen für eine aktive Ressourcenüberwachung erfüllt sind, wird die definierte Task aufgerufen. Zusätzlich zum Aufrufen der Übertragungs- oder Befehlstask mit demselben Zielagenten oder demselben Zieldateinamen können Sie die Taskdefinition auch zur Ausführungszeit ändern. Dies tun Sie, indem Sie Variablennamen in die Taskdefinitions-XML einfügen. Wenn der Monitor feststellt, dass die Auslöserbedingungen erfüllt sind, und dass die Taskdefinition Variablennamen enthält, ersetzt er die Variablennamen mit den Variablenwerten und ruft dann die Task auf.

Zugehörige Tasks

Vorgehensweise, wenn durch die Variablensubstitution mehrere Dateien zu einem einzigen Dateinamen führen

Warteschlange überwachen und Variablensubstitution verwenden

Mit dem Befehl **fteCreateMonitor** können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

Informationen zu diesem Vorgang

In diesem Beispiel stellt der Quellenagent AGENT_VENUS eine Verbindung zum Warteschlangenmanager QM_VENUS her. AGENT_VENUS überwacht die Warteschlange START_QUEUE auf QM_VENUS. Der Agent fragt die Warteschlange alle 30 Minuten ab.

Sobald die Warteschlange eine vollständige Nachrichtengruppe enthält, sendet die Überwachungstask die Gruppe an eine Datei auf einem von mehreren Zielagenten, die alle mit dem Warteschlangenmanager QM_MARS verbunden sind. Der Name der Datei, in die die Nachrichtengruppe übertragen wird, wird von der IBM MQ-Nachrichteneigenschaft `usr.fileName` der ersten Nachricht der Gruppe bestimmt. Der Name des Agenten, an den die Nachrichtengruppe gesendet wird, wird von der IBM MQ-Nachrichteneigenschaft `usr.toAgent` der ersten Nachricht der Gruppe bestimmt. Wenn der Header `usr.toAgent` nicht gesetzt ist, wird als Standardwert für den Zielagenten AGENT_MAGENTA verwendet.

Wenn Sie `useGroups="true"` und nicht gleichzeitig `groupId="${GROUPID}"` angeben, wird bei der Übertragung nur die erste Nachricht in der Warteschlange berücksichtigt. Wenn Sie beispielsweise 'fileName' mithilfe der Variablensubstitution generieren, besteht die Möglichkeit, dass der Inhalt von `a.txt` nicht korrekt ist. Dies liegt daran, dass 'fileName' von der Überwachung generiert wird, bei der Übertragung aber eine Nachricht abgerufen wird, bei der es sich nicht um die handelt, von der die Datei 'fileName' generiert werden sollte.

Vorgehensweise

1. Erstellen Sie die Task-XML, die die Task definiert, die die Überwachung bei ihrer Auslösung ausführen soll.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
```

```

    <userID>USER1</userID>
  </originator>
  <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
  <destinationAgent agent="{toAgent}" QMgr="QM_MARS" />
  <transferSet>
    <item mode="binary" checksumMethod="none">
      <source>
        <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
      </source>
      <destination type="file" exist="overwrite">
        <file>/reports/{fileName}.rpt</file>
      </destination>
    </item>
  </transferSet>
</managedTransfer>
</request>

```

Die Variablen, die durch die Werte der IBM MQ-Nachrichtenheader ersetzt werden, sind **fett** hervorgehoben. Diese Task-XML wird in die Datei /home/USER1/task.xml gespeichert.

- Erstellen Sie eine Ressourcenüberwachung zur Überwachung der Warteschlange START_QUEUE. Führen Sie folgenden Befehl aus:

```

fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA

```

- Ein Benutzer oder ein Programm schreibt eine Gruppe von Nachrichten an die Warteschlange START_QUEUE.

Für die erste Nachricht dieser Gruppe sind die folgenden IBM MQ-Nachrichteneigenschaften gesetzt

```

usr.fileName=larmer
usr.toAgent=AGENT_VIOLET

```

- Die Überwachung wird ausgelöst, sobald die vollständige Gruppe geschrieben wird. Der Agent überträgt die IBM MQ-Nachrichteneigenschaften in die Task-XML.

Die Task-XML wird entsprechend wie folgt umgesetzt:

```

<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

Ergebnisse

Die von der Task-XML definierte Übertragung wird ausgeführt. Die vollständige Gruppe von Nachrichten, die von 'AGENT_VENUS' aus 'START_QUEUE' gelesen werden, wird in eine Datei mit dem Namen /reports/larmer.rpt auf dem System geschrieben, auf dem AGENT_VIOLET ausgeführt wird.

Nächste Schritte

Jede Nachricht in eine eigene Datei übertragen

Wenn eine Warteschlange überwacht werden und jede Nachricht in eine eigene Warteschlange übertragen werden soll, können Sie ähnlich wie zuvor in diesem Abschnitt beschrieben vorgehen.

1. Erstellen Sie den Monitor wie zuvor beschrieben, indem Sie den Parameter **-tr completeGroups** im Befehl **fteCreateMonitor** angeben.
2. Geben Sie in der Task-XML Folgendes an:

```
<queue useGroups="true" groupId="${GROUPID}">START_QUEUE</queue>
```

Wenn die Nachrichten in die Quellenwarteschlange gestellt werden, sollten sie nicht in eine IBM MQ-Gruppe eingefügt werden. Fügen Sie jeder Nachricht die IBM MQ-Nachrichteneigenschaften hinzu. Geben Sie beispielsweise für die Eigenschaft `usr.filename` für jede Nachricht einen eindeutigen Dateinamen an. Dadurch werden die einzelnen Nachrichten in der Quellenwarteschlange vom Managed File Transfer Agent wie jeweils eine eigene Gruppe verarbeitet.

Zugehörige Konzepte

[„Daten von Nachrichten in Dateien übertragen“](#) auf Seite 292

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nachrichten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

[„Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“](#) auf Seite 252

Wenn die Auslöserbedingungen für eine aktive Ressourcenüberwachung erfüllt sind, wird die definierte Task aufgerufen. Zusätzlich zum Aufrufen der Übertragungs- oder Befehlstask mit demselben Zielagenten oder demselben Zieldateinamen können Sie die Taskdefinition auch zur Ausführungszeit ändern. Dies tun Sie, indem Sie Variablennamen in die Taskdefinitions-XML einfügen. Wenn der Monitor feststellt, dass die Auslöserbedingungen erfüllt sind, und dass die Taskdefinition Variablennamen enthält, ersetzt er die Variablennamen mit den Variablenwerten und ruft dann die Task auf.

[Die Zieldateien, die bei einer Übertragung erstellt wurden, die von der Überwachung einer Warteschlangenressource gestartet wurde, enthalten die falschen Daten](#)

Zugehörige Tasks

[„MFT-Überwachungstasks für den Start von Befehlen und Scripts konfigurieren“](#) auf Seite 245

Ressourcenmonitore sind nicht auf die Ausführung von Dateiübertragungen als ihre zugeordnete Task beschränkt. Sie können den Monitor auch so konfigurieren, dass er andere Befehle vom Überwachungsagenten, einschließlich ausführbarer Programme, Ant-Scripts oder JCL-Jobs, aufruft. Wenn Sie Befehle aufrufen möchten, bearbeiten Sie die Monitor-Taskdefinitions-XML so, dass sie ein oder mehrere Befehls-elemente mit entsprechenden Befehlsaufrufparametern, wie z. B. Argumente und Eigenschaften, enthält.

[„Beispiel: MFT-Ressource konfigurieren“](#) auf Seite 251

Sie können eine IBM MQ -Warteschlange als Ressource angeben, die von einer Ressourcenüberwachung überwacht werden soll, indem Sie den Parameter **-mq** mit dem Befehl **fteCreateMonitor** verwenden.

Zugehörige Verweise

[fteCreateMonitor: MFT-Ressourcenüberwachung erstellen](#)

[Von MFT gelesene MQ-Nachrichteneigenschaften aus Nachrichten in Quellenwarteschlangen](#)

Wiederholungsverhalten des Überwachungsprogramms für Übertragungen aus Nachrichten an Dateien konfigurieren

Wenn eine von einer Ressourcenüberwachung ausgelöste Übertragung von Nachrichten an Dateien fehlschlägt und die Nachrichtengruppe, die das Überwachungsprogramm ausgelöst hat, in der Warteschlange bleibt, wird die betreffende Übertragung in nachfolgenden Abfrageintervallen erneut übergeben. Wie oft die Übertragung maximal wiederholt wird, richtet sich nach der Eigenschaft **monitorGroupRetryLimit** des Überwachungsagenten.

Informationen zu diesem Vorgang

Jedes Mal, wenn eine neue Nachricht-Datei-Übertragung ausgelöst wird, wird für die Übertragungstask eine neue Übertragungs-ID generiert.

Bei einem Neustart des Agenten löst das Überwachungsprogramm selbst dann eine erneute Übertragung aus, wenn die Anzahl der ausgelösten Übertragungen den Wert von **monitorGroupRetryLimit** in der Datei `agent.properties` überschritten hat. Der Wert der Eigenschaft **monitorGroupRetryLimit** gibt an, wie oft ein Überwachungsprogramm maximal eine erneute Nachricht-Datei-Übertragung auslöst, wenn die Nachrichtengruppe weiterhin in der Warteschlange verbleibt. Der Standardwert dieser Eigenschaft ist 10. Für diese Eigenschaft kann eine beliebige positive ganze Zahl oder -1 angegeben werden. Bei Angabe von -1 für diese Eigenschaft löst das Überwachungsprogramm die Übertragung unbegrenzt oft erneut aus, solange die Auslöserbedingung nicht erfüllt ist.

Kommt es durch einen Übertragungsversuch zu einer Überschreitung des Wertes der Eigenschaft **monitorGroupRetryLimit**, schreibt der Agent einen Fehler in sein Ereignisprotokoll.

Eine einzelne Nachricht wird wie eine einzelne Gruppe behandelt. Solange die Nachricht in der Warteschlange verbleibt und die Anzahl der ausgelösten Übertragungen unterhalb des Grenzwerts der Eigenschaft **monitorGroupRetryLimit** liegt, wird die Übertragung bei jedem Abfrageintervall erneut ausgelöst.

Gehen Sie zur Definition der Eigenschaft **monitorGroupRetryLimit** auf dem Überwachungsagenten wie folgt vor:

Vorgehensweise

1. Stoppen Sie den Überwachungsagenten mit dem Befehl **fteStopAgent**.
2. Bearbeiten Sie die `agent.properties`-Datei für den Überwachungsagenten, um die folgende Zeile einzuschließen:

```
monitorGroupRetryLimit=number_of_retries
```

Die Datei `agent.properties` befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_name`.

3. Starten Sie den Überwachungsagenten mit dem Befehl **fteStartAgent**.

Zugehörige Tasks

„Beispiel: MFT-Ressource konfigurieren“ auf Seite 251

Sie können eine IBM MQ -Warteschlange als Ressource angeben, die von einer Ressourcenüberwachung überwacht werden soll, indem Sie den Parameter **-mq** mit dem Befehl **fteCreateMonitor** verwenden.

Auslöserdatei verwenden

Sie können den Inhalt einer Auslöserdatei in einem Ressourcenmonitor verwenden, um eine Gruppe von Dateien zu definieren, die in einer einzigen Übertragungsanforderung übertragen werden sollen. Jedes Mal, wenn eine übereinstimmende Auslöserdatei gefunden wird, wird ihr Inhalt für Quellendateipfade und optional für Zieldateipfade geparkt. Diese Dateipfade werden dann verwendet, um die Dateielemente in der von Ihnen angegebenen Task-Transfer-XML-Datei zu definieren, die als einzelne Übertragungsanforderung an den Agenten übergeben wird. Die Definition des Ressourcenmonitors bestimmt, ob der Auslöserinhalt aktiviert ist.

Sie können die Dateiinhaltsauslösung aktivieren, wenn Sie einen Monitor erstellen, indem Sie den Parameter **-tc** (Auslöserinhalt) angeben. Dieser Parameter **-tc** gilt nur für die Dateiauslöseroptionen `match` und `noSizeChange`. Weitere Informationen zur Erstellung eines Überwachungsprogramms finden Sie unter **fteCreateMonitor: MFT-Ressourcenüberwachung erstellen**.

Bei Verwendung einer Auslöserinhaltsdatei ist das Standardformat jeder Zeile entweder:

- Ein einzelner Quellendateipfad oder
- Ein Quellendateipfad und ein Zieldateipfad, getrennt durch ein Komma

Dabei werden weiße Leerzeichen als Teil der Dateipfade verarbeitet. Es ist möglich, das Standardzeilenformat zu ändern, indem Sie die Parameter **-tcr** und **-tcc** im Befehl **fteCreateMonitor** angeben. Weitere Informationen finden Sie unter „Erweiterte Optionen“ auf Seite 263.

Nachdem eine Auslöserdatei syntaktisch analysiert wurde, wird eine Liste der Dateipfade generiert und auf die von Ihnen angegebene Übertragungstask-XML angewendet. Wie bei allen Monitoren ist das Format der Übertragungstask-XML eine vollständige Übertragungstask-XML, die vom **fteCreateTransfer**-Befehl mit einem einzelnen definierten Element oder einer Datei generiert wird. Das einzelne Element muss die Substitutionsvariablen `{contentSource}` und optional `{contentDestination}` als Ersetzungen für die Quellen- und Zieldateipfade verwenden. Der Monitor erweitert die Übertragungstask-XML, um ein Dateielement für jede Zeile (Dateipfad) in der Auslöserdatei einzuschließen.

Sie können die Dateiinhaltsauslösung nicht mit dem Parameter **-bs** verwenden, da der Parameter **-tc** eine Übertragungsanforderung für jede Auslöserdatei impliziert.

Beispiel

Im folgenden Beispiel wird ein Monitor definiert, der für eine Datei ausgelöst wird, die in `trig` endet, und die Dateipfade in dieser Datei liest.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

Der Befehl **fteCreateTransfer** erstellt eine Datei mit dem Namen `task.xml` für eine einzelne Datei mit dem Quellendateipfad `{contentSource}`. For example:

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
  </source>
</item>
```

Der Befehl **fteCreateMonitor** sucht nach Dateien, die auf `trig` im Verzeichnis `/home/trigdir` enden, und verwendet den Inhalt, um eine einzelne Übertragungsanforderung zu erstellen, die auf `task.xml` für alle Pfade in dieser Auslöserdatei basiert. Das Format der Auslöserdatei muss ein Dateipfad (nur Quelle) in jeder Zeile ohne Komma-Trennzeichen sein. For example:

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
:
```

Alle Dateien werden in das Verzeichnis `/file/destdir` mit dem Dateinamen und nicht in den Dateipfad geliefert. Das heißt, `/home/file/first.txt` wird an `/file/destdir/first.txt` übergeben.

Ändern Sie hingegen den Parameter **-dd /file/destdir** im Befehl **fteCreateTransfer** in `-df ${contentDestination}` und das Format des Inhalts einer Auslöserdatei in `'Quellendateipfad,Zieldateipfad'`, so können Sie für den gleichen Zielagenten unterschiedliche Zielpfade definieren. For example:

```
/home/file/first.txt,/home/other/sixth.txt
```

Die Zielposition wird dann `/home/other/sixth.txt`.

Die Substitutionsvariablen können mit einem Token versehen werden. Beispielsweise können Sie den Dateinamen mit `{contentDestination{token=-1}}` von dem angegebenen Pfad trennen. Wenn das **fteCreateTransfer**-Ziel als `-df /file/destdir/{contentDestination{token=-1}}` definiert ist, ist das neue Ziel für `/home/file/first.txt` daher `/file/destdir/sixth.txt`.

Erweiterte Optionen

Sie können das Standardzeilenformat für den Inhalt der Auslöserdatei ändern, indem Sie den Parameter **-tcx** *regex* verwenden. Geben Sie einen regulären Ausdruck an, der dem erforderlichen Zeilenformat entspricht, und stellt entweder eine oder zwei Erfassungsgruppen bereit. Die erste Erfassungsgruppe ist die Quelle, und die zweite, optionale Erfassungsgruppe ist die Zieladresse. For example:

- Der Quellen- und der Zielpfad werden durch einen Silbentrennungsstrich getrennt:

```
((?:[^-])+) - ((?:[^-])+) 
```

In diesem Beispiel wird das Trennzeichen in drei Positionen definiert, und alle drei Exemplare des Bindestrichs, - können in ein beliebiges Zeichen geändert werden. Stellen Sie sicher, dass Sie alle Sonderzeichen verlassen.

- Die Quellen- und Zielpfade werden durch ein Komma mit nachgestellten Leerzeichen getrennt. Die durch ein Nummernzeichen (#) gekennzeichneten Kommentare werden ignoriert.

```
((?:[^\, ]+), ((?:[^\, ]+)) *(?:#.*)+ 
```

Dateipfade dürfen kein Nummernzeichen (#) enthalten. Normalerweise lautet ein Eintrag wie folgt: `/home/source/from.txt,/home/destination/to.txt # some comment`.

Wenn Sie den Parameter **-tcx** verwenden, stellen Sie sicher, dass der reguläre Ausdruck korrekt entworfen und getestet ist, damit der Ausdruck Fehler erkennen und die Auslöserdateien korrekt syntaktisch analysieren kann.

Sie können die Reihenfolge der Erfassung mit dem Parameter **-tcc destSrc** umkehren. Wenn Sie diesen Parameter angeben, ist die erste Erfassungsgruppe der Zieldateipfad, und die zweite Gruppe ist der Quelldateipfad.

Behandlung von Fehlern

Leere Auslöserdatei

Wenn die Auslöserdatei leer ist, ist das Ergebnis keine Dateiübertragung. Dies bedeutet, dass der Monitor eine Übertragungsanforderung erstellt, aber keine Dateielemente angegeben sind.

Auslöserdatei mit Fehlern

Wenn ein Eintrag in einer Auslöserdatei nicht gegen das erwartete Format syntaktisch analysiert werden kann, wird keine Übertragungsanforderung generiert. Ein Monitorfehlerprotokoll wird veröffentlicht, und der Fehler wird auch im Ereignisprotokoll aufgezeichnet. Die Auslöserdatei wird als verarbeitet markiert, und die Überwachung versucht erst dann, die Datei zu verarbeiten, wenn die Datei aktualisiert wurde.

XML für nicht übereinstimmende Übertragungstasks

Die Übertragungstask-XML muss mit der Auslöserdatei übereinstimmen, d. h. wenn die Übertragungstask-XML sowohl `${contentSource}` als auch `${contentDestination}` aufweist, müssen alle Auslöserdateien für diesen Monitor über Quellen- und Zieldateipfade und ähnlich wie für die Umkehrdatei verfügen. Im ersten Fall meldet der Monitor einen Substitutionsfehler von `${contentDestination}`, wenn die Auslöserdatei nur den Quelldateipfad bereitstellt.

Beispiele

Das folgende Beispiel zeigt einen grundlegenden Inhaltsauslöser, bei dem der Inhalt einer Auslöserdatei nur einen Quelldateipfad hat:

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

Der Parameter **-tcr** definiert zwei Erfassungsgruppen einer Folge von Zeichen, die durch ein Leerzeichen voneinander getrennt sind. Der Parameter **-tcc destSrc** und die Option geben an, dass die Erfassungsgruppen als Ziel und dann als Quelle verarbeitet werden sollen.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -df ${contentDestination} ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
-tcr "((?:[^\ ])+) ((?:[^\ ])+)" -tcc destSrc
```

Zugehörige Konzepte

„Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“ auf Seite 252

Wenn die Auslöserbedingungen für eine aktive Ressourcenüberwachung erfüllt sind, wird die definierte Task aufgerufen. Zusätzlich zum Aufrufen der Übertragungs- oder Befehlstask mit demselben Zielagenten oder demselben Zieldateinamen können Sie die Taskdefinition auch zur Ausführungszeit ändern. Dies tun Sie, indem Sie Variablennamen in die Taskdefinitions-XML einfügen. Wenn der Monitor feststellt, dass die Auslöserbedingungen erfüllt sind, und dass die Taskdefinition Variablennamen enthält, ersetzt er die Variablennamen mit den Variablenwerten und ruft dann die Task auf.

Zugehörige Tasks

„Warteschlange überwachen und Variablensubstitution verwenden“ auf Seite 258

Mit dem Befehl **fteCreateMonitor** können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

Zugehörige Verweise

fteCreateMonitor: [MFT-Ressourcenüberwachung erstellen](#)

fteCreateTransfer: [Neue Dateiübertragung starten](#)

MFT-Ressourcenmonitore protokollieren

Sie können Diagnoseinformationen zu Ressourcenmonitoren abrufen, indem Sie die Protokollierung verwenden.

Informationen zu diesem Vorgang

Sie können die Protokollierung für Ressourcenmonitore verwenden, indem Sie entweder den Befehl **fteSetAgentLogLevel** oder die Datei `agent.properties` verwenden, um die Ressourcenmonitorprotokollierung zu steuern.

Beachten Sie, dass die vorhandenen Tracepunkte weiterhin für die Erfassung von Informationen verwendet werden.

Die Ressourcenüberwachungsprotokolle werden in eine Datei mit dem Namen `resmoneventN.log` geschrieben, wobei *N* für eine Zahl steht, z. B. `resmonevent0.log`. Die Ereignisprotokolldateien zeichnen mehrere Aktionen auf, die stattfinden, wenn ein Monitor eine Ressource abfragt, z. B. ein Verzeichnis oder eine Warteschlange.



Achtung: Alle Ressourcenmonitore eines Agenten schreiben in dieselbe Protokolldatei.

Eine Beispielausgabe einer Datei `resmoneventN.log` finden Sie unter [Vorgehensweise, wenn der MFT-Verzeichnisressourcenmonitor keine Dateien auslöst](#).

In der folgenden Tabelle ist der Typ der Ereignisse aufgeführt, die vom Ressourcenmonitor in die Protokolldatei geschrieben werden. In der dritten Spalte wird die Protokollebene beschrieben, die zum Erfassen jedes Ereignisses erforderlich ist, bei dem die niedrigste Ebene INFO und die höchste Stufe VERBOSE ist.

Beachten Sie, dass die Einstellung einer höheren Protokollebene auch Ereignisse auf niedrigerer Ebene schreibt. Wenn Sie beispielsweise die Protokollebene auf MODERATE setzen, werden auch INFO-Level-Ereignisse, aber keine Ereignisse vom Wert VERBOSE geschrieben.

Zahl	Ereignis	Protokollebene	Beschreibung
1	Überwachungsprogramm erstellt	INFO	Es wurde ein Ressourcenmonitor erstellt.
2	Überwachungsprogramm gelöscht	INFO	Ein Ressourcenmonitor wurde gelöscht.
3	Überwachungsprogramm gestoppt	INFO	Ein Ressourcenmonitor wurde gestoppt.
4	Überwachungsprogramm gestartet	INFO	Ein Ressourcenmonitor wurde gestartet.
5	Abstimmung starten	INFO	Ein Ressourcenmonitor hat einen neuen Abfragezyklus gestartet.
6	Abstimmung beenden	INFO	Ein Abfragezyklus für Ressourcenmonitorabfragen wurde beendet.
7	Musterübereinstimmung	PRÜFEN	Es wurde eine Datei im Auslösermonitorverzeichnis oder eine Nachricht in einer Warteschlange gefunden, die mit dem angegebenen Muster übereinstimmt.
8	Musterabweichung	PRÜFEN	Es wurde eine nicht übereinstimmende Datei im Auslösermonitorverzeichnis oder eine Nachricht in einer Warteschlange gefunden, die nicht mit dem angegebenen Muster übereinstimmt.
9	Übertragungsanforderung	INFO	Eine Übertragung wurde vom Ressourcenmonitor eingeleitet.
10	Verzeichnis zu tief	PRÜFEN	Das vom Ressourcenmonitor überwachte Verzeichnis enthält mehr Unterverzeichnisse, die abgefragt werden sollen, als die in der Ressourcenmonitorkonfiguration angegebene Anzahl.
11	Datei gesperrt	MODERIEREN	Die vom Ressourcenmonitor überwachte Auslöserdatei ist durch einen anderen Prozess gesperrt.
12	Dateigröße klein	MODERIEREN	Die Auslöserdatei ist kleiner als die in der Ressourcenmonitorkonfiguration angegebene Größe.
13	Dateigröße instabil	MODERIEREN	Die Auslöserdatei wird häufiger geändert, als von der Ressourcenmonitorkonfiguration erwartet wird.
14	Zu viele Pollings	MODERIEREN	Ein Ressourcenmonitor hat eine instabile Auslöserdatei zu viele Male abgefragt.
15	Übereinstimmung mit Elementen	INFO	Die Gesamtzahl der Auslöserdateien, die in dem Verzeichnis gefunden wurden, das von einem Ressourcenmonitor abgefragt wird.

Zahl	Ereignis	Protokollebene	Beschreibung
16	Übertragungselemente	INFO	Die Gesamtzahl der Elemente in der Übertragungsanforderung.
17	FDC generiert	MODERIEREN	Ein Ressourcenmonitor hat eine Ausnahmebedingung generiert.
18	Übertragungsanforderung	INFO	Übertragungsanforderung von Ressourcenmonitor übergeben.
19.	Monitor-Start fehlgeschlagen	MODERIEREN	Ein Ressourcenmonitor konnte nicht gestartet werden.
20	Protokoll gelöscht	INFO	Die Monitorprotokolldaten wurden gelöscht.
21	Löschen des Monitorprotokolls fehlgeschlagen	INFO	Der Versuch, Monitorprotokolldaten zu löschen, ist fehlgeschlagen.
22	Übertragungs-ID	INFO	ID der Übertragungsanforderung wurde von Monitor übergeben.
23	Stapelverarbeitung	INFO	Gesamtzahl der Übertragungsanforderungen für übereinstimmende Elemente: N , wobei N eine Zahl ist.
 24	Verbunden	PRÜFEN	Eine Ressourcenüberwachung hat eine Verbindung zum Agentenwarteschlangenmanager hergestellt.
 25	Unterbrochen	PRÜFEN	Eine Ressourcenüberwachung hat die Verbindung zum Agentenwarteschlangenmanager getrennt.
 26	Fehler beim Trennen der Verbindung	PRÜFEN	Eine Ressourcenüberwachung hat beim Trennen der Verbindung zum Agentenwarteschlangenmanager ein Problem festgestellt.

Prozedur

- Wenn Sie den Befehl **fteSetAgentLogLevel** zum Aktivieren und Inaktivieren der Ressourcenüberwachungsprotokollierung verwenden möchten, finden Sie im Abschnitt [fteSetAgentLog-Stufe](#) eine Beschreibung des Parameters **logMonitor** sowie Beispiele für die Verwendung der verschiedenen Optionen.
- Informationen zur Verwendung der `agent.properties`-Datei zur Steuerung der Ressourcenüberwachungsprotokollierung finden Sie in [Die MFT agent.properties-Datei](#) für eine Beschreibung der zusätzlichen Eigenschaften, die es Ihnen ermöglichen, die folgenden Protokollierungsaktivitäten auszuführen:
 - Protokollierung aktivieren oder inaktivieren
 - Die Größe der einzelnen Protokolldateien begrenzen
 - Begrenzen Sie die Anzahl der Protokolle, die von Ressourcenmonitoren generiert werden können

Beispiel

Das folgende Beispiel zeigt die Protokollierung auf verbose-Ebene für den Agenten HA2im Warteschlangenmanager MFTDEMO:

```
<?xml version="1.0"?>
<log:log version="6.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:log="https://www.ibm.com/log">
```

```
<log:originator>
  <log:request>
    <log:hostName>192.168.7.1</log:hostName>
    <log:userID>johndoe</log:userID>
  </log:request>
</log:originator>
<log:endpoint agent="HA2" QMgr="MFTDEMO"/>
<log:logMonitor>MON1="verbose"</log:logMonitor>
</log:log>
```

Zugehörige Verweise

Befehl ['fteSetAgentLogLevel'](#)

Die MFT [agent.properties](#)-Datei

MFT-Ressourcenüberwachung starten

Von IBM MQ 9.3.0 können Sie Ressourcenüberwachungen starten, ohne einen Agenten mit dem Befehl **fteStartMonitor** stoppen oder erneut starten zu müssen.

Vorbereitende Schritte

Wenn die Benutzerberechtigungsverwaltung aktiviert wurde, indem das Attribut **authorityChecking** in der Datei [agent.properties](#) auf **true** gesetzt wurde, müssen Sie über die Berechtigung **Überwachen** oder **Operationen überwachen** verfügen, um eine Ressourcenüberwachung zu starten. Weitere Informationen zur Verwaltung der Benutzerberechtigung finden Sie unter [Benutzerberechtigungen auf MFT-Agentenaktionen beschränken](#).

Informationen zu diesem Vorgang

Sie können den Befehl **fteStartMonitor** von jedem System aus ausführen, auf dem die Managed File Transfer-Befehlskomponente installiert ist. Dies bedeutet, dass Sie eine Ressourcenüberwachung von einer beliebigen Position aus starten können und nicht auf das System beschränkt sind, auf dem der Agent ausgeführt wird, der Eigner der Ressourcenüberwachung ist. Informationen zu den erforderlichen und optionalen Parametern für diesen Befehl finden Sie im Abschnitt [fteStartMonitor \(Eine MFT-Ressourcenüberwachung starten\)](#).

Prozedur

- Um den Status eines Agenten vor oder nach der Ausführung des Befehls **fteStartMonitor** zu ermitteln, verwenden Sie den Befehl [fteListMonitors](#) mit dem Parameter **-v**, wie im folgenden Beispiel gezeigt:

```
fteListMonitors -ma monitoring_agent_name -v
```

- Geben Sie zum Starten einer Ressourcenüberwachung in einem Agenten, der auf demselben System ausgeführt wird, den Befehl **fteStartMonitor** wie folgt ein:

```
fteStartMonitor -mn monitor_name -ma agent_name
```

- Wenn Sie eine Ressourcenüberwachung in einem Agenten starten möchten, der auf einem anderen System ausgeführt wird, geben Sie den Befehl **fteStartMonitor** wie folgt ein:

```
fteStartMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Wenn der Befehlswarteschlangenmanager auch der Agentenwarteschlangenmanager für den Überwachungsagenten ist, ist der Parameter **-mm** optional. Andernfalls müssen Sie den Agentenwarteschlangenmanager mit dem Parameter **-mm** angeben.

Ergebnisse

Wenn der Agent aktiv ist, wird die Ressourcenüberwachung gestartet, wenn sie derzeit gestoppt ist. Der Befehl gibt die folgenden Nachrichten aus und protokolliert ein Ereignis im `output0.log` des Agenten.

BFGCL0816I:
Eine Anforderung zum Starten der Ressourcenüberwachung '*Überwachungsname*' des Agenten '*Agentenname*' wurde ausgegeben.
BFGCL0251I: The request has successfully completed.

Informationen zu den Nachrichten, die der Befehl ausgibt, wenn er die Ressourcenüberwachung nicht starten kann, finden Sie in [fteStartMonitor \(Eine MFT-Ressourcenüberwachung starten\)](#).

Zugehörige Konzepte

„Konzepte der MFT-Ressourcenüberwachung“ auf Seite 240

Dieser Abschnitt enthält eine Übersicht über die wichtigsten Begriffe im Zusammenhang mit der Managed File Transfer-Ressourcenüberwachung.

Zugehörige Tasks

„MFT-Ressourcenüberwachung stoppen“ auf Seite 268

Von IBM MQ 9.3.0 können Sie die Ressourcenüberwachung stoppen, ohne einen Agenten mit dem Befehl **fteStopMonitor** stoppen oder erneut starten zu müssen.

Zugehörige Verweise

[fteStartMonitor \(Eine MFT-Ressourcenüberwachung starten\)](#)

MFT-Ressourcenüberwachung stoppen

Von IBM MQ 9.3.0 können Sie die Ressourcenüberwachung stoppen, ohne einen Agenten mit dem Befehl **fteStopMonitor** stoppen oder erneut starten zu müssen.

Vorbereitende Schritte

Wenn die Benutzerberechtigungsverwaltung aktiviert wurde, indem das Attribut **authorityChecking** in der Datei [agent.properties](#) auf **true** gesetzt wurde, müssen Sie über die Berechtigung **Überwachen** oder **Operationen überwachen** verfügen, um eine Ressourcenüberwachung zu stoppen. Weitere Informationen zur Verwaltung der Benutzerberechtigung finden Sie unter [Benutzerberechtigungen auf MFT-Agentenaktionen beschränken](#).

Informationen zu diesem Vorgang

Sie können den Befehl **fteStopMonitor** von jedem System aus ausführen, auf dem die Managed File Transfer-Befehlskomponente installiert ist. Dies bedeutet, dass Sie eine Ressourcenüberwachung von einer beliebigen Stelle aus stoppen können und nicht auf das System beschränkt sind, auf dem der Agent, der Eigner der Ressourcenüberwachung ist, ausgeführt wird. Informationen zu den erforderlichen und optionalen Parametern für diesen Befehl finden Sie in [fteStopMonitor \(Eine MFT-Ressourcenüberwachung stoppen\)](#).

Wenn eine Ressourcenüberwachung gestoppt wird, schreibt sie eine Nachricht in das Ereignisprotokoll des Agenten für Ressourcenüberwachung, `resmoneventnumber.log`. Wenn die Ressourcenüberwachung mit dem Befehl **fteStopMonitor** gestoppt wird, enthält die Nachricht den Namen des Benutzers, der die Stoppanforderung ausgegeben hat:

```
Ressourcenüberwachung durch Benutzer '<mquser_id>' gestoppt
```

Eine Ressourcenüberwachung wird automatisch gestartet, wenn ihr Agent erneut gestartet wird, auch wenn die Ressourcenüberwachung zuvor gestoppt wurde, indem der Befehl **fteStopMonitor** verwendet wurde.

Der Agentenprozess stoppt Überwachungsanforderungen seriell und nicht parallel. Wenn beispielsweise ein Agent eine Anforderung zum Stoppen der Überwachung M1 erhält und anschließend eine andere Anforderung, die Überwachung M2 in schneller Folge zu stoppen, so stoppt er zuerst M1, bevor er versucht, M2 zu stoppen.

Prozedur

- Um den Status eines Agenten vor oder nach der Ausführung des Befehls **fteStopMonitor** zu ermitteln, verwenden Sie den Befehl **fteListMonitors** mit dem Parameter **-v**, wie im folgenden Beispiel gezeigt:

```
fteListMonitors -ma monitoring_agent_name -v
```

- Wenn Sie eine Ressourcenüberwachung in einem Agenten stoppen möchten, der auf demselben System ausgeführt wird, geben Sie den Befehl **fteStopMonitor** wie folgt ein:

```
fteStopMonitor -mn monitor_name -ma agent_name
```

- Wenn Sie eine Ressourcenüberwachung in einem Agenten stoppen möchten, der auf einem anderen System ausgeführt wird, geben Sie den Befehl **fteStopMonitor** wie folgt ein:

```
fteStopMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Wenn der Befehlswarteschlangenmanager auch der Agentenwarteschlangenmanager für den Überwachungsagenten ist, ist der Parameter **-mm** optional. Andernfalls müssen Sie den Agentenwarteschlangenmanager mit dem Parameter **-mm** angeben.

Ergebnisse

Wenn der Agent aktiv ist, wird die Ressourcenüberwachung gestoppt, wenn sie gerade gestartet wird. Der Befehl gibt die folgenden Nachrichten aus und protokolliert ein Ereignis im `output0.log` des Agenten.

```
BFGCL0813I:  
Agent 'SOURCE' hat eine Anforderung zum Stoppen der  
Ressourcenüberwachung 'MNTR' ausgegeben.  
BFGCL0251I: The request has successfully completed.
```

Informationen zu den Nachrichten, die vom Befehl ausgegeben werden, wenn er die Ressourcenüberwachung nicht stoppen kann, finden Sie in [fteStopMonitor \(Eine MFT-Ressourcenüberwachung stoppen\)](#).

Zugehörige Konzepte

„Konzepte der MFT-Ressourcenüberwachung“ auf Seite 240

Dieser Abschnitt enthält eine Übersicht über die wichtigsten Begriffe im Zusammenhang mit der Managed File Transfer-Ressourcenüberwachung.

Zugehörige Tasks

„MFT-Ressourcenüberwachung starten“ auf Seite 267

Von IBM MQ 9.3.0 können Sie Ressourcenüberwachungen starten, ohne einen Agenten mit dem Befehl **fteStartMonitor** stoppen oder erneut starten zu müssen.

Zugehörige Verweise

[fteStopMonitor \(Eine MFT-Ressourcenüberwachung stoppen\)](#)

MFT-Ressourcenüberwachungen sichern und wiederherstellen

Sie können die Ressourcenmonitore sichern, die Sie für die zukünftige Verwendung zur Verfügung haben wollen, indem Sie ihre Definitionen in eine XML-Datei exportieren, die Sie dann importieren können, um einen neuen Ressourcenmonitor aus der Sicherung zu erstellen.

Informationen zu diesem Vorgang

Möglicherweise müssen Sie die zuvor definierten Ressourcenmonitore sichern, damit Sie ihre Definitionen in der Zukunft erneut verwenden können, z. B. um die Ressourcenmonitore in einer anderen Infrastruktur erneut zu erstellen oder wenn ein Ressourcenmonitor aufgrund von WS-Managerproblemen erneut erstellt werden muss.

Sie können eine einzelne Ressourcenmanagerdefinition sichern, indem Sie entweder den Befehl **fteCreateMonitor** oder den Befehl **fteListMonitors** mit dem Parameter **-ox** verwenden. In beiden Fällen wird die Ressourcenmanagerdefinition gesichert, indem sie in eine XML-Datei exportiert wird.

Anschließend können Sie mit dem Parameter **-ix** des Befehls **fteCreateMonitor** einen neuen Ressourcenmanager erstellen, indem Sie die Definition aus der XML-Datei importieren.

Mit dem Parameter **-ox** können Sie jeweils nur eine Ressourcenmonitordefinition sichern.

Der Parameter **-od** wird zum Befehl **fteListMonitors** hinzugefügt. Wenn Sie diesen Parameter angeben, können Sie mehrere Ressourcenmonitorressourcen gleichzeitig sichern, indem Sie die zugehörigen Definitionen in Massendaten in ein angegebener Verzeichnis exportieren. Jede Ressourcenüberwachungsdefinition wird in einer separaten XML-Datei mit einem Namen im Format *agent name.monitor name.xml* gespeichert.

Der Parameter **-od** ist besonders nützlich, wenn Sie über eine große Anzahl von Ressourcenmonitoren verfügen, die Sie sichern wollen, da Sie den Befehl **fteListMonitors -od** nur einmal ausführen müssen, anstatt den **fteListMonitors -ox**-Befehl separat für jede Ressourcendefinition ausführen zu müssen, oder ein separates Script verwenden müssen, um den Befehl **fteListMonitors -ox** für jede Ressourcenüberwachung auszuführen.

Prozedur

- Verwenden Sie einen der folgenden Befehle, um die Definition einer Ressourcenüberwachung zu sichern, indem Sie sie in eine XML-Datei exportieren:

- Der Befehl **fteCreateMonitor** mit dem Parameter **-ox**.
- Der Befehl **fteListMonitors** mit dem Parameter **-ox**.

Wenn Sie den Parameter **-ox** verwenden, müssen Sie auch die Parameter **-ma** und **-mn** angeben, wie im folgenden Beispiel gezeigt:

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -ox filename1.xml
```

- Verwenden Sie den Befehl **fteListMonitors** mit dem Parameter **-od** wie im folgenden Beispiel gezeigt, um mehrere Ressourcenüberwachungsdefinitionen zu sichern, indem Sie sie in XML-Dateien in einem angegebenen Verzeichnis exportieren:

```
fteListMonitors -od /usr/mft/resmonbackup
```

Sie müssen ein gültiges Zielverzeichnis angeben, wenn Sie Ressourcenmonitore in Massendaten sichern. Wenn Sie keinen Zielpfad angeben, führt dies zu einer Fehlernachricht wie im folgenden Beispiel gezeigt:

```
BFGCL0762E: Das Ausgabeverzeichnis wurde nicht angegeben. Führen Sie den Befehl erneut aus, und geben Sie einen gültigen Pfad an.
```

Der Parameter **-od** darf nicht mit dem Parameter **-ox** kombiniert werden, da andernfalls die folgende Fehlernachricht angezeigt wird:

```
BFGCL0761E: Es ist nicht zulässig, die Parameter '-od' und '-ox' zusammen anzugeben.
```

Sie können eine bestimmte Gruppe von Ressourcenmonitoren definieren, die in die Sicherung eingeschlossen werden sollen. Wenn Sie beispielsweise den Parameter **-ma** verwenden, um den Namen eines Agenten anzugeben, können Sie alle Ressourcenmonitore für diesen Agenten sichern, wie im folgenden Beispiel gezeigt:

```
fteListMonitors -ma AGENT1 -od /usr/mft/resmonbackup
```

Sie können auch Platzhalterzeichen verwenden, indem Sie ein Platzhalterzeichen (*) verwenden, wenn Sie ein Muster definieren, das für übereinstimmende Agentennamen oder Monitornamen oder beides verwendet werden soll. Im folgenden Beispiel werden alle Ressourcenmonitore gesichert, deren Namen mit einem bestimmten Muster übereinstimmen und die sich in einem Agenten befinden, der einen Namen hat, der mit einem angegebenen Muster übereinstimmt:

```
fteListMonitors -ma AGENT* -mn MON* -od /usr/mft/resmonbackup
```

Während der Befehl ausgeführt wird, werden die folgenden Statusberichtsnachrichten angezeigt:

Es wurden insgesamt *number* übereinstimmende Ressourcenmonitordefinitionen gefunden. *index* von *number* -Ressourcenmonitordefinitionen wurden in einem Dateisystem gespeichert.

Wenn Sie die Option 'verbose' verwenden, wird die laufende Summe weiterhin angezeigt, aber nicht angezeigt.

index von *number* -Ressourcenmonitordefinitionen im Dateisystem gespeichert

Mit dem Befehl wird der Name der zu speichernden Monitordefinition angezeigt, z. B.:

BFGCL0762I: Definition des Überwachungsprogramms 'FILEMON' des Agenten 'XFERAGENT' als FILEMON.XFERAGENT.XML in Dateisystem.

- Verwenden Sie den Befehl **fteListMonitors** mit dem Parameter **-od**, um eine Ressourcenüberwachung für einen bestimmten Agenten zu sichern, indem Sie sie in eine XML-Datei in einem angegebenen Verzeichnis exportieren:

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -od /usr/mft/resmonbackup
```

Die Verwendung des Parameters **-od** zum Sichern eines einzelnen Ressourcenmonitors ähnelt der Verwendung des Parameters **-ox**, mit der Ausnahme, dass der Ausgabedateiname das Format *agent name.monitor name.xml* hat.

- Verwenden Sie den Befehl **fteCreateMonitor** mit dem Parameter **-ix** wie im folgenden Beispiel gezeigt, um Ressourcenüberwachungsdefinitionen aus einer Sicherung wiederherzustellen:

```
fteCreateMonitor -ix file name
```

Weitere Beispiele zur Verwendung des Parameters **-od** finden Sie unter [fteListMonitors: MFT-Ressourcenmonitor auflisten](#).

Zugehörige Verweise

[fteCreateMonitor: MFT-Ressourcenüberwachung erstellen](#)

[fteListMonitors: MFT-Ressourcenüberwachungen auflisten](#)

Protokoll der Ressourcenüberwachung löschen

Sie können das Protokoll einer Ressourcenüberwachung löschen, damit eine andere Dateiübertragungsanforderung für eine Datei übergeben werden kann, die aufgrund eines Fehlers vorher nicht übertragen wurde. Zum Löschen des Protokolls für die Ressourcenüberwachung verwenden Sie den Befehl **fteClearMonitorHistory** oder den IBM MQ Explorer.

Vorbereitende Schritte

Wenn die Benutzerberechtigungsverwaltung aktiviert wurde, indem das Attribut **authorityChecking** in der Datei *agent.properties* auf **true** gesetzt wurde, muss der Benutzer, der das Überwachungsprotokoll löscht, über die entsprechende Berechtigung verfügen, wie in der folgenden Tabelle dargestellt.

Tabelle 15. Erforderliche Benutzerberechtigung für die Ausführung des Befehls fteClearMonitorHistory		
Benutzer, der das Überwachungsprotokoll löscht	MFT-Zugriffsberechtigung	Erforderliche Berechtigung
Der gleiche Benutzer wie der, der die Ressourcenüberwachung erstellt hat.	Überwachung	BROWSE auf SYSTEM.FTE.AUTH-MON1.<monitor_agentename> Dies entspricht der Berechtigung, die zum Erstellen oder Löschen der Ressourcenüberwachung erforderlich ist.

Tabelle 15. Erforderliche Benutzerberechtigung für die Ausführung des Befehls **fteClearMonitorHistory** (Forts.)

Benutzer, der das Überwachungsprotokoll löscht	MFT-Zugriffsberechtigung	Erforderliche Berechtigung
Jeder Benutzer mit Ausnahme des Benutzers, der die Ressourcenüberwachung erstellt hat.	Überwachung von Operationen	SET auf SYS-TEM.FTE.AUTHPS1.<Agentenna-me> Dies entspricht der Berechtigung, die zum Löschen der Ressourcenüberwachung erforderlich ist.

Weitere Informationen zur Verwaltung der Benutzerberechtigung finden Sie unter [Benutzerberechtigungen auf MFT-Agentenaktionen beschränken](#).

Wenn ein Benutzer ohne die erforderliche Berechtigung versucht, das Ressourcenüberwachungsprotokoll zu löschen, gibt der Befehl **fteClearMonitorHistory** eine Fehlermeldung aus und protokolliert den Fehler in der Datei `output0.log` des Agenten. Weitere Informationen finden Sie unter [fteClearMonitorHistory: Protokoll der Ressourcenüberwachung löschen](#).

Informationen zu diesem Vorgang

Wenn eine Dateiübertragung eingeleitet wurde und eine Datei aus einem beliebigen Grund nicht übertragen werden kann, wählt die Ressourcenüberwachung bei der nächsten Abfrage diese Datei nicht erneut zur Übertragung aus, da im Überwachungsprotokoll angezeigt wird, dass die Datei in einer früheren Abfrage angezeigt und seither nicht mehr geändert wurde (siehe „Konzepte der MFT-Ressourcenüberwachung“ auf Seite 240).

Wenn die Übertragung einer Datei in einer Version vor IBM MQ 9.1.3 fehlschlägt, kann die Dateiübertragung nur erneut eingeleitet werden, wenn die Datei gelöscht und erneut in das Verzeichnis eingefügt wird, wenn die Datei aktualisiert wird, damit das Attribut mit dem Datum der letzten Änderung geändert wird oder wenn die Ressourcenüberwachung selbst erneut erstellt wird.

Sie können das Ressourcenüberwachungsprotokoll jedoch entweder mit dem Befehl **fteClearMonitorHistory** oder mit dem IBM MQ Explorer löschen. Durch das Löschen des Protokolls ist eine weitere Übertragungsanforderung für eine Datei möglich, die nicht übertragen werden konnte, ohne dass die Datei gelöscht und wieder in das Verzeichnis eingefügt oder aktualisiert werden muss, um das Attribut für das Datum der letzten Änderung zu aktualisieren. Dies ist beispielsweise in Situationen hilfreich, wenn die Datei übertragen werden muss, aber eine Änderung der Datei nicht möglich ist. Wenn das Protokoll einer Ressourcenüberwachung gelöscht werden kann, bedeutet das auch, dass die Ressourcenüberwachung nicht erneut erstellt werden muss, um eine weitere Übertragungsanforderung für eine Datei zu übergeben, die nicht übertragen wurde.

 Das im Lieferumfang von Managed File Transfer on z/OS enthaltene SCSQFCMD-Beispielmember enthält ein JCL-Script zum Löschen des Monitorprotokolls.

Prozedur

- Um mit dem Befehl **fteClearMonitorHistory** das Protokoll der Ressourcenüberwachung zu löschen, geben Sie ihn im folgenden Format ein:

```
fteClearMonitorHistory -p <configuration> -ma <agent name> -mn <monitor name> -w 1000
```

Es sind nur die Parameter **-ma** und **-mn** erforderlich. Alle anderen Parameter sind optional. Weitere Informationen zur Verwendung des Befehls **fteClearMonitorHistory** sowie Beispiele dazu finden Sie unter [fteClearMonitorHistory: Protokoll der Ressourcenüberwachung löschen](#).

Wenn das Protokoll erfolgreich gelöscht wurde, gibt der Befehl die folgende Nachricht aus:

BFGCL0780I: Eine Anforderung zum Löschen des Protokolls der Ressourcenüberwachung 'Überwachungsname' des Agenten 'Agentenname' wurde ausgegeben.
BFGCL0251I: The request has successfully completed.

und protokolliert den Erfolg in der Datei output0.log des Agenten.

Wenn der Versuch, das Ressourcenüberwachungsprotokoll zu löschen, fehlschlägt, gibt **fteClearMonitorHistory** eine Fehlernachricht aus und protokolliert den Fehler in der Datei output0.log des Agenten.

- Wenn Sie die Ressourcenüberwachungsansicht im IBM MQ Explorer MFT-Plug-in verwenden möchten, um das Protokoll der Ressourcenüberwachung zu löschen, klicken Sie mit der rechten Maustaste auf die Ressourcenüberwachung und wählen Sie im Dropdown-Menü die Option **Clear History** aus.

Wenn das Protokoll erfolgreich gelöscht wurde, wird die folgende Nachricht angezeigt:

BFGUI00171: Protokoll der Ressourcenüberwachung erfolgreich gelöscht.

Wenn der Versuch, das Protokoll zu löschen, fehlschlägt, wird eine Fehlernachricht angezeigt. For example:

BFGUI0016E Das Protokoll der angegebenen Ressourcenüberwachung konnte nicht gelöscht werden - Ursache 2059

Mit Dateiübertragungsvorlagen arbeiten

Sie können Dateiübertragungsschablonen verwenden, um häufige Dateiübertragungseinstellungen für wiederholte oder komplexe Übertragungen zu speichern. Sie können entweder über die Befehlszeile mit dem Befehl **fteCreateTemplate** oder über IBM MQ Explorer mit dem Assistenten **Neue Vorlage für Managed File Transfer erstellen** eine Übertragungsvorlage erstellen oder bei der Erstellung einer Dateiübertragung eine Vorlage speichern, indem Sie das Kontrollkästchen **Übertragungseinstellungen als Vorlage sichern** aktivieren. Im Fenster **Übertragungsvorlagen** werden alle Übertragungsvorlagen angezeigt, die Sie in Ihrem Managed File Transfer-Netz erstellt haben.

Informationen zu diesem Vorgang

Zur Erstellung einer Übertragungsvorlage über die Befehlszeile verwenden Sie den Befehl **fteCreateTemplate**. Wenn Sie anschließend eine von Ihnen in der Befehlszeile erstellte Übertragungsschablone übergeben möchten, klicken Sie in IBM MQ Explorer auf **Übergeben**.

So zeigen Sie Übertragungsvorlagen in IBM MQ Explorer an:

Vorgehensweise

1. Erweitern Sie den Knoten **Verwaltete Dateiübertragung** in der Navigatoransicht. In der Inhaltsansicht wird die **Verwaltete Dateiübertragungszentrale** angezeigt.
2. Alle Koordinationswarteschlangenmanager werden in der Navigatoransicht angezeigt. Erweitern Sie den Namen des Koordinations-Warteschlangenmanagers, den Sie für die geplante Übertragung verwendet haben. Wenn Sie den Koordinations-Warteschlangenmanager ändern möchten, mit dem Sie verbunden sind, klicken Sie mit der rechten Maustaste auf den Namen des Koordinations-Warteschlangenmanagers, den Sie in der Navigatoransicht verwenden möchten, und klicken Sie auf **Verbinden**.
3. Klicken Sie auf **Übertragungsvorlagen**. Das Fenster **Übertragungsvorlagen** wird in der Inhaltsansicht angezeigt.
4. Im Fenster **Übertragungsvorlagen** werden die folgenden Angaben zu den Dateiübertragungen angezeigt:
 - a) **Name** Der Name Ihrer Dateiübertragungsvorlage.
 - b) **Quelle** Der Name des Agenten, der zum Übertragen der Datei aus dem Quellensystem verwendet wird.
 - c) **Quellendatei** Der Name der auf ihr Hostsystem zu übertragenden Datei.
Zur Anzeige dieses Felds müssen die Informationen der Übertragungsvorlage erweitert werden.
 - d) **Ziel** Der Name des Agenten, der zum Empfangen der Datei am Zielsystem verwendet wird.

- e) **Zieldatei** Der Name der Datei, nachdem diese auf das Zielsystem übertragen wurde.
Zur Anzeige dieses Felds müssen die Informationen der Übertragungsvorlage erweitert werden.
- f) **Geplanter Start (ausgewählte Zeitzone)** Der geplante Zeitpunkt (Datum und Uhrzeit) für den Start der Dateiübertragung in der vom Administrator verwendeten Zeitzone. Klicken Sie zum Ändern der angezeigten Zeitzone auf **Fenster > Benutzervorgaben > IBM MQ Explorer > Managed File Transfer** und wählen Sie eine alternative Zeitzone in der Liste **Zeitzone:** aus. Klicken Sie auf **OK**.
- g) **Auslöseereignisse** Der Ereignistyp, der den Start der Dateiübertragung auslöst. Der Typ kann einer der folgenden Werte sein: Ist vorhanden, Ist nicht vorhanden oder überschreitet.

Ergebnisse

Sie können den Inhalt des Fensters **Übertragungsvorlagen** aktualisieren, indem Sie in der Symbolleiste der Inhaltsansicht auf 'Aktualisieren'  klicken.

Klicken Sie zum Abschicken einer Übertragungsvorlage und zum Starten der in der Vorlage definierten Übertragung mit der rechten Maustaste auf die Vorlage, und klicken Sie auf **Übergeben**.

Um eine Übertragungsvorlage zu ändern, klicken Sie mit der rechten Maustaste auf den Namen der Vorlage und klicken Sie auf **Bearbeiten**. Alle in der ursprünglichen Vorlage enthaltenen Dateien werden als Teil einer Übertragungsgruppe aufgeführt, selbst wenn sie in der ursprünglichen Vorlage nicht zu einer Gruppe zusammengefasst waren. Zum Entfernen einer Datei aus der Vorlage müssen Sie die Dateispezifikation aus der Gruppe auswählen und auf **Auswahl entfernen** klicken. Zum Hinzufügen neuer Dateispezifikationen verwenden Sie hingegen die Felder der Vorlagenanzeige und klicken auf **Zu Gruppe hinzufügen**. Sobald Sie Ihre Änderungen vorgenommen haben, werden Sie zur Eingabe eines neuen Namens für die bearbeitete Vorlage aufgefordert.

Um eine Dateiübertragung auf Basis einer Übertragungsvorlage zu erstellen, klicken Sie mit der rechten Maustaste auf den Namen der Vorlage und klicken Sie auf **Als neue Übertragung bearbeiten**.

Klicken Sie zum Erstellen einer Duplikatkopie einer Übertragungsvorlage mit der rechten Maustaste auf den Vorlagennamen, und klicken Sie auf **Kopieren**. Die duplizierte Übertragungsvorlage wird automatisch unter dem Namen der ursprünglichen Vorlage, an den "(copy)" angehängt wird, gespeichert.

Zum Löschen einer Übertragungsvorlage führen Sie einen Rechtsklick auf den Vorlagennamen durch und klicken Sie dann auf **Löschen**.

Zugehörige Tasks

„Dateiübertragungsvorlage mit IBM MQ Explorer erstellen“ auf Seite 274

Eine Dateiübertragungsvorlage können Sie in IBM MQ Explorer oder über die Befehlszeile erstellen. Sie können diese Vorlage verwenden, um mit den Vorlagendetails eine neue Dateiübertragung zu erstellen, oder um die Vorlage abzuschicken, um die Dateiübertragung zu starten.

Zugehörige Verweise

[fteCreateTemplate](#): Neue Dateiübertragungsschablone erstellen

[fteListTemplates](#)

[fteDeleteTemplates](#)

Dateiübertragungsvorlage mit IBM MQ Explorer erstellen

Eine Dateiübertragungsvorlage können Sie in IBM MQ Explorer oder über die Befehlszeile erstellen. Sie können diese Vorlage verwenden, um mit den Vorlagendetails eine neue Dateiübertragung zu erstellen, oder um die Vorlage abzuschicken, um die Dateiübertragung zu starten.

Informationen zu diesem Vorgang

Zur Erstellung einer Dateiübertragungsschablone über die Befehlszeile verwenden Sie den Befehl [fteCreateTemplate](#).

Führen Sie die folgenden Schritte aus, um eine Dateiübertragungsschablone mit dem Assistenten **Neue Schablone für Managed File Transfer erstellen** in IBM MQ Explorer zu erstellen:

Vorgehensweise

1. Klicken Sie in der Navigatoransicht auf **Verwaltete Dateiübertragung**. In der Inhaltsansicht wird die **Verwaltete Dateiübertragungszentrale** angezeigt.
2. In der Navigatoransicht werden alle Ihre Koordinations-Warteschlangenmanager angezeigt. Erweitern Sie den Namen des Koordinations-Warteschlangenmanagers, den Sie für die geplante Übertragung verwendet haben. Wenn Sie den Koordinations-Warteschlangenmanager ändern möchten, mit dem Sie verbunden sind, klicken Sie mit der rechten Maustaste auf den Namen des Koordinations-Warteschlangenmanagers, den Sie in der Navigatoransicht verwenden möchten, und klicken Sie auf **Verbinden**.
3. Starten Sie den Assistenten **Erstellen von Neuen Schablonen für Verwaltete Dateiübertragungen**, indem Sie mit der rechten Maustaste auf **Übertragungsschablonen** und anschließend auf **Neue Schablone** klicken.
4. Befolgen Sie die Anweisungen in der Assistentenanzeige. Für jede Anzeige steht eine kontextabhängige Hilfe zur Verfügung. Zum Zugriff auf die kontextabhängige Hilfe unter Windows drücken Sie F1. Unter Linux drücken Sie Ctrl+F1 oder Shift+F1.

Wenn Ihre neue Schablone alle erforderlichen Übertragungsdetails enthält, aktivieren Sie auf der Seite **Übertragungszusammenfassung** das Kontrollkästchen **Übertragungseinstellungen als Schablone sichern**, sofern es noch nicht aktiviert ist. Geben Sie dann im Feld "Name" einen Namen für die Schablone ein. Wenn Sie eine Schablone erstellen, die noch nicht alle erforderlichen Übertragungsdetails enthält, ist das Kontrollkästchen **Übertragungseinstellungen als Schablone sichern** automatisch aktiviert.

Zugehörige Tasks

„Mit Dateiübertragungsvorlagen arbeiten“ auf Seite 273

Sie können Dateiübertragungsschablonen verwenden, um häufige Dateiübertragungseinstellungen für wiederholte oder komplexe Übertragungen zu speichern. Sie können entweder über die Befehlszeile mit dem Befehl **fteCreateTemplate** oder über IBM MQ Explorer mit dem Assistenten **Neue Vorlage für Managed File Transfer erstellen** eine Übertragungsvorlage erstellen oder bei der Erstellung einer Dateiübertragung eine Vorlage speichern, indem Sie das Kontrollkästchen **Übertragungseinstellungen als Vorlage sichern** aktivieren. Im Fenster **Übertragungsvorlagen** werden alle Übertragungsvorlagen angezeigt, die Sie in Ihrem Managed File Transfer-Netz erstellt haben.

Zugehörige Verweise

fteCreateTemplate: Neue Dateiübertragungsschablone erstellen

[fteListTemplates](#)

[fteDeleteTemplates](#)

Definition für Dateiübertragungsvorlage sichern

Dateiübertragungsvorlagen enthalten ein XML-Dokument, in dem die Spezifikationen für die Quellen- und Zieldatei für die Übertragung definiert werden. Sie können diese XML-Datei als Eingabe im Befehl **fteCreateTemplate** verwenden, um eine Dateiübertragungsvorlage erneut zu erstellen.

Informationen zu diesem Vorgang

Um das XML-Dokument mit den Spezifikationen für die Quellen- und Zieldateien für eine Übertragungsvorlage zu sichern, verwenden Sie den Befehl [fteCreateTransfer](#) oder den IBM MQ Explorer. Gehen Sie zum Erstellen einer XML-formatierten Sicherungsdatei für die Übertragungsvorlage folgendermaßen vor:

Prozedur

- Methode eins: Generieren Sie mit dem Parameter **-gt** in einem [fteCreateTransfer](#)-Befehl eine XML-Nachricht für die Übertragungsvorlage an eine neue Datei.
- Methode zwei: Erstellen Sie die Vorlage mit dem IBM MQ Explorer.
Gehen Sie auf der Seite *Zusammenfassung der Übertragungsschablone* folgendermaßen vor:

- a) Kopieren Sie die *XML-Voranzeige der Anforderungsnachricht*.
- b) Speichern Sie diese XML-Nachricht für die Übertragungsvorlage in einer neuen Datei.
- Methode drei: Verwenden Sie den IBM MQ Explorer, um vorhandene Vorlagen zu sichern.
 - a) Wechseln Sie zu **Managed File Transfer > Name des Warteschlangenmanagers > Übertragungsvorlagen**.
 - b) Markieren Sie in der Anzeige 'Übertragen' die Vorlage, die gesichert werden soll, klicken Sie mit der rechten Maustaste und wählen Sie im Popup-Menü den Eintrag **Bearbeiten** aus.
 - c) Klicken Sie auf **Weiter**, bis Sie zur Seite *Zusammenfassung der Übertragungsschablone* gelangen.
 - d) Kopieren Sie die *XML-Voranzeige der Anforderungsnachricht*.
 - e) Speichern Sie diese XML-Nachricht für die Übertragungsvorlage in einer neuen Datei.

Ergebnisse

Die XML-Nachrichtendatei für die Übertragungsvorlage, die Sie mit einem der oben genannten Methoden erstellt haben, können Sie als Eingabe im Befehl `fteCreateTemplate` verwenden. In den Informationen zum Befehl `fteCreateTemplate` finden Sie weitere Einzelheiten zur Verwendung dieses Befehls.

Zugehörige Verweise

[Befehl 'fteCreateTemplate'](#)

[fteListTemplates](#), [Befehl](#)

Daten von Dateien in Nachrichten übertragen

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

Im Abschnitt „[Daten von Nachrichten in Dateien übertragen](#)“ auf Seite 292 finden Sie Informationen zur Übertragung von Nachrichten an Dateien.

Der Zielagent bei einer Datei-an-Nachricht-Übertragung darf kein Protokollbridgeagent oder Connect:Direct-Bridgeagent sein.

Sie können Dateidaten an IBM MQ-Nachrichtendaten übertragen. Die IBM MQ-Nachrichten können von Anwendungen gelesen und verwendet werden. Die folgenden Typen von Datei-zu-Nachricht-Übertragung werden unterstützt:

- Aus einer einzelnen Datei in eine einzelne Nachricht. Für die Nachricht ist keine IBM MQ-Gruppen-ID festgelegt.
- Aus einer einzelnen Datei in mehrere Nachrichten, indem die Datei in Nachrichten mit einer bestimmten Länge aufgeteilt wird. Diese Nachrichten haben alle die gleiche IBM MQ-Gruppen-ID.
- Aus einer einzelnen Datei an mehrere Nachrichten, indem eine Textdatei am Begrenzer eines regulären Java-Ausdrucks aufgeteilt wird. Diese Nachrichten haben alle die gleiche IBM MQ-Gruppen-ID.
- Aus einer einzelnen Datei in mehrere Nachrichten, indem eine Binärdatei auf einen hexadezimalen Begrenzer aufgeteilt wird. Diese Nachrichten haben alle die gleiche IBM MQ-Gruppen-ID.

Wenn Sie eine Binärdatei mit einer Bytefolge als Begrenzer aufteilen möchten, verwenden Sie den Parameter `-sqdb` des Befehls `fteCreateTransfer`. Weitere Informationen finden Sie im Abschnitt [-sqdb-Parameter](#).

Standardmäßig sind die Nachrichten, die von einer Datei-in-Nachricht-Übertragung erstellt werden, permanent. Die Nachrichten können so eingestellt werden, dass sie nicht persistent sind oder dass der Persistenzwert durch die Zielwarteschlange definiert wird.

Wenn Sie angeben, dass eine Datei in mehrere Nachrichten aufgeteilt werden soll, erhalten alle aus der Datei erstellten Nachrichten die gleiche IBM MQ-Gruppen-ID. Wenn Sie keine Aufteilung in mehrere Nachrichten angeben, wird aus der Datei nur eine Nachricht erstellt, die dann keine IBM MQ-Gruppen-ID erhält.

Bei der Übertragung von Dateien in große oder viele kleine Nachrichten müssen unter Umständen einige IBM MQ- oder Managed File Transfer-Eigenschaften geändert werden. Weitere Informationen zu finden Sie unter [Anleitung zum Festlegen von MQ-Attributen und MFT-Eigenschaften, die der Nachrichtengröße zugeordnet sind](#).

Anmerkung: Wenn die Zielwarteschlange entweder eine Clusterwarteschlange oder ein Aliasname für eine Clusterwarteschlange ist, wird beim Übertragen einer Datei in eine Warteschlange eine Fehlermeldung angezeigt, wenn die Agenteneigenschaft 'enableClusterQueueInputOutput' nicht auf 'true' gesetzt wurde. Weitere Informationen finden Sie unter [Was tun, wenn die Zielwarteschlange eine Clusterwarteschlange ist, oder ein Aliasname für eine Clusterwarteschlange](#)

Zugehörige Tasks

„Agenten zum Ausführen von Datei-zu-Nachricht-Übertragungen konfigurieren“ auf Seite 278
Standardmäßig können Agenten keine Datei-zu-Nachricht- oder Nachricht-zu-Datei-Übertragungen ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft 'enableQueueInputOutput' auf den Wert 'true' setzen. Damit ein Schreibzugriff auf IBM MQ-Clusterwarteschlangen möglich ist, müssen Sie die Agenteneigenschaft 'enableClusterQueueInputOutput' ebenfalls auf 'true' setzen.

„Beispiel: Übertragen einer einzelnen Datei zu einer einzelnen Nachricht“ auf Seite 279
Sie können eine Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-dq** beim Befehl **fteCreateTransfer** verwenden. Die Quelldatei muss kleiner sein als die maximale Nachrichtenlänge, die bei der Zielwarteschlange eingestellt ist. Die Zielwarteschlange muss sich nicht unbedingt auf dem Warteschlangenmanager befinden, zu dem der Zielagent eine Verbindung herstellt, die beiden zugehörigen Warteschlangenmanager müssen jedoch miteinander kommunizieren können.

„Beispiel: Einzelne Datei in mehrere Nachrichten nach Länge aufteilen“ auf Seite 281
Sie können eine Datei mit dem Parameter **-qs** des Befehls **fteCreateTransfer** in mehrere IBM MQ-Nachrichten aufteilen. Die Datei wird in Abschnitte mit fester Länge aufgeteilt, wobei jeder Abschnitt in eine einzelne Nachricht geschrieben wird.

„Beispiel: Eine Textdatei mit einem regulären Ausdruckbegrenzer trennen und den Begrenzer in den Nachrichten eingrenzen“ auf Seite 284
Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen und den über den regulären Ausdruck erhaltenen Treffer in die Nachrichten einfügen. Dazu verwenden Sie die Parameter **-dqdt** und **-qi** des Befehls **fteCreateTransfer**.

„Beispiel: Aufteilen einer Textdatei in mehrere Nachrichten mithilfe eines regulären Ausdrucks als Begrenzer“ auf Seite 282
Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen. Verwenden Sie hierfür den Parameter **-dqdt** des Befehls **fteCreateTransfer**.

„Beispiel: Einstellen der IBM MQ-Nachrichteneigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 287
Sie können mit dem Parameter **-qmp** beim Befehl **fteCreateTransfer** angeben, ob IBM MQ-Nachrichteneigenschaften bei der ersten Nachricht festgelegt werden, die von der Übertragung in die Zielwarteschlange geschrieben wird. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Beispiel: Einstellen von benutzerdefinierten Eigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 288
Benutzerdefinierte Metadaten werden bei der ersten Nachricht, die von der Übertragung in die Zielwarteschlange geschrieben wird, als IBM MQ-Nachrichteneigenschaften festgelegt. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Neue Dateiübertragung starten“ auf Seite 228

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

Zugehörige Verweise

[„Fehler bei Übertragung von Datei-zu-Nachrichten“ auf Seite 291](#)

Wenn eine Übertragung von Dateien zu Nachrichten fehlschlägt, nachdem der Agent mit dem Schreiben von Dateidaten in die Zielwarteschlange begonnen hat, schreibt der Agent eine Nachricht in die Warteschlange, um auf eine Anwendung hinzuweisen, die die Nachrichten konsumiert, die ein Fehler aufgetreten ist.

[MQ-Nachrichteneigenschaften, die von MFT für Nachrichten festgelegt werden, die in Zielwarteschlangen geschrieben werden](#)

[Anleitung zum Festlegen von MQ-Attributen und MFT-Eigenschaften, die der Nachrichtengröße zugeordnet sind](#)

Agenten zum Ausführen von Datei-zu-Nachricht-Übertragungen konfigurieren

Standardmäßig können Agenten keine Datei-zu-Nachricht- oder Nachricht-zu-Datei-Übertragungen ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft 'enableQueueInputOutput' auf den Wert 'true' setzen. Damit ein Schreibzugriff auf IBM MQ-Clusterwarteschlangen möglich ist, müssen Sie die Agenteneigenschaft 'enableClusterQueueInputOutput' ebenfalls auf 'true' setzen.

Informationen zu diesem Vorgang

Wenn Sie versuchen, eine Datei-zu-Nachricht-Übertragung an einen Zielagenten auszuführen, dessen Eigenschaft enableQueueInputOutput nicht auf 'true' gesetzt ist, schlägt die Übertragung fehl. Die im Koordinationswarteschlangenmanager veröffentlichte Übertragungsprotokollnachricht enthält folgende Nachricht:

```
BFGI00197E: An attempt to write to a queue was rejected by the destination agent. The agent must have enableQueueInputOutput=true set in the agent.properties file to support transferring to a queue.
```

Gehen Sie wie folgt vor, um dem Agenten die Möglichkeit zu geben, in Warteschlangen zu schreiben und aus Warteschlangen zu lesen:

Vorgehensweise

1. Stoppen Sie den Zielagenten mit dem Befehl **fteStopAgent**.
2. Bearbeiten Sie die Datei agent.properties, um die Zeile enableQueueInputOutput=true einzuschließen.
Die Datei agent.properties befindet sich im Verzeichnis *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name*.
3. Optional: Bearbeiten Sie die Datei agent.properties, um die Zeile enableClusterQueueInputOutput=true einzuschließen. Die Datei agent.properties befindet sich im Verzeichnis *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name*.
4. Starten Sie den Zielagenten mit dem Befehl **fteStartAgent**.

Zugehörige Konzepte

[„Daten von Dateien in Nachrichten übertragen“ auf Seite 276](#)

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

Zugehörige Tasks

[„Beispiel: Übertragen einer einzelnen Datei zu einer einzelnen Nachricht“ auf Seite 279](#)

Sie können eine Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-dq** beim Befehl **fteCreateTransfer** verwenden. Die Quellendatei muss kleiner sein als die maximale Nachrichtenlänge, die bei der Zielwarteschlange eingestellt ist. Die Zielwarteschlange muss sich nicht

unbedingt auf dem Warteschlangenmanager befinden, zu dem der Zielagent eine Verbindung herstellt, die beiden zugehörigen Warteschlangenmanager müssen jedoch miteinander kommunizieren können.

„Beispiel: Einzelne Datei in mehrere Nachrichten nach Länge aufteilen“ auf Seite 281

Sie können eine Datei mit dem Parameter **-qs** des Befehls **fteCreateTransfer** in mehrere IBM MQ-Nachrichten aufteilen. Die Datei wird in Abschnitte mit fester Länge aufgeteilt, wobei jeder Abschnitt in eine einzelne Nachricht geschrieben wird.

„Beispiel: Eine Textdatei mit einem regulären Ausdruckbegrenzer trennen und den Begrenzer in den Nachrichten eingrenzen“ auf Seite 284

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen und den über den regulären Ausdruck erhaltenen Treffer in die Nachrichten einfügen. Dazu verwenden Sie die Parameter **-dqdt** und **-qi** des Befehls **fteCreateTransfer**.

„Beispiel: Aufteilen einer Textdatei in mehrere Nachrichten mithilfe eines regulären Ausdrucks als Begrenzer“ auf Seite 282

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen. Verwenden Sie hierfür den Parameter **-dqdt** des Befehls **fteCreateTransfer**.

„Beispiel: Einstellen der IBM MQ-Nachrichteneigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 287

Sie können mit dem Parameter **-qmp** beim Befehl **fteCreateTransfer** angeben, ob IBM MQ-Nachrichteneigenschaften bei der ersten Nachricht festgelegt werden, die von der Übertragung in die Zielwarteschlange geschrieben wird. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Beispiel: Einstellen von benutzerdefinierten Eigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 288

Benutzerdefinierte Metadaten werden bei der ersten Nachricht, die von der Übertragung in die Zielwarteschlange geschrieben wird, als IBM MQ-Nachrichteneigenschaften festgelegt. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

Zugehörige Verweise

fteStopAgent

fteStartAgent

Die MFT `agent.properties`-Datei

„Fehler bei Übertragung von Datei-zu-Nachrichten“ auf Seite 291

Wenn eine Übertragung von Dateien zu Nachrichten fehlschlägt, nachdem der Agent mit dem Schreiben von Dateidaten in die Zielwarteschlange begonnen hat, schreibt der Agent eine Nachricht in die Warteschlange, um auf eine Anwendung hinzuweisen, die die Nachrichten konsumiert, die ein Fehler aufgetreten ist.

Beispiel: Übertragen einer einzelnen Datei zu einer einzelnen Nachricht

Sie können eine Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-dq** beim Befehl **fteCreateTransfer** verwenden. Die Quellendatei muss kleiner sein als die maximale Nachrichtenlänge, die bei der Zielwarteschlange eingestellt ist. Die Zielwarteschlange muss sich nicht unbedingt auf dem Warteschlangenmanager befinden, zu dem der Zielagent eine Verbindung herstellt, die beiden zugehörigen Warteschlangenmanager müssen jedoch miteinander kommunizieren können.

Informationen zu diesem Vorgang

Die Quellendatei wird als `/tmp/single_record.txt` bezeichnet und befindet sich auf demselben System wie der Quellenagent `AGENT_NEPTUNE`. Der Quellenagent `AGENT_NEPTUNE` verwendet den Warteschlangenmanager `QM_NEPTUNE`. Der Zielagent heißt `AGENT_VENUS` und stellt eine Verbindung zum Warteschlangenmanager `QM_VENUS` her. Die Zielwarteschlange `RECEIVING_QUEUE` befindet sich

auf dem Warteschlangenmanager QM_MERCURY. QM_MERCURY befindet sich im gleichen IBM MQ-Netz wie der Warteschlangenmanager QM_VENUS, der darauf zugreifen kann.

Vorgehensweise

Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -dm QM_VENUS  
-dq RECEIVING_QUEUE@QM_MERCURY /tmp/single_record.txt
```

Wenn sich die Zielwarteschlange auf einem anderen Warteschlangenmanager befindet als dem Warteschlangenmanager, der vom Zielagenten verwendet wird, müssen Sie den Wert des Parameters **-dq** im Format *Warteschlangennamen@Warteschlangenmanagername* angeben. Wenn dieser Wert keine Angabe für *@Warteschlangenmanagername* enthält, geht der Zielagent davon aus, dass sich die Zielwarteschlange im eigenen Warteschlangenmanager (im Warteschlangenmanager des Zielagenten) befindet. Die Ausnahmebedingung ist, wenn die Agenteneigenschaft 'enableClusterQueueInputOutput' auf 'true' gesetzt wurde. In diesem Fall verwendet der Zielagent Standardprozeduren für die IBM MQ, um festzustellen, wo sich die Warteschlange befindet.

Der Quellenagent AGENT_NEPTUNE liest die Daten aus der Datei /tmp/single_record.txt und überträgt diese Daten an den Zielagenten, AGENT_VENUS. Der Zielagent AGENT_VENUS sendet die Daten an eine persistente Nachricht in der Warteschlange RECEIVING_QUEUE@QM_MERCURY. Für die Nachricht ist keine IBM MQ-Gruppen-ID festgelegt.

Zugehörige Konzepte

[„Daten von Dateien in Nachrichten übertragen“](#) auf Seite 276

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

Zugehörige Tasks

[„Agenten zum Ausführen von Datei-zu-Nachricht-Übertragungen konfigurieren“](#) auf Seite 278

Standardmäßig können Agenten keine Datei-zu-Nachricht- oder Nachricht-zu-Datei-Übertragungen ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft 'enableQueueInputOutput' auf den Wert 'true' setzen. Damit ein Schreibzugriff auf IBM MQ-Clusterwarteschlangen möglich ist, müssen Sie die Agenteneigenschaft 'enableClusterQueueInputOutput' ebenfalls auf 'true' setzen.

[„Beispiel: Einzelne Datei in mehrere Nachrichten nach Länge aufteilen“](#) auf Seite 281

Sie können eine Datei mit dem Parameter **-qs** des Befehls **fteCreateTransfer** in mehrere IBM MQ-Nachrichten aufteilen. Die Datei wird in Abschnitte mit fester Länge aufgeteilt, wobei jeder Abschnitt in eine einzelne Nachricht geschrieben wird.

[„Beispiel: Eine Textdatei mit einem regulären Ausdruckbegrenzer trennen und den Begrenzer in den Nachrichten eingrenzen“](#) auf Seite 284

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen und den über den regulären Ausdruck erhaltenen Treffer in die Nachrichten einfügen. Dazu verwenden Sie die Parameter **-dqdt** und **-qi** des Befehls **fteCreateTransfer**.

[„Beispiel: Aufteilen einer Textdatei in mehrere Nachrichten mithilfe eines regulären Ausdrucks als Begrenzer“](#) auf Seite 282

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen. Verwenden Sie hierfür den Parameter **-dqdt** des Befehls **fteCreateTransfer**.

[„Beispiel: Einstellen der IBM MQ-Nachrichteneigenschaften bei einer Datei-zu-Nachricht-Übertragung“](#) auf Seite 287

Sie können mit dem Parameter **-qmp** beim Befehl **fteCreateTransfer** angeben, ob IBM MQ-Nachrichteneigenschaften bei der ersten Nachricht festgelegt werden, die von der Übertragung in die Zielwarteschlange geschrieben wird. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Beispiel: Einstellen von benutzerdefinierten Eigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 288

Benutzerdefinierte Metadaten werden bei der ersten Nachricht, die von der Übertragung in die Zielwarteschlange geschrieben wird, als IBM MQ-Nachrichteneigenschaften festgelegt. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Neue Dateiübertragung starten“ auf Seite 228

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

Zugehörige Verweise

„Fehler bei Übertragung von Datei-zu-Nachrichten“ auf Seite 291

Wenn eine Übertragung von Dateien zu Nachrichten fehlschlägt, nachdem der Agent mit dem Schreiben von Dateidaten in die Zielwarteschlange begonnen hat, schreibt der Agent eine Nachricht in die Warteschlange, um auf eine Anwendung hinzuweisen, die die Nachrichten konsumiert, die ein Fehler aufgetreten ist.

Beispiel: Einzelne Datei in mehrere Nachrichten nach Länge aufteilen

Sie können eine Datei mit dem Parameter **-qs** des Befehls **fteCreateTransfer** in mehrere IBM MQ-Nachrichten aufteilen. Die Datei wird in Abschnitte mit fester Länge aufgeteilt, wobei jeder Abschnitt in eine einzelne Nachricht geschrieben wird.

Informationen zu diesem Vorgang

Die Quellendatei wird als `/tmp/source.file` bezeichnet und ist 36 KB groß. Die Quellendatei befindet sich im gleichen System wie der Quellenagent `AGENT_NEPTUNE`. Der Quellenagent `AGENT_NEPTUNE` stellt eine Verbindung zum Warteschlangenmanager `QM_NEPTUNE` her. Der Zielagent heißt `AGENT_MERCURY` und stellt eine Verbindung zum Warteschlangenmanager `QM_MERCURY` her. Die Zielwarteschlange `RECEIVING_QUEUE` befindet sich ebenfalls im Warteschlangenmanager `QM_MERCURY`. Durch die Übertragung wird die Quellendatei in Abschnitte mit einer Größe von 1 KB aufgeteilt, und jeder dieser Abschnitte wird in eine Nachricht in der Zielwarteschlange `RECEIVING_QUEUE` geschrieben.

Vorgehensweise

Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -qs 1K /tmp/source.file
```

Der Quellenagent `AGENT_NEPTUNE` liest die Daten aus der Datei `/tmp/source.file` und überträgt diese Daten an den Zielagenten `AGENT_MERCURY`. Der Zielagent `AGENT_MERCURY` schreibt die Daten in 36 persistente Nachrichten mit einer Größe von 1 KB in der Warteschlange `RECEIVING_QUEUE@QM_MERCURY`. Diese Nachrichten haben alle die gleiche IBM MQ-Gruppen-ID und für die letzte Nachricht der Gruppe ist das IBM MQ-Flag `LAST_MSG_IN_GROUP` gesetzt.

Zugehörige Konzepte

„Daten von Dateien in Nachrichten übertragen“ auf Seite 276

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

Zugehörige Tasks

„Agenten zum Ausführen von Datei-zu-Nachricht-Übertragungen konfigurieren“ auf Seite 278

Standardmäßig können Agenten keine Datei-zu-Nachricht- oder Nachricht-zu-Datei-Übertragungen ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft `'enableQueueInputOutput'` auf den Wert `'true'` setzen. Damit ein Schreibzugriff auf IBM MQ-Clusterwarteschlangen möglich ist, müssen Sie die Agenteneigenschaft `'enableClusterQueueInputOutput'` ebenfalls auf `'true'` setzen.

„Beispiel: Übertragen einer einzelnen Datei zu einer einzelnen Nachricht“ auf Seite 279

Sie können eine Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-dq** beim Befehl **fteCreateTransfer** verwenden. Die Quelldatei muss kleiner sein als die maximale Nachrichtenlänge, die bei der Zielwarteschlange eingestellt ist. Die Zielwarteschlange muss sich nicht unbedingt auf dem Warteschlangenmanager befinden, zu dem der Zielagent eine Verbindung herstellt, die beiden zugehörigen Warteschlangenmanager müssen jedoch miteinander kommunizieren können.

„Beispiel: Eine Textdatei mit einem regulären Ausdruckbegrenzer trennen und den Begrenzer in den Nachrichten eingrenzen“ auf Seite 284

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen und den über den regulären Ausdruck erhaltenen Treffer in die Nachrichten einfügen. Dazu verwenden Sie die Parameter **-dqdt** und **-qi** des Befehls **fteCreateTransfer**.

„Beispiel: Aufteilen einer Textdatei in mehrere Nachrichten mithilfe eines regulären Ausdrucks als Begrenzer“ auf Seite 282

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen. Verwenden Sie hierfür den Parameter **-dqdt** des Befehls **fteCreateTransfer**.

„Beispiel: Einstellen der IBM MQ-Nachrichteneigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 287

Sie können mit dem Parameter **-qmp** beim Befehl **fteCreateTransfer** angeben, ob IBM MQ-Nachrichteneigenschaften bei der ersten Nachricht festgelegt werden, die von der Übertragung in die Zielwarteschlange geschrieben wird. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Beispiel: Einstellen von benutzerdefinierten Eigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 288

Benutzerdefinierte Metadaten werden bei der ersten Nachricht, die von der Übertragung in die Zielwarteschlange geschrieben wird, als IBM MQ-Nachrichteneigenschaften festgelegt. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Neue Dateiübertragung starten“ auf Seite 228

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

Zugehörige Verweise

„Fehler bei Übertragung von Datei-zu-Nachrichten“ auf Seite 291

Wenn eine Übertragung von Dateien zu Nachrichten fehlschlägt, nachdem der Agent mit dem Schreiben von Dateidaten in die Zielwarteschlange begonnen hat, schreibt der Agent eine Nachricht in die Warteschlange, um auf eine Anwendung hinzuweisen, die die Nachrichten konsumiert, die ein Fehler aufgetreten ist.

Beispiel: Aufteilen einer Textdatei in mehrere Nachrichten mithilfe eines regulären Ausdrucks als Begrenzer

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen. Verwenden Sie hierfür den Parameter **-dqdt** des Befehls **fteCreateTransfer**.

Informationen zu diesem Vorgang

Die Datei wird in Abschnitte unterschiedlicher Länge aufgeteilt, von denen jeder in eine einzelne Nachricht geschrieben wird. Die Textdatei wird an jedem Punkt aufgeteilt, an dem der Text in der Datei einem

vorgegebenen regulären Ausdruck entspricht. Die Quellendatei wird als /tmp/names.text bezeichnet und hat den folgenden Inhalt:

```
Jenny Jones,John Smith,Jane Brown
```

Der reguläre Ausdruck, der angibt, wo die Datei aufgeteilt werden soll, ist das Kommazeichen (,).

Die Quellendatei befindet sich auf dem gleichen System wie der Quellenagent *AGENT_NEPTUNE*, der sich mit dem Warteschlangenmanager *QM_NEPTUNE* verbindet. Die Zielwarteschlange *RECEIVING_QUEUE* befindet sich auf dem Warteschlangenmanager *QM_MERCURY*. *QM_MERCURY* ist ebenfalls der Warteschlangenmanager, der vom Zielagent *AGENT_MERCURY* verwendet wird. Die Übertragung teilt die Quellendatei in Abschnitte auf und schreibt jeden davon in eine Nachricht bei *RECEIVING_QUEUE*.

Vorgehensweise

Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -t text -dqdp postfix -dqdt "," /tmp/names.text
```

Der Quellenagent *AGENT_NEPTUNE* liest die Daten aus der Datei /tmp/names.text und überträgt diese Daten an den Zielagenten *AGENT_MERCURY*. Der Zielagent *AGENT_MERCURY* schreibt die Daten in drei persistente Nachrichten in der Warteschlange *RECEIVING_QUEUE*. Diese Nachrichten haben alle die gleiche IBM MQ-Gruppen-ID und für die letzte Nachricht der Gruppe ist das IBM MQ-Flag *LAST_MSG_IN_GROUP* gesetzt.

Die Daten in den Nachrichten sind wie folgt.

- Erste Nachricht:

```
Jenny Jones
```

- Zweite Nachricht:

```
John Smith
```

- Dritte Nachricht:

```
Jane Brown
```

Zugehörige Konzepte

„Daten von Dateien in Nachrichten übertragen“ auf Seite 276

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

Zugehörige Tasks

„Agenten zum Ausführen von Datei-zu-Nachricht-Übertragungen konfigurieren“ auf Seite 278

Standardmäßig können Agenten keine Datei-zu-Nachricht- oder Nachricht-zu-Datei-Übertragungen ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft 'enableQueueInputOutput' auf den Wert 'true' setzen. Damit ein Schreibzugriff auf IBM MQ-Clusterwarteschlangen möglich ist, müssen Sie die Agenteneigenschaft 'enableClusterQueueInputOutput' ebenfalls auf 'true' setzen.

„Beispiel: Übertragen einer einzelnen Datei zu einer einzelnen Nachricht“ auf Seite 279

Sie können eine Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-dq** beim Befehl **fteCreateTransfer** verwenden. Die Quellendatei muss kleiner sein als die maximale Nachrichtenlänge, die bei der Zielwarteschlange eingestellt ist. Die Zielwarteschlange muss sich nicht unbedingt auf dem Warteschlangenmanager befinden, zu dem der Zielagent eine Verbindung herstellt, die beiden zugehörigen Warteschlangenmanager müssen jedoch miteinander kommunizieren können.

„Beispiel: Einzelne Datei in mehrere Nachrichten nach Länge aufteilen“ auf Seite 281

Sie können eine Datei mit dem Parameter **-qs** des Befehls **fteCreateTransfer** in mehrere IBM MQ-Nachrichten aufteilen. Die Datei wird in Abschnitte mit fester Länge aufgeteilt, wobei jeder Abschnitt in eine einzelne Nachricht geschrieben wird.

„Beispiel: Eine Textdatei mit einem regulären Ausdruckbegrenzer trennen und den Begrenzer in den Nachrichten eingrenzen“ auf Seite 284

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen und den über den regulären Ausdruck erhaltenen Treffer in die Nachrichten einfügen. Dazu verwenden Sie die Parameter **-dqdt** und **-qi** des Befehls **fteCreateTransfer**.

„Beispiel: Einstellen der IBM MQ-Nachrichteneigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 287

Sie können mit dem Parameter **-qmp** beim Befehl **fteCreateTransfer** angeben, ob IBM MQ-Nachrichteneigenschaften bei der ersten Nachricht festgelegt werden, die von der Übertragung in die Zielwarteschlange geschrieben wird. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Beispiel: Einstellen von benutzerdefinierten Eigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 288

Benutzerdefinierte Metadaten werden bei der ersten Nachricht, die von der Übertragung in die Zielwarteschlange geschrieben wird, als IBM MQ-Nachrichteneigenschaften festgelegt. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Neue Dateiübertragung starten“ auf Seite 228

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

Zugehörige Verweise

„Fehler bei Übertragung von Datei-zu-Nachrichten“ auf Seite 291

Wenn eine Übertragung von Dateien zu Nachrichten fehlschlägt, nachdem der Agent mit dem Schreiben von Dateidaten in die Zielwarteschlange begonnen hat, schreibt der Agent eine Nachricht in die Warteschlange, um auf eine Anwendung hinzuweisen, die die Nachrichten konsumiert, die ein Fehler aufgetreten ist.

Von MFT verwendete reguläre Ausdrücke

Beispiel: Eine Textdatei mit einem regulären Ausdruckbegrenzer trennen und den Begrenzer in den Nachrichten eingrenzen

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen und den über den regulären Ausdruck erhaltenen Treffer in die Nachrichten einfügen. Dazu verwenden Sie die Parameter **-dqdt** und **-qi** des Befehls **fteCreateTransfer**.

Informationen zu diesem Vorgang

Übertragen Sie eine einzelne Textdatei in mehrere Nachrichten in einer Warteschlange. Die Datei wird in Abschnitte unterschiedlicher Länge aufgeteilt, von denen jeder in eine einzelne Nachricht geschrieben wird. Die Textdatei wird an jedem Punkt aufgeteilt, an dem der Text in der Datei einem vorgegebenen regulären Ausdruck entspricht. Die Quellendatei wird als `/tmp/customers.text` bezeichnet und hat den folgenden Inhalt:

```
Customer name: John Smith
Customer contact details: john@example.net
Customer number: 314
```

```
Customer name: Jane Brown
```

```
Customer contact details: jane@example.com
Customer number: 42
```

```
Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

Der reguläre Ausdruck, der angibt, wo die Datei geteilt werden soll `Customer\snumber:\s\d+`, die mit dem Text "Kundennummer:" gefolgt von einer beliebigen Anzahl von Ziffern übereinstimmt. Reguläre Ausdrücke, die in der Befehlszeile angegeben werden, müssen in doppelte Anführungszeichen eingeschlossen werden, um zu verhindern, dass die Befehlshell den regulären Ausdruck auswertet. Der reguläre Ausdruck wird als regulärer Java-Ausdruck ausgewertet. Weitere Informationen finden Sie im Abschnitt [Von MFT verwendete reguläre Ausdrücke](#).

Standardmäßig ist die Anzahl der Zeichen, mit denen ein regulärer Ausdruck übereinstimmen kann, auf fünf festgelegt. Der reguläre Ausdruck, der in diesem Beispiel verwendet wird, stimmt mit Zeichenfolgen überein, die länger als fünf Zeichen sind. Um Übereinstimmungen zu aktivieren, die länger als fünf Zeichen sind, bearbeiten Sie die Agenteneigenschaftendatei so, dass sie die Eigenschaft **maxDelimiterMatchLength** enthält.

Standardmäßig ist der Text, der mit dem regulären Ausdruck übereinstimmt, nicht in den Nachrichten enthalten. Verwenden Sie den Parameter **-qi**, um den Text einzuschließen, der dem regulären Ausdruck in den Nachrichten entspricht, wie in diesem Beispiel. Die Quelldatei befindet sich auf demselben System wie der Quellenagent AGENT_NEPTUNE, der eine Verbindung zum WS-Manager QM_NEPTUNE herstellt. Die Zielwarteschlange RECEIVING_QUEUE befindet sich auf dem Warteschlangenmanager QM_MERCURY. QM_MERCURY ist auch der Warteschlangenmanager, der vom Zielagenten AGENT_MERCURY verwendet wird. Bei der Übertragung wird die Quelldatei in Abschnitte aufgeteilt und jede dieser Abschnitte in eine Nachricht in RECEIVING_QUEUE geschrieben.

Vorgehensweise

1. Stoppen Sie den Zielagenten mit dem folgenden Befehl:

```
fteStopAgent AGENT_MERCURY
```

2. Fügen Sie die folgende Zeile zur Agenteneigenschaftendatei für AGENT_MERCURY hinzu:

```
maxDelimiterMatchLength=25
```

Anmerkung: Wenn Sie den Wert von **maxDelimiterMatchLength** erhöhen, kann die Leistung verringert werden.

3. Starten Sie den Zielagenten mit dem folgenden Befehl:

```
fteStartAgent AGENT_MERCURY
```

4. Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY
-dq RECEIVING_QUEUE
text -dqdt "Customer\snumber:\s\d+" -qi -dqdp postfix /tmp/customers.text
```

Der Quellenagent AGENT_NEPTUNE liest die Daten aus der Datei `/tmp/customers.text` und überträgt diese Daten an den Zielagenten AGENT_MERCURY. Der Zielagent (AGENT_MERCURY) schreibt die Daten in drei persistente Nachrichten in der Warteschlange RECEIVING_QUEUE. Diese Nachrichten haben alle die gleiche IBM MQ-Gruppen-ID und für die letzte Nachricht der Gruppe ist das IBM MQ-Flag `LAST_MSG_IN_GROUP` gesetzt.

Die Daten in den Nachrichten sind wie folgt.

- Erste Nachricht:

```
Customer name: John Smith
Customer contact details: john@example.net
Customer number: 314
```

- Zweite Nachricht:

```
Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42
```

- Dritte Nachricht:

```
Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

Zugehörige Konzepte

[„Daten von Dateien in Nachrichten übertragen“ auf Seite 276](#)

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

Zugehörige Tasks

[„Agenten zum Ausführen von Datei-zu-Nachricht-Übertragungen konfigurieren“ auf Seite 278](#)

Standardmäßig können Agenten keine Datei-zu-Nachricht- oder Nachricht-zu-Datei-Übertragungen ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft 'enableQueueInputOutput' auf den Wert 'true' setzen. Damit ein Schreibzugriff auf IBM MQ-Clusterwarteschlangen möglich ist, müssen Sie die Agenteneigenschaft 'enableClusterQueueInputOutput' ebenfalls auf 'true' setzen.

[„Beispiel: Übertragen einer einzelnen Datei zu einer einzelnen Nachricht“ auf Seite 279](#)

Sie können eine Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-dq** beim Befehl **fteCreateTransfer** verwenden. Die Quelldatei muss kleiner sein als die maximale Nachrichtenlänge, die bei der Zielwarteschlange eingestellt ist. Die Zielwarteschlange muss sich nicht unbedingt auf dem Warteschlangenmanager befinden, zu dem der Zielagent eine Verbindung herstellt, die beiden zugehörigen Warteschlangenmanager müssen jedoch miteinander kommunizieren können.

[„Beispiel: Einzelne Datei in mehrere Nachrichten nach Länge aufteilen“ auf Seite 281](#)

Sie können eine Datei mit dem Parameter **-qs** des Befehls **fteCreateTransfer** in mehrere IBM MQ-Nachrichten aufteilen. Die Datei wird in Abschnitte mit fester Länge aufgeteilt, wobei jeder Abschnitt in eine einzelne Nachricht geschrieben wird.

[„Beispiel: Aufteilen einer Textdatei in mehrere Nachrichten mithilfe eines regulären Ausdrucks als Begrenzer“ auf Seite 282](#)

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen. Verwenden Sie hierfür den Parameter **-dqdt** des Befehls **fteCreateTransfer**.

[„Beispiel: Einstellen der IBM MQ-Nachrichteneigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 287](#)

Sie können mit dem Parameter **-qmp** beim Befehl **fteCreateTransfer** angeben, ob IBM MQ-Nachrichteneigenschaften bei der ersten Nachricht festgelegt werden, die von der Übertragung in die Zielwarteschlange geschrieben wird. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

[„Beispiel: Einstellen von benutzerdefinierten Eigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 288](#)

Benutzerdefinierte Metadaten werden bei der ersten Nachricht, die von der Übertragung in die Zielwarteschlange geschrieben wird, als IBM MQ-Nachrichteneigenschaften festgelegt. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Neue Dateiübertragung starten“ auf Seite 228

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

Zugehörige Verweise

Die MFT `agent.properties`-Datei

Von MFT verwendete reguläre Ausdrücke

Beispiel: Einstellen der IBM MQ-Nachrichteneigenschaften bei einer Datei-zu-Nachricht-Übertragung

Sie können mit dem Parameter `-qmp` beim Befehl `fteCreateTransfer` angeben, ob IBM MQ-Nachrichteneigenschaften bei der ersten Nachricht festgelegt werden, die von der Übertragung in die Zielwarteschlange geschrieben wird. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

Informationen zu diesem Vorgang

Schließen Sie den Parameter `-qmp true` im Befehl `fteCreateTransfer` ein. In diesem Beispiel lautet die MQMD-Benutzer-ID des Benutzers, der den Befehl übergibt, `larmer`.

Vorgehensweise

Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM -qmp true
-t text /tmp/source_file.txt
```

Die IBM MQ-Nachrichteneigenschaften der ersten Nachricht, die vom Zielagenten `AGENT_SATURN` in die Warteschlange `MY_QUEUE` beim Warteschlangenmanager `MyQM` gestellt werden, werden auf die folgenden Werte gesetzt:

```
usr.WMQFTETransferId=414cbaedefa234889d999a8ed09782395ea213ebbc9377cd
usr.WMQFTETransferMode=text
usr.WMQFTESourceAgent=AGENT_JUPITER
usr.WMQFTEDestinationAgent=AGENT_SATURN
usr.WMQFTEFileName=source_file.txt
usr.WMQFTEFileSize=1024
usr.WMQFTEFileLastModified=1273740879040
usr.WMQFTEFileIndex=0
usr.WMQFTEmqmdUser=larmer
```

Zugehörige Konzepte

„Daten von Dateien in Nachrichten übertragen“ auf Seite 276

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

Zugehörige Tasks

„Agenten zum Ausführen von Datei-zu-Nachricht-Übertragungen konfigurieren“ auf Seite 278

Standardmäßig können Agenten keine Datei-zu-Nachricht- oder Nachricht-zu-Datei-Übertragungen ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft `'enableQueueInputOutput'` auf den Wert `'true'` setzen. Damit ein Schreibzugriff auf IBM MQ-Clusterwarteschlangen möglich ist, müssen Sie die Agenteneigenschaft `'enableClusterQueueInputOutput'` ebenfalls auf `'true'` setzen.

„Beispiel: Übertragen einer einzelnen Datei zu einer einzelnen Nachricht“ auf Seite 279

Sie können eine Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter `-dq` beim Befehl `fteCreateTransfer` verwenden. Die Quelldatei muss kleiner sein als die maximale Nachrichtenlänge, die bei der Zielwarteschlange eingestellt ist. Die Zielwarteschlange muss sich nicht

unbedingt auf dem Warteschlangenmanager befinden, zu dem der Zielagent eine Verbindung herstellt, die beiden zugehörigen Warteschlangenmanager müssen jedoch miteinander kommunizieren können.

„Beispiel: Einzelne Datei in mehrere Nachrichten nach Länge aufteilen“ auf Seite 281

Sie können eine Datei mit dem Parameter **-qs** des Befehls **fteCreateTransfer** in mehrere IBM MQ-Nachrichten aufteilen. Die Datei wird in Abschnitte mit fester Länge aufgeteilt, wobei jeder Abschnitt in eine einzelne Nachricht geschrieben wird.

„Beispiel: Eine Textdatei mit einem regulären Ausdruckbegrenzer trennen und den Begrenzer in den Nachrichten eingrenzen“ auf Seite 284

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen und den über den regulären Ausdruck erhaltenen Treffer in die Nachrichten einfügen. Dazu verwenden Sie die Parameter **-dqdt** und **-qi** des Befehls **fteCreateTransfer**.

„Beispiel: Aufteilen einer Textdatei in mehrere Nachrichten mithilfe eines regulären Ausdrucks als Begrenzer“ auf Seite 282

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen. Verwenden Sie hierfür den Parameter **-dqdt** des Befehls **fteCreateTransfer**.

„Beispiel: Einstellen von benutzerdefinierten Eigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 288

Benutzerdefinierte Metadaten werden bei der ersten Nachricht, die von der Übertragung in die Zielwarteschlange geschrieben wird, als IBM MQ-Nachrichteneigenschaften festgelegt. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

„Neue Dateiübertragung starten“ auf Seite 228

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

Zugehörige Verweise

„Fehler bei Übertragung von Datei-zu-Nachrichten“ auf Seite 291

Wenn eine Übertragung von Dateien zu Nachrichten fehlschlägt, nachdem der Agent mit dem Schreiben von Dateidaten in die Zielwarteschlange begonnen hat, schreibt der Agent eine Nachricht in die Warteschlange, um auf eine Anwendung hinzuweisen, die die Nachrichten konsumiert, die ein Fehler aufgetreten ist.

MQ-Nachrichteneigenschaften, die von MFT für Nachrichten festgelegt werden, die in Zielwarteschlangen geschrieben werden

Beispiel: Einstellen von benutzerdefinierten Eigenschaften bei einer Datei-zu-Nachricht-Übertragung

Benutzerdefinierte Metadaten werden bei der ersten Nachricht, die von der Übertragung in die Zielwarteschlange geschrieben wird, als IBM MQ-Nachrichteneigenschaften festgelegt. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

Informationen zu diesem Vorgang

Schließen Sie die Parameter **-qmp true** und **-md account=123456** in den Befehl **fteCreateTransfer** ein, um die Eigenschaft **usr.account** im Header RFH2 auf 123456 zu setzen.

Vorgehensweise

Geben Sie den folgenden Befehl ein:


```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM
-qmp true -md account=123456 /tmp/source_file.txt
```

Zusätzlich zu den IBM MQ-Standardnachrichteneigenschaften wird die benutzerdefinierte Eigenschaft im Nachrichtenheader der ersten vom Zielagenten AGENT_SATURN in die Warteschlange MY_QUEUE beim Warteschlangenmanager MyQM gestellte Nachricht festgelegt. Der Header wird auf den folgenden Wert festgelegt:

```
usr.account=123456
```

Das Präfix `usr` wird der Bezeichnung der benutzerdefinierten Metadaten vorangestellt.

Zugehörige Konzepte

[„Daten von Dateien in Nachrichten übertragen“ auf Seite 276](#)

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

Zugehörige Tasks

[„Agenten zum Ausführen von Datei-zu-Nachricht-Übertragungen konfigurieren“ auf Seite 278](#)

Standardmäßig können Agenten keine Datei-zu-Nachricht- oder Nachricht-zu-Datei-Übertragungen ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft 'enableQueueInputOutput' auf den Wert 'true' setzen. Damit ein Schreibzugriff auf IBM MQ-Clusterwarteschlangen möglich ist, müssen Sie die Agenteneigenschaft 'enableClusterQueueInputOutput' ebenfalls auf 'true' setzen.

[„Beispiel: Übertragen einer einzelnen Datei zu einer einzelnen Nachricht“ auf Seite 279](#)

Sie können eine Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-dq** beim Befehl **fteCreateTransfer** verwenden. Die Quelldatei muss kleiner sein als die maximale Nachrichtenlänge, die bei der Zielwarteschlange eingestellt ist. Die Zielwarteschlange muss sich nicht unbedingt auf dem Warteschlangenmanager befinden, zu dem der Zielagent eine Verbindung herstellt, die beiden zugehörigen Warteschlangenmanager müssen jedoch miteinander kommunizieren können.

[„Beispiel: Einzelne Datei in mehrere Nachrichten nach Länge aufteilen“ auf Seite 281](#)

Sie können eine Datei mit dem Parameter **-qs** des Befehls **fteCreateTransfer** in mehrere IBM MQ-Nachrichten aufteilen. Die Datei wird in Abschnitte mit fester Länge aufgeteilt, wobei jeder Abschnitt in eine einzelne Nachricht geschrieben wird.

[„Beispiel: Eine Textdatei mit einem regulären Ausdruckbegrenzer trennen und den Begrenzer in den Nachrichten eingrenzen“ auf Seite 284](#)

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen und den über den regulären Ausdruck erhaltenen Treffer in die Nachrichten einfügen. Dazu verwenden Sie die Parameter **-dqdt** und **-qi** des Befehls **fteCreateTransfer**.

[„Beispiel: Aufteilen einer Textdatei in mehrere Nachrichten mithilfe eines regulären Ausdrucks als Begrenzer“ auf Seite 282](#)

Sie können eine Textdatei in mehrere Nachrichten übertragen, indem Sie die Datei bei jeder Übereinstimmung mit einem angegebenen regulären Java-Ausdruck aufteilen. Verwenden Sie hierfür den Parameter **-dqdt** des Befehls **fteCreateTransfer**.

[„Beispiel: Einstellen der IBM MQ-Nachrichteneigenschaften bei einer Datei-zu-Nachricht-Übertragung“ auf Seite 287](#)

Sie können mit dem Parameter **-qmp** beim Befehl **fteCreateTransfer** angeben, ob IBM MQ-Nachrichteneigenschaften bei der ersten Nachricht festgelegt werden, die von der Übertragung in die Zielwarteschlange geschrieben wird. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

[„Neue Dateiübertragung starten“ auf Seite 228](#)

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

Zugehörige Verweise

MQ-Nachrichteneigenschaften, die von MFT für Nachrichten festgelegt werden, die in Zielwarteschlangen geschrieben werden

Beispiel: Benutzerdefinierte Nachrichteneigenschaft für eine Datei-zu-Nachricht-Übertragung hinzufügen

Wenn Sie Managed File Transfer für verwaltete Datei-zu-Nachricht-Übertragungen verwenden, können Sie für die Ergebnismessage eine benutzerdefinierte Nachrichteneigenschaft hinzufügen.

Informationen zu diesem Vorgang

Zur Definition der benutzerdefinierten Nachrichteneigenschaft können Sie folgende Methoden verwenden:

- Geben Sie den Parameter **-md** in der Übertragungsanforderung an. Weitere Informationen finden Sie unter „[Beispiel: Einstellen von benutzerdefinierten Eigenschaften bei einer Datei-zu-Nachricht-Übertragung](#)“ auf Seite 288.
- Verwenden Sie die Ant-Task 'fte:filecopy' oder 'fte:filemove'. Nachfolgend sehen Sie ein Beispiel mit der Task 'fte:filecopy':

```
<project xmlns:fte="antlib.com.ibm.wmqfte.ant.taskdefs" default="complete">
<!-- Initialise the properties used in this script.-->

<target name="init" description="initialise task properties">
  <property name="src.file" value="/home/user/file1.bin"/>
  <property name="dst.queue" value="TEST.QUEUE@qm2"/>
  <fte:uuid property="job.name" length="8"
prefix="copyjob#"/>
</target>
<target name="step1" depends="init" description="transfer file">

<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result">

<fte:metadata>
<fte:entry name="fileName" value="{FileName}"/>
</fte:metadata>

<fte:filespec srcfilespec="{src.file}" dstqueue="{dst.queue}"
dstmsgprops="true"/>

</fte:filecopy>

</target>
</project>
```

- Verwenden Sie eine Ressourcenüberwachung mit Variablensubstitution. Das folgende Beispiel zeigt die XML einer Übertragungstask:

```
<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor
xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="5.00"
xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./Monitor.xsd">
  <name>METADATA</name>
  <pollInterval units="minutes">5</pollInterval>
  <batch maxSize="5"/>
  <agent>AGENT1</agent>
  <resources>
    <directory recursionLevel="0">e:\temp</directory>
  </resources>
  <triggerMatch>
```

```

<conditions>
  <allOf>
    <condition>
      <fileMatch>
        <pattern>*.txt</pattern>
      </fileMatch>
    </condition>
  </allOf>
</conditions>
</triggerMatch>
<tasks>
  <task>
    <name/>
    <transfer>
      <request version="5.00"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
        <managedTransfer>
          <originator>
            <hostName>mqjason.raleigh.ibm.com.</hostName>
            <userID>administrator</userID>
          </originator>
          <sourceAgent QMgr="AGENTQM" agent="AGENT1"/>
          <destinationAgent QMgr="AGENTQM" agent="AGENT2"/>
          <transferSet priority="0">
            <metaDataSet>
              <metaData key="FileName">${FileName}</metaData>
            </metaDataSet>
            <item checksumMethod="MD5" mode="text">
              <source disposition="delete" recursive="false">
                <file>${FilePath}</file>
              </source>
              <destination type="queue">
                <queue persistent="true"
setMqProps="true">TEST.QUEUE@AGENTQM</queue>
              </destination>
            </item>
          </transferSet>
        </job>
      </managedTransfer>
    </request>
  </transfer>
</task>
</tasks>
<originator>
  <hostName>mqjason.raleigh.ibm.com.</hostName>
  <userID>administrator</userID>
</originator>
</monitor:monitor>

```

Zugehörige Tasks

„[Beispiel: Einstellen der IBM MQ-Nachrichteneigenschaften bei einer Datei-zu-Nachricht-Übertragung](#)“ auf Seite 287

Sie können mit dem Parameter **-qmp** beim Befehl **fteCreateTransfer** angeben, ob IBM MQ-Nachrichteneigenschaften bei der ersten Nachricht festgelegt werden, die von der Übertragung in die Zielwarteschlange geschrieben wird. Mit IBM MQ-Nachrichteneigenschaften kann eine Anwendung Nachrichten für die Verarbeitung auswählen oder Informationen zu einer Nachricht abrufen, ohne auf IBM MQ Message Descriptor-Header (MQMD) oder MQRFH2-Header zugreifen zu müssen.

Zugehörige Verweise

[Die Ant-Task 'fte:filecopy'](#)

[Die Ant-Task 'fte:filemove'](#)

Fehler bei Übertragung von Datei-zu-Nachrichten

Wenn eine Übertragung von Dateien zu Nachrichten fehlschlägt, nachdem der Agent mit dem Schreiben von Dateidaten in die Zielwarteschlange begonnen hat, schreibt der Agent eine Nachricht in die Warteschlange, um auf eine Anwendung hinzuweisen, die die Nachrichten konsumiert, die ein Fehler aufgetreten ist.

Die Nachricht, die in die Zielwarteschlange geschrieben wird, wenn ein Fehler auftritt:

- Ist leer
- Hat die gleiche IBM MQ-Gruppen-ID wie die vorherige Nachricht, die vom Agenten in die Zielwarteschlange geschrieben wurde
- Hat das IBM MQ-Flag-Set LAST_MSG_IN_GROUP
- Enthält zusätzliche IBM MQ-Nachrichteneigenschaften, wenn Nachrichteneigenschaften aktiviert sind. Weitere Informationen finden Sie im Artikel [MQ-Nachrichteneigenschaften, die von MFT für Nachrichten festgelegt werden, die in Zielwarteschlangen geschrieben werden](#).

Beispiel

Eine Übertragung wird angefordert, indem der folgende Befehl ausgeführt wird:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE
-qmp true -qs 1K /tmp/source1.txt
```

Die Datei `source1.txt` ist 48 KB. Die Übertragung teilt diese Datei in 1-KB-Nachrichten auf und schreibt diese Nachrichten in die Zielwarteschlange `RECEIVING_QUEUE`.

Während die Übertragung in Bearbeitung ist, tritt ein Fehler auf dem Quellenagenten auf, nachdem der Agent 16 Nachrichten in `RECEIVING_QUEUE` geschrieben hat.

Der Agent schreibt eine leere Nachricht in `RECEIVING_QUEUE`. Neben der Standardgruppe von Nachrichteneigenschaften verfügt die leere Nachricht über die folgenden Nachrichteneigenschaftengruppe:

```
usr.WMQFTEResultCode = 40
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

Ab IBM MQ 9.3.0 wird nur eine leere Nachricht gesendet, wenn eine Übertragung aus einer Datei aufgrund eines Fehlers bei der Überprüfung der Begrenzergröße fehlschlägt. Außerdem werden die Benutzereigenschaften zu dieser Nachricht hinzugefügt, wenn der Übertragungsfehler auf das Begrenzungszeichen zurückzuführen ist, das die festgelegte Größe auf dem Zielagenten überschreitet.

Zugehörige Konzepte

„[Daten von Dateien in Nachrichten übertragen](#)“ auf Seite 276

Mithilfe der entsprechenden Managed File Transfer-Funktion können Sie Daten aus einer Datei in eine Nachricht oder auch in mehrere Nachrichten in einer IBM MQ-Warteschlange übertragen.

Zugehörige Tasks

„[Agenten zum Ausführen von Datei-zu-Nachricht-Übertragungen konfigurieren](#)“ auf Seite 278

Standardmäßig können Agenten keine Datei-zu-Nachricht- oder Nachricht-zu-Datei-Übertragungen ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft 'enableQueueInputOutput' auf den Wert 'true' setzen. Damit ein Schreibzugriff auf IBM MQ-Clusterwarteschlangen möglich ist, müssen Sie die Agenteneigenschaft 'enableClusterQueueInputOutput' ebenfalls auf 'true' setzen.

„[Neue Dateiübertragung starten](#)“ auf Seite 228

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

Zugehörige Verweise

Die MFT `agent.properties`-Datei

[MQ-Nachrichteneigenschaften, die von MFT für Nachrichten festgelegt werden, die in Zielwarteschlangen geschrieben werden](#)

Daten von Nachrichten in Dateien übertragen

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nach-

richten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

Weitere Informationen zu Datei-Nachricht-Übertragungen finden Sie im Abschnitt [„Daten von Dateien in Nachrichten übertragen“](#) auf Seite 276.



Achtung: Als Quellenagent für eine Nachricht-Datei-Übertragung kommt weder ein Protokollbridgeagent noch ein Connect:Direct-Bridgeagent in Frage.

Sie können IBM MQ-Nachrichtendaten an eine Datei übertragen. Die folgenden Typen von Nachrichten-in-Datei-Übertragung werden unterstützt:

- Von einer einzelnen Nachricht in eine einzelne Datei
- Von mehreren Nachrichten in eine einzelne Datei
- Aus mehreren Nachrichten mit der gleichen IBM MQ-Gruppen-ID an eine einzelne Datei
- Von mehreren Nachrichten in eine einzelne Datei, einschließlich eines Textes oder eines binären Begrenzers zwischen den Daten der einzelnen Nachrichten, die in die Datei geschrieben werden.

Bei der Übertragung von Dateien aus großen oder vielen kleinen Nachrichten müssen Sie unter Umständen einige der IBM MQ- oder Managed File Transfer-Eigenschaften ändern. Weitere Informationen zu finden Sie unter [Anleitung zum Festlegen von MQ -Attributen und MFT-Eigenschaften, die der Nachrichtengröße zugeordnet sind](#).

Bei einer Nachricht-zu-Datei-Übertragung durchsucht der Quellenagent die Nachrichten aus der Quellenwarteschlange, im Gegensatz zum zerstörerischen GET in früheren Versionen von IBM MQ. Die Nachrichten werden aus der Quellenwarteschlange entfernt, nachdem alle Nachrichten (in einer Gruppe, wenn die Nachrichtengruppierung verwendet wird) durchsucht und Daten in die Zielfeile geschrieben wurden. Auf diese Weise können Nachrichten in der Quellenwarteschlange verbleiben, wenn eine Übertragung fehlschlägt oder abgebrochen wird. Aufgrund dieser Änderung muss die Berechtigung für BROWSE zusammen mit der GET-Berechtigung zum Ausführen von Nachrichten an Dateiübertragungen bereitgestellt werden.

Managed File Transfer vergleicht die Übertragungs-ID und den Wert des Attributs groupId in den XML-Nutzdaten der Übertragungsanforderung. Wenn diese beiden IDs äquivalent sind, verwendet der Quellenagent die Kennung als Übereinstimmungsoption für Nachrichten-ID (im Gegensatz zu einer Gruppenkennung-Abgleich) für den ersten MQGET-Versuch, der in der Eingabewarteschlange für die Übertragung von Nachrichten zu Datei erstellt wird.

Zugehörige Tasks

[„Beispiel: MFT-Ressource konfigurieren“](#) auf Seite 251

Sie können eine IBM MQ -Warteschlange als Ressource angeben, die von einer Ressourcenüberwachung überwacht werden soll, indem Sie den Parameter **-mq** mit dem Befehl **fteCreateMonitor** verwenden.

Zugehörige Verweise

[Von MFT gelesene MQ-Nachrichteneigenschaften aus Nachrichten in Quellenwarteschlangen](#)

[Anleitung zum Festlegen von MQ-Attributen und MFT-Eigenschaften, die der Nachrichtengröße zugeordnet sind](#)

Agent für die Ausführung der Übertragung einer Nachricht an eine Datei konfigurieren

Standardmäßig können Agenten keine Übertragung einer Nachricht an eine Datei oder einer Datei an eine Nachricht ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft enableQueueInputOutput auf den Wert 'true' setzen.

Informationen zu diesem Vorgang

Wenn Sie die Übertragung einer Nachricht an eine Datei von einem Quellenagenten aus ausführen möchten, für den die Eigenschaft enableQueueInputOutput nicht auf 'true' gesetzt ist, schlägt die Übertra-

gung fehl. Die im Koordinationswarteschlangenmanager veröffentlichte Übertragungsprotokollnachricht enthält folgende Nachricht:

```
BFGI00197E: An attempt to read from a queue was rejected by the source agent.  
The agent must have enableQueueInputOutput=true set in the agent.properties file  
to support transferring from a queue.
```

Gehen Sie wie folgt vor, um dem Agenten die Möglichkeit zu geben, in Warteschlangen zu schreiben und aus Warteschlangen zu lesen:

Vorgehensweise

1. Stoppen Sie den Quellenagenten mit dem Befehl **fteStopAgent**.
2. Bearbeiten Sie die Datei `agent.properties`, um die Zeile `enableQueueInputOutput=true` einzuschließen.

Die Datei `agent.properties` befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/source_agent_name`.

3. Starten Sie den Quellenagenten mit dem Befehl **fteStartAgent**.

Zugehörige Konzepte

„Daten von Nachrichten in Dateien übertragen“ auf Seite 292

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nachrichten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

Zugehörige Tasks

„Beispiel: Aus einer Warteschlange in eine einzelne Datei übertragen“ auf Seite 295

Sie können eine IBM MQ-Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-sq** mit dem Befehl **fteCreateTransfer** verwenden.

„Beispiel: Gruppe von Nachrichten aus einer Warteschlange in eine einzelne Datei übertragen“ auf Seite 296

Sie können eine einzelne vollständige Gruppe in einer IBM MQ-Warteschlange als Quelle einer Dateiübertragung angeben, indem Sie die Parameter **-sq** und **-sqgi** mit dem Befehl **fteCreateTransfer** verwenden.

„Beispiel: Textbegrenzer vor den Daten aus jeder Nachricht einfügen“ auf Seite 297

Bei der Übertragung im Textmodus aus einer Quellenwarteschlange in eine Datei können Sie angeben, dass vor den Daten aus einzelnen Nachrichten ein Textbegrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter **-sq**, **-sqdt** und **-sqdp** mit dem Befehl **fteCreateTransfer**.

„Beispiel: Binärer Begrenzer nach den Daten aus jeder Nachricht einfügen“ auf Seite 298

Bei der Übertragung im Binärmodus aus einer Quellenwarteschlange an eine Datei können Sie angeben, dass nach den Daten aus einzelnen Nachrichten ein binärer Begrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter **-sq**, **-sqdb** und **-sqdp** mit dem Befehl **fteCreateTransfer**.

„Warteschlange überwachen und Variablensubstitution verwenden“ auf Seite 258

Mit dem Befehl **fteCreateMonitor** können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

„Beispiel: Fehlschlagen einer Nachricht-zu-Datei-Übertragung mit IBM MQ-Nachrichteneigenschaften herbeiführen“ auf Seite 302

Sie können eine Nachricht-zu-Datei-Übertragung fehlschlagen lassen, indem Sie die Nachrichteneigenschaft `usr.UserReturnCode` IBM MQ auf einen Wert ungleich null setzen. Sie können auch zusätzliche Informationen zur Fehlerursache angeben, indem Sie die Nachrichteneigenschaft `usr.UserSupplement` IBM MQ festlegen.

Zugehörige Verweise

Die MFT `agent.properties`-Datei

Beispiel: Aus einer Warteschlange in eine einzelne Datei übertragen

Sie können eine IBM MQ-Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter `-sq` mit dem Befehl `fteCreateTransfer` verwenden.

Informationen zu diesem Vorgang

Die Quelldaten befinden sich in drei Nachrichten in der Warteschlange `START_QUEUE`. Diese Warteschlange muss sich im Warteschlangenmanager `QM_NEPTUNE` des Quellenagenten befinden.

Vorgehensweise

Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE
                 -da AGENT_VENUS -df /out/three_to_one.txt
                 -sq START_QUEUE
```

Die Daten in den Nachrichten an die Warteschlange `START_QUEUE` werden in die Datei `/out/three_to_one.txt` auf dem System geschrieben, auf dem `AGENT_VENUS` ausgeführt wird.

Zugehörige Konzepte

[„Daten von Nachrichten in Dateien übertragen“](#) auf Seite 292

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nachrichten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

Zugehörige Tasks

[„Agent für die Ausführung der Übertragung einer Nachricht an eine Datei konfigurieren“](#) auf Seite 293

Standardmäßig können Agenten keine Übertragung einer Nachricht an eine Datei oder einer Datei an eine Nachricht ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft `enableQueueInputOutput` auf den Wert `'true'` setzen.

[„Beispiel: Gruppe von Nachrichten aus einer Warteschlange in eine einzelne Datei übertragen“](#) auf Seite 296

Sie können eine einzelne vollständige Gruppe in einer IBM MQ-Warteschlange als Quelle einer Dateiübertragung angeben, indem Sie die Parameter `-sq` und `-sqgi` mit dem Befehl `fteCreateTransfer` verwenden.

[„Beispiel: Textbegrenzer vor den Daten aus jeder Nachricht einfügen“](#) auf Seite 297

Bei der Übertragung im Textmodus aus einer Quellenwarteschlange in eine Datei können Sie angeben, dass vor den Daten aus einzelnen Nachrichten ein Textbegrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter `-sq`, `-sqdt` und `-sqdp` mit dem Befehl `fteCreateTransfer`.

[„Beispiel: Binärer Begrenzer nach den Daten aus jeder Nachricht einfügen“](#) auf Seite 298

Bei der Übertragung im Binärmodus aus einer Quellenwarteschlange an eine Datei können Sie angeben, dass nach den Daten aus einzelnen Nachrichten ein binärer Begrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter `-sq`, `-sqdb` und `-sqdp` mit dem Befehl `fteCreateTransfer`.

[„Warteschlange überwachen und Variablensubstitution verwenden“](#) auf Seite 258

Mit dem Befehl `fteCreateMonitor` können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

„[Beispiel: Fehlschlagen einer Nachricht-zu-Datei-Übertragung mit IBM MQ-Nachrichteneigenschaften herbeiführen](#)“ auf Seite 302

Sie können eine Nachricht-zu-Datei-Übertragung fehlschlagen lassen, indem Sie die Nachrichteneigenschaft `usr.UserReturnCode` IBM MQ auf einen Wert ungleich null setzen. Sie können auch zusätzliche Informationen zur Fehlerursache angeben, indem Sie die Nachrichteneigenschaft `usr.UserSupplement` IBM MQ festlegen.

Zugehörige Verweise

Von MFT gelesene MQ-Nachrichteneigenschaften aus Nachrichten in Quellenwarteschlangen

fteCreateTransfer: Neue Dateiübertragung starten

Beispiel: Gruppe von Nachrichten aus einer Warteschlange in eine einzelne Datei übertragen

Sie können eine einzelne vollständige Gruppe in einer IBM MQ-Warteschlange als Quelle einer Dateiübertragung angeben, indem Sie die Parameter **-sq** und **-sqgi** mit dem Befehl **fteCreateTransfer** verwenden.

Informationen zu diesem Vorgang

In diesem Beispiel befinden sich in der Warteschlange `START_QUEUE` zehn Nachrichten. Diese Warteschlange muss sich im Warteschlangenmanager `QM_NEPTUNE` des Quellenagenten befinden. Die ersten drei Nachrichten gehören zu einer Gruppe mit der IBM MQ-Gruppen-ID `41424b3ef3a22020202020202020202020202020202020201111`; diese Gruppe ist keine vollständige Gruppe. Die nächsten fünf Nachrichten gehören zu einer Gruppe mit der IBM MQ-Gruppen-ID `41424b3ef3a2202020202020202020202020202020202020222`; diese Gruppe ist vollständig. Die übrigen beiden Nachrichten gehören zu einer Gruppe mit der IBM MQ-Gruppen-ID `41424b3ef3a2202020202020202020202020202020202020333`; auch diese Gruppe ist vollständig.

Vorgehensweise

Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS  
-df /out/group.txt -sqgi -sq START_QUEUE
```

Die Daten in den Nachrichten, die zu der ersten vollständigen Gruppe in der Warteschlange `START_QUEUE`, der Gruppe mit der IBM MQ-Gruppen-ID `41424b3ef3a2202020202020202020202020202020202020222` gehören, werden in die Datei `/out/group.txt` in dem System geschrieben, auf dem `AGENT_VENUS` ausgeführt wird.

Zugehörige Konzepte

„[Daten von Nachrichten in Dateien übertragen](#)“ auf Seite 292

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nachrichten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

Zugehörige Tasks

„[Agent für die Ausführung der Übertragung einer Nachricht an eine Datei konfigurieren](#)“ auf Seite 293
Standardmäßig können Agenten keine Übertragung einer Nachricht an eine Datei oder einer Datei an eine Nachricht ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft `enableQueueInputOutput` auf den Wert 'true' setzen.

„[Beispiel: Aus einer Warteschlange in eine einzelne Datei übertragen](#)“ auf Seite 295

Sie können eine IBM MQ-Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-sq** mit dem Befehl **fteCreateTransfer** verwenden.

„Beispiel: Textbegrenzer vor den Daten aus jeder Nachricht einfügen“ auf Seite 297

Bei der Übertragung im Textmodus aus einer Quellenwarteschlange in eine Datei können Sie angeben, dass vor den Daten aus einzelnen Nachrichten ein Textbegrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter **-sq**, **-sqdt** und **-sqdp** mit dem Befehl **fteCreateTransfer**.

„Beispiel: Binärer Begrenzer nach den Daten aus jeder Nachricht einfügen“ auf Seite 298

Bei der Übertragung im Binärmodus aus einer Quellenwarteschlange an eine Datei können Sie angeben, dass nach den Daten aus einzelnen Nachrichten ein binärer Begrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter **-sq**, **-sqdb** und **-sqdp** mit dem Befehl **fteCreateTransfer**.

„Warteschlange überwachen und Variablensubstitution verwenden“ auf Seite 258

Mit dem Befehl **fteCreateMonitor** können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

„Beispiel: Fehlschlagen einer Nachricht-zu-Datei-Übertragung mit IBM MQ-Nachrichteneigenschaften herbeiführen“ auf Seite 302

Sie können eine Nachricht-zu-Datei-Übertragung fehlschlagen lassen, indem Sie die Nachrichteneigenschaft `usr.UserReturnCode` IBM MQ auf einen Wert ungleich null setzen. Sie können auch zusätzliche Informationen zur Fehlerursache angeben, indem Sie die Nachrichteneigenschaft `usr.UserSupplement` IBM MQ festlegen.

Zugehörige Verweise

fteCreateTransfer: Neue Dateiübertragung starten

Beispiel: Textbegrenzer vor den Daten aus jeder Nachricht einfügen

Bei der Übertragung im Textmodus aus einer Quellenwarteschlange in eine Datei können Sie angeben, dass vor den Daten aus einzelnen Nachrichten ein Textbegrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter **-sq**, **-sqdt** und **-sqdp** mit dem Befehl **fteCreateTransfer**.

Informationen zu diesem Vorgang

In diesem Beispiel befinden sich in der Warteschlange `START_QUEUE` vier Nachrichten. Diese Warteschlange befindet sich im Warteschlangenmanager `QM_NEPTUNE` des Quellenagenten. Der vor den Daten der einzelnen Nachrichten einzufügende Textbegrenzer kann als Java-Literalzeichenfolge angegeben werden, wie z. B. `\n\u002D\u002D\u002D\n`.

Vorgehensweise

Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/output.txt  
-t text -sqdt "\n\u002D\u002D\u002D\n" -sqdp prefix -sq START_QUEUE
```

Der Textbegrenzer wird vom Quellenagenten `AGENT_NEPTUNE` zu Beginn der Daten aus jeder der vier Nachrichten in `START_QUEUE` hinzugefügt. Diese Daten werden in die Zieldatei `/out/output.txt` geschrieben.

Zugehörige Konzepte

„Daten von Nachrichten in Dateien übertragen“ auf Seite 292

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nachrichten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

Zugehörige Tasks

„Agent für die Ausführung der Übertragung einer Nachricht an eine Datei konfigurieren“ auf Seite 293

Standardmäßig können Agenten keine Übertragung einer Nachricht an eine Datei oder einer Datei an eine Nachricht ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft `enable-QueueInputOutput` auf den Wert 'true' setzen.

„[Beispiel: Aus einer Warteschlange in eine einzelne Datei übertragen](#)“ auf Seite 295

Sie können eine IBM MQ-Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter `-sq` mit dem Befehl `fteCreateTransfer` verwenden.

„[Beispiel: Gruppe von Nachrichten aus einer Warteschlange in eine einzelne Datei übertragen](#)“ auf Seite 296

Sie können eine einzelne vollständige Gruppe in einer IBM MQ-Warteschlange als Quelle einer Dateiübertragung angeben, indem Sie die Parameter `-sq` und `-sqgi` mit dem Befehl `fteCreateTransfer` verwenden.

„[Beispiel: Binärer Begrenzer nach den Daten aus jeder Nachricht einfügen](#)“ auf Seite 298

Bei der Übertragung im Binärmodus aus einer Quellenwarteschlange an eine Datei können Sie angeben, dass nach den Daten aus einzelnen Nachrichten ein binärer Begrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter `-sq`, `-sqdb` und `-sqdp` mit dem Befehl `fteCreateTransfer`.

„[Warteschlange überwachen und Variablensubstitution verwenden](#)“ auf Seite 258

Mit dem Befehl `fteCreateMonitor` können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

„[Beispiel: Fehlschlagen einer Nachricht-zu-Datei-Übertragung mit IBM MQ-Nachrichteneigenschaften herbeiführen](#)“ auf Seite 302

Sie können eine Nachricht-zu-Datei-Übertragung fehlschlagen lassen, indem Sie die Nachrichteneigenschaft `usr.UserReturnCode` IBM MQ auf einen Wert ungleich null setzen. Sie können auch zusätzliche Informationen zur Fehlerursache angeben, indem Sie die Nachrichteneigenschaft `usr.UserSupplement` IBM MQ festlegen.

Zugehörige Verweise

[fteCreateTransfer: Neue Dateiübertragung starten](#)

Beispiel: Binärer Begrenzer nach den Daten aus jeder Nachricht einfügen

Bei der Übertragung im Binärmodus aus einer Quellenwarteschlange an eine Datei können Sie angeben, dass nach den Daten aus einzelnen Nachrichten ein binärer Begrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter `-sq`, `-sqdb` und `-sqdp` mit dem Befehl `fteCreateTransfer`.

Informationen zu diesem Vorgang

In diesem Beispiel befinden sich in der Warteschlange `START_QUEUE` drei Nachrichten. Diese Warteschlange befindet sich im Warteschlangenmanager `QM_NEPTUNE` des Quellenagenten. Der binäre Begrenzer, der nach den Daten aus jeder Nachricht eingefügt werden soll, muss als eine durch Kommas getrennte Liste hexadezimaler Bytes angegeben werden; Beispiel: `x34, xE7, xAE`.

Vorgehensweise

Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/binary.file  
-sqdp postfix -sqdb x34,xE7,xAE -sq START_QUEUE
```

Der binäre Begrenzer wird den Daten aus jeder der drei Nachrichten in `START_QUEUE` durch den Quellenagenten `AGENT_NEPTUNE` angehängt. Diese Daten werden in die Zieldatei `/out/binary.file` geschrieben.

Zugehörige Konzepte

„[Daten von Nachrichten in Dateien übertragen](#)“ auf Seite 292

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nachrichten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

Zugehörige Tasks

„Agent für die Ausführung der Übertragung einer Nachricht an eine Datei konfigurieren“ auf Seite 293
Standardmäßig können Agenten keine Übertragung einer Nachricht an eine Datei oder einer Datei an eine Nachricht ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft `enableQueueInputOutput` auf den Wert 'true' setzen.

„Beispiel: Aus einer Warteschlange in eine einzelne Datei übertragen“ auf Seite 295
Sie können eine IBM MQ-Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-sq** mit dem Befehl **fteCreateTransfer** verwenden.

„Beispiel: Gruppe von Nachrichten aus einer Warteschlange in eine einzelne Datei übertragen“ auf Seite 296

Sie können eine einzelne vollständige Gruppe in einer IBM MQ-Warteschlange als Quelle einer Dateiübertragung angeben, indem Sie die Parameter **-sq** und **-sqgi** mit dem Befehl **fteCreateTransfer** verwenden.

„Beispiel: Textbegrenzer vor den Daten aus jeder Nachricht einfügen“ auf Seite 297

Bei der Übertragung im Textmodus aus einer Quellenwarteschlange in eine Datei können Sie angeben, dass vor den Daten aus einzelnen Nachrichten ein Textbegrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter **-sq**, **-sqdt** und **-sqdp** mit dem Befehl **fteCreateTransfer**.

„Warteschlange überwachen und Variablensubstitution verwenden“ auf Seite 258

Mit dem Befehl **fteCreateMonitor** können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

„Beispiel: Fehlschlagen einer Nachricht-zu-Datei-Übertragung mit IBM MQ-Nachrichteneigenschaften herbeiführen“ auf Seite 302

Sie können eine Nachricht-zu-Datei-Übertragung fehlschlagen lassen, indem Sie die Nachrichteneigenschaft `usr.UserReturnCode` IBM MQ auf einen Wert ungleich null setzen. Sie können auch zusätzliche Informationen zur Fehlerursache angeben, indem Sie die Nachrichteneigenschaft `usr.UserSupplement` IBM MQ festlegen.

Zugehörige Verweise

fteCreateTransfer: Neue Dateiübertragung starten

Warteschlange überwachen und Variablensubstitution verwenden

Mit dem Befehl **fteCreateMonitor** können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

Informationen zu diesem Vorgang

In diesem Beispiel stellt der Quellenagent `AGENT_VENUS` eine Verbindung zum Warteschlangenmanager `QM_VENUS` her. `AGENT_VENUS` überwacht die Warteschlange `START_QUEUE` auf `QM_VENUS`. Der Agent fragt die Warteschlange alle 30 Minuten ab.

Sobald die Warteschlange eine vollständige Nachrichtengruppe enthält, sendet die Überwachungstask die Gruppe an eine Datei auf einem von mehreren Zielagenten, die alle mit dem Warteschlangenmanager `QM_MARS` verbunden sind. Der Name der Datei, in die die Nachrichtengruppe übertragen wird, wird von der IBM MQ-Nachrichteneigenschaft `usr.fileName` der ersten Nachricht der Gruppe bestimmt. Der Name des Agenten, an den die Nachrichtengruppe gesendet wird, wird von der IBM MQ-Nachrichteneigenschaft `usr.toAgent` der ersten Nachricht der Gruppe bestimmt. Wenn der Header `usr.toAgent` nicht gesetzt ist, wird als Standardwert für den Zielagenten `AGENT_MAGENTA` verwendet.

Wenn Sie `useGroups="true"` und nicht gleichzeitig `groupId="{{GROUPID}}"` angeben, wird bei der Übertragung nur die erste Nachricht in der Warteschlange berücksichtigt. Wenn Sie beispielsweise 'file-Name' mithilfe der Variablensubstitution generieren, besteht die Möglichkeit, dass der Inhalt von `a.txt` nicht korrekt ist. Dies liegt daran, dass 'file-Name' von der Überwachung generiert wird, bei der Übertragung aber eine Nachricht abgerufen wird, bei der es sich nicht um die handelt, von der die Datei 'file-Name' generiert werden sollte.

Vorgehensweise

1. Erstellen Sie die Task-XML, die die Task definiert, die die Überwachung bei ihrer Auslösung ausführen soll.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="{{toAgent}}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{{GROUPID}}>START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/{{fileName}}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Die Variablen, die durch die Werte der IBM MQ-Nachrichtenheader ersetzt werden, sind **fett** hervorgehoben. Diese Task-XML wird in die Datei `/home/USER1/task.xml` gespeichert.

2. Erstellen Sie eine Ressourcenüberwachung zur Überwachung der Warteschlange `START_QUEUE`. Führen Sie folgenden Befehl aus:

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. Ein Benutzer oder ein Programm schreibt eine Gruppe von Nachrichten an die Warteschlange `START_QUEUE`.

Für die erste Nachricht dieser Gruppe sind die folgenden IBM MQ-Nachrichteneigenschaften gesetzt

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. Die Überwachung wird ausgelöst, sobald die vollständige Gruppe geschrieben wird. Der Agent überträgt die IBM MQ-Nachrichteneigenschaften in die Task-XML.

Die Task-XML wird entsprechend wie folgt umgesetzt:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
```

```

<destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
<transferSet>
  <item mode="binary" checksumMethod="none">
    <source>
      <queue useGroups="true" groupId="{GROUPEID}">START_QUEUE</queue>
    </source>
    <destination type="file" exist="overwrite">
      <file>/reports/larmer.rpt</file>
    </destination>
  </item>
</transferSet>
</managedTransfer>
</request>

```

Ergebnisse

Die von der Task-XML definierte Übertragung wird ausgeführt. Die vollständige Gruppe von Nachrichten, die von 'AGENT_VENUS' aus 'START_QUEUE' gelesen werden, wird in eine Datei mit dem Namen /reports/larmer.rpt auf dem System geschrieben, auf dem AGENT_VIOLET ausgeführt wird.

Nächste Schritte

Jede Nachricht in eine eigene Datei übertragen

Wenn eine Warteschlange überwacht werden und jede Nachricht in eine eigene Warteschlange übertragen werden soll, können Sie ähnlich wie zuvor in diesem Abschnitt beschrieben vorgehen.

1. Erstellen Sie den Monitor wie zuvor beschrieben, indem Sie den Parameter **-tr completeGroups** im Befehl **fteCreateMonitor** angeben.
2. Geben Sie in der Task-XML Folgendes an:

```

<queue useGroups="true" groupId="{GROUPEID}">START_QUEUE</queue>

```

Wenn die Nachrichten in die Quellenwarteschlange gestellt werden, sollten sie nicht in eine IBM MQ-Gruppe eingefügt werden. Fügen Sie jeder Nachricht die IBM MQ-Nachrichteneigenschaften hinzu. Geben Sie beispielsweise für die Eigenschaft `usr.filename` für jede Nachricht einen eindeutigen Dateinamen an. Dadurch werden die einzelnen Nachrichten in der Quellenwarteschlange vom Managed File Transfer Agent wie jeweils eine eigene Gruppe verarbeitet.

Zugehörige Konzepte

[„Daten von Nachrichten in Dateien übertragen“ auf Seite 292](#)

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nachrichten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

[„Anpassen von MFT Ressourcenüberwachungsaufgaben mit Variablensubstitution“ auf Seite 252](#)

Wenn die Auslöserbedingungen für eine aktive Ressourcenüberwachung erfüllt sind, wird die definierte Task aufgerufen. Zusätzlich zum Aufrufen der Übertragungs- oder Befehlstask mit demselben Zielagenten oder demselben Zieldateinamen können Sie die Taskdefinition auch zur Ausführungszeit ändern. Dies tun Sie, indem Sie Variablennamen in die Taskdefinitions-XML einfügen. Wenn der Monitor feststellt, dass die Auslöserbedingungen erfüllt sind, und dass die Taskdefinition Variablennamen enthält, ersetzt er die Variablennamen mit den Variablenwerten und ruft dann die Task auf.

[Die Zieldateien, die bei einer Übertragung erstellt wurden, die von der Überwachung einer Warteschlangenressource gestartet wurde, enthalten die falschen Daten](#)

Zugehörige Tasks

[„MFT-Überwachungstasks für den Start von Befehlen und Scripts konfigurieren“ auf Seite 245](#)

Ressourcenmonitore sind nicht auf die Ausführung von Dateiübertragungen als ihre zugeordnete Task beschränkt. Sie können den Monitor auch so konfigurieren, dass er andere Befehle vom Überwachungsagenten, einschließlich ausführbarer Programme, Ant-Scripts oder JCL-Jobs, aufruft. Wenn Sie Befehle

aufrufen möchten, bearbeiten Sie die Monitor-Taskdefinitions-XML so, dass sie ein oder mehrere Befehls-
elemente mit entsprechenden Befehlsaufrufparametern, wie z. B. Argumente und Eigenschaften, enthält.

„[Beispiel: MFT-Ressource konfigurieren](#)“ auf Seite 251

Sie können eine IBM MQ -Warteschlange als Ressource angeben, die von einer Ressourcenüberwachung
überwacht werden soll, indem Sie den Parameter **-mq** mit dem Befehl **fteCreateMonitor** verwenden.

Zugehörige Verweise

fteCreateMonitor: MFT-Ressourcenüberwachung erstellen

[Von MFT gelesene MQ-Nachrichteneigenschaften aus Nachrichten in Quellenwarteschlangen](#)

Beispiel: Fehlschlagen einer Nachricht-zu-Datei-Übertragung mit IBM MQ- Nachrichteneigenschaften herbeiführen

Sie können eine Nachricht-zu-Datei-Übertragung fehlschlagen lassen, indem Sie die Nachrichteneigen-
schaft `usr.UserReturnCode` IBM MQ auf einen Wert ungleich null setzen. Sie können auch zusätzliche
Informationen zur Fehlerursache angeben, indem Sie die Nachrichteneigenschaft `usr.UserSupplement`
IBM MQ festlegen.

Informationen zu diesem Vorgang

In diesem Beispiel wird eine Übertragung zwischen der Warteschlange `INPUT_QUEUE` und der Datei /
`home/user/output.file` ausgeführt.

Ein Benutzer erstellt Nachrichten und reiht sie in die Warteschlange `INPUT_QUEUE` ein. Der Quellenagent
ruft Nachrichten aus der Warteschlange `INPUT_QUEUE` ab und sendet die Übertragungsdaten an den
Zielagenten. Der Zielagent schreibt diese Daten in die Datei `/home/user/output.file`.

Der Benutzer, der Nachrichten in die Warteschlange `INPUT_QUEUE` stellt, möchte die laufende Übertra-
gung stoppen und alle bereits in die Zieldatei geschriebenen Daten löschen.

Vorgehensweise

1. Der Benutzer schreibt eine Nachricht an die Warteschlange `INPUT_QUEUE`, in der folgende IBM MQ-
Nachrichteneigenschaften gesetzt sind:

```
usr.UserReturnCode=1  
usr.UserSupplement="Cancelling transfer - sent wrong data."
```

2. Der Quellenagent liest die IBM MQ-Nachrichteneigenschaften und stoppt die Verarbeitung der Nach-
richten aus der Warteschlange. Der Zielagent löscht alle bereits in das Zielverzeichnis geschriebenen
Dateidaten.
3. Der Quellenagent sendet dem Koordinationswarteschlangenmanager eine Übertragungsprotokoll-
nachricht, die den Übertragungsfehler meldet.

Die Nachricht hat folgenden Inhalt:

```
<?xml version="1.0" encoding="UTF-8"?>  
<transaction version="1.00"  
  ID="414d5120514d31202020202020202020207e970d4920008702" agentRole="sourceAgent"  
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"  
  xmlns="">  
  <action time="2008-11-02T21:28:09.593Z">progress</action>  
  <sourceAgent agent="FTEAGENT" QMgr="QM1">  
    <systemInfo architecture="x86" name="Windows 7"  
      version="6.1 build 7601 Service Pack 1" />  
  </sourceAgent>  
  <destinationAgent agent="FTEAGENT" QMgr="QM1">  
    <systemInfo architecture="x86" name="Windows 7"  
      version="6.1 build 7601 Service Pack 1" />  
  </destinationAgent>  
  <originator>  
    <hostName>reportserver.com</hostName>  
    <userID>USER1</userID>  
    <mqmdUserID>USER1 </mqmdUserID>
```

```

</originator>
<transferSet index="0" size="1"
  startTime="2008-11-02T21:28:09.281Z"
  total="1">
  <item mode="binary">
    <source>
      <queue>INPUT_QUEUE@QM1</queue>
    </source>
    <destination exist="error">
      <file>/home/user/output.file</file>
    </destination>
    <status resultCode="1">
      <supplement>Cancelling transfer - sent wrong data.</supplement>
    </status>
  </item>
</transferSet>
</transaction>

```

Zugehörige Konzepte

[„Daten von Nachrichten in Dateien übertragen“](#) auf Seite 292

Mit der Managed File Transfer-Funktion zur Nachrichtenübertragung in eine Datei können Sie eine oder auch mehrere Nachrichten aus einer IBM MQ-Warteschlange in eine Datei, in ein Dataset (unter z/OS) oder in einen Benutzerdateibereich übertragen. Wenn Sie eine Anwendung verwenden, die IBM MQ-Nachrichten erstellt oder verarbeitet, können Sie diese Nachrichten mit dieser Funktion von Managed File Transfer in eine Datei auf jedem System in Ihrem Managed File Transfer-Netz übertragen.

Zugehörige Tasks

[„Agent für die Ausführung der Übertragung einer Nachricht an eine Datei konfigurieren“](#) auf Seite 293

Standardmäßig können Agenten keine Übertragung einer Nachricht an eine Datei oder einer Datei an eine Nachricht ausführen. Zur Aktivierung dieser Funktion müssen Sie die Agenteneigenschaft `enableQueueInputOutput` auf den Wert 'true' setzen.

[„Beispiel: Aus einer Warteschlange in eine einzelne Datei übertragen“](#) auf Seite 295

Sie können eine IBM MQ-Warteschlange als Ziel einer Dateiübertragung angeben, indem Sie den Parameter **-sq** mit dem Befehl **fteCreateTransfer** verwenden.

[„Beispiel: Gruppe von Nachrichten aus einer Warteschlange in eine einzelne Datei übertragen“](#) auf Seite 296

Sie können eine einzelne vollständige Gruppe in einer IBM MQ-Warteschlange als Quelle einer Dateiübertragung angeben, indem Sie die Parameter **-sq** und **-sqgi** mit dem Befehl **fteCreateTransfer** verwenden.

[„Beispiel: Textbegrenzer vor den Daten aus jeder Nachricht einfügen“](#) auf Seite 297

Bei der Übertragung im Textmodus aus einer Quellenwarteschlange in eine Datei können Sie angeben, dass vor den Daten aus einzelnen Nachrichten ein Textbegrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter **-sq**, **-sqdt** und **-sqdp** mit dem Befehl **fteCreateTransfer**.

[„Beispiel: Binärer Begrenzer nach den Daten aus jeder Nachricht einfügen“](#) auf Seite 298

Bei der Übertragung im Binärmodus aus einer Quellenwarteschlange an eine Datei können Sie angeben, dass nach den Daten aus einzelnen Nachrichten ein binärer Begrenzer eingefügt werden soll. Verwenden Sie hierfür die Parameter **-sq**, **-sqdb** und **-sqdp** mit dem Befehl **fteCreateTransfer**.

[„Warteschlange überwachen und Variablensubstitution verwenden“](#) auf Seite 258

Mit dem Befehl **fteCreateMonitor** können Sie eine Warteschlange überwachen und Nachrichten aus dieser Warteschlange in eine Datei übertragen. Der Wert jeder IBM MQ-Nachrichteneigenschaft in der ersten aus der überwachten Warteschlange gelesenen Nachricht kann in die Task-XML-Definition übertragen werden und somit das Übertragungsverhalten bestimmen.

Zugehörige Verweise

[Von MFT gelesene MQ-Nachrichteneigenschaften aus Nachrichten in Quellenwarteschlangen](#)

Die Protokollbridge

Die Protokollbridge ermöglicht in Ihrem Managed File Transfer- bzw. MFT-Netz den Zugriff auf Dateien, die in Ihrer lokalen Domäne oder remote auf einem Dateiserver außerhalb Ihres MFT-Netzes gespeichert sind. Dieser Dateiserver kann die FTP-, FTPS- oder SFTP-Netzprotokolle verwenden. Für jeden Dateiserver

ist mindestens ein dedizierter Agent erforderlich. Der dedizierte Agent wird als Protokoll-Bridge-Agent bezeichnet. Ein Bridge-Agent kann mit mehreren Dateiservern interagieren.

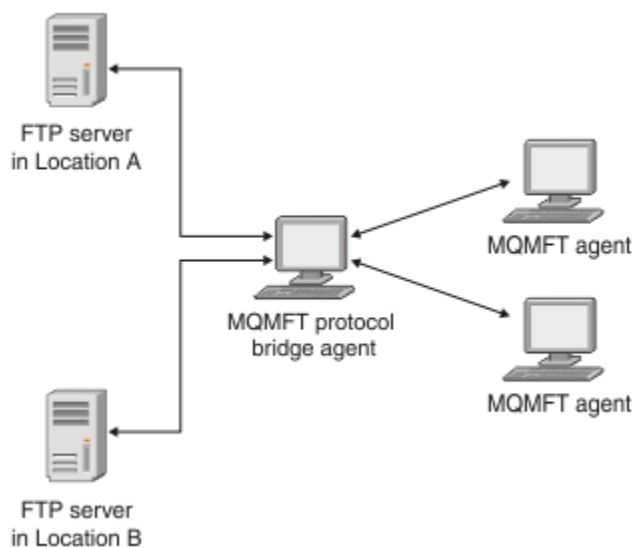
Die Protokollbridge wird als Teil der Servicekomponente von Managed File Transfer bereitgestellt. Auf einem einzelnen System mit MFT können mehrere dedizierte Agenten installiert sein, die Verbindungen zu verschiedenen Dateiservern herstellen.

Sie können einen Protokoll-Bridge-Agenten verwenden, um Dateien gleichzeitig an mehrere Endpunkte zu übertragen. MFT stellt eine Datei namens `ProtocolBridgeProperties.xml` zur Verfügung, die Sie bearbeiten können, um die verschiedenen Protokolldateiserver zu definieren, an die Sie Dateien übertragen möchten. Der Befehl **fteCreateBridgeAgent** fügt die Details des Standardprotokolldateiservers zu `ProtocolBridgeProperties.xml` für Sie hinzu. Diese Datei wird im Dateiformat der Protokollbrückeneigenschaften beschrieben.

Sie können den Protokollbrückenagenten verwenden, um die folgenden Aktionen auszuführen:

- Hochladen von Dateien aus dem MFT-Netz auf einen Remote Server mittels FTP, FTPS oder SFTP
- Herunterladen von Dateien von einem Remote Server mittels FTP, FTPS oder SFTP in das MFT-Netz

Anmerkung: Der Protokollbrückenagent kann nur FTP-, FTPS- oder SFTP-Server unterstützen, auf die Dateien durch ihren absoluten Dateipfad zugegriffen werden können. Wenn in einer Übertragungsanforderung ein relativer Dateipfad angegeben ist, versucht der Protokoll-Bridge-Agent, den relativen Pfad in einen absoluten Dateipfad basierend auf dem Ausgangsverzeichnis zu konvertieren, das für die Anmeldung am Protokollserver verwendet wird. Diese Protokollserver, die den Zugriff auf Dateien, die nur auf dem aktuellen Verzeichnis basieren, zulassen, werden vom Protokoll-Bridge-Agenten nicht unterstützt.



Das Diagramm zeigt zwei FTP-Server an verschiedenen Standorten. Diese FTP-Server werden zum Austausch von Dateien mit den Managed File Transfer-Agenten verwendet. Der Protokollbrückenagent befindet sich zwischen den FTP-Servern und dem restlichen MFT-Netz und ist so konfiguriert, dass er mit beiden FTP-Servern kommunizieren kann.

Stellen Sie sicher, dass neben dem Protokollbrückenagenten noch ein weiterer Agent im MFT-Netz vorhanden ist. Der Protokollbrückenagent ist nur eine Brücke zum FTP-, FTPS- oder SFTP-Server und schreibt keine übertragenen Dateien auf die lokale Platte. Wenn Sie Dateien in den FTP-, FTPS- oder SFTP-Server übertragen wollen, müssen Sie den Protokollbrückenagenten als Ziel oder Quelle für die Dateiübertragung (die den FTP-, FTPS- oder SFTP-Server darstellen) und einen anderen Standardagenten als die entsprechende Quelle oder Zieladresse verwenden.

Wenn Sie Dateien mit der Protokoll-Bridge übertragen, muss die Brücke über die Berechtigung zum Lesen des Quellen- oder Zielverzeichnisses verfügen, das die Dateien enthält, die übertragen werden sollen. Wenn Sie beispielsweise Dateien aus dem Verzeichnis `/home/fte/bridge`, das Ausführungsbe-

rechtigungen (d -- x -- x -- x) hat, übertragen möchten, schlagen alle Übertragungen, die Sie in diesem Verzeichnis versuchen, mit der folgenden Fehlernachricht fehl:

```
BFGBR0032E: Attempt to read filename from the protocol file server  
has failed with server error 550. Failed to open file.
```

Protokoll-Bridge-Agenten konfigurieren

Ein Protokollbridgeagent gleicht einem MFT-Standardagenten. Erstellen Sie mit dem Befehl **fteCreateBridgeAgent** einen Protokollbridgeagenten. Sie können einen Protokollbridgeagenten mit der `ProtocolBridgeProperties.xml`-Datei konfigurieren, die in [Dateiformat für Protokollbridge-Eigenschaften](#) beschrieben ist. Wenn Sie eine frühere Version verwenden, konfigurieren Sie den Agenten mit den in [Erweiterte Agenteneigenschaften: Protokollbridge](#) und [Erweiterte Agenteneigenschaften: Protokollbridgeagentenprotokollierung](#) beschriebenen Eigenschaften für die Protokollbridge. Bei allen Versionen kann auch eine Berechtigungsnachweisuordnung konfiguriert werden. Eine entsprechende Beschreibung finden Sie im Abschnitt „[Berechtigungsnachweise für einen Dateiserver zuordnen](#)“ auf Seite 313. Nachdem Sie einen Protokoll-Bridge-Agenten für einen bestimmten Protokolldateiserver konfiguriert haben, können Sie diesen Agenten nur zu diesem Zweck verwenden.

Protokollbrückenwiederherstellung

Wenn der Protokoll-Bridge-Agent keine Verbindung zum Dateiserver herstellen kann, weil der Dateiserver nicht verfügbar ist, werden alle Dateiübertragungsanforderungen in die Warteschlange gestellt, bis der Dateiserver verfügbar wird. Wenn der Protokoll-Bridge-Agent keine Verbindung zum Dateiserver herstellen kann, weil der Agent die falsche Berechtigungsnachweise verwendet, schlägt die Übertragung fehl, und die Übertragungsprotokollnachricht spiegelt diesen Fehler wider. Wenn der Protokoll-Bridge-Agent aus irgendeinem Grund beendet wird, werden alle angeforderten Dateiübertragungen beibehalten und fortgesetzt, wenn die Protokoll-Bridge erneut gestartet wird.

Während der Dateiübertragung werden Dateien normalerweise als temporäre Dateien an der Zieladresse geschrieben und dann umbenannt, wenn die Übertragung abgeschlossen ist. Wenn es sich bei dem Übertragungsziel jedoch um einen Protokolldateiserver handelt, der als eingeschränkter Schreibzugriff konfiguriert ist (Benutzer können Dateien auf den Protokolldateiserver hochladen, diese hochgeladenen Dateien jedoch nicht ändern können; effektiv Benutzer können nur einmal schreiben), werden die übertragenen Dateien direkt an das Ziel geschrieben. Wenn also während der Übertragung ein Problem auftritt, bleiben die teilweise geschriebenen Dateien auf dem Zielprotokolldateiserver und können von Managed File Transfer weder gelöscht noch bearbeitet werden. In dieser Situation schlägt die Übertragung fehl.

Zugehörige Tasks

„[Beispiel: Protokollbridgeagenten für die Verwendung von privaten Schlüsselberechtigungs-nachweisen für einen UNIX-SFTP-Server konfigurieren](#)“ auf Seite 318

Dieses Beispiel veranschaulicht, wie Sie die `ProtocolBridgeCredentials.xml`-Datei generieren und konfigurieren können. Hierbei handelt es sich um ein typisches Beispiel, die Details können also je nach Plattform variieren, die Prinzipien bleiben jedoch identisch.

„[Eigenschaften für Protokolldateiserver mit der ProtocolBridgeProperties.xml -Datei definieren](#)“ auf Seite 306

Definieren Sie die Eigenschaften eines oder mehrerer Protokolldateiserver, auf die Sie Dateien übertragen möchten, und verwenden Sie dazu die `ProtocolBridgeProperties.xml`-Datei, die von Managed File Transfer im Konfigurationsverzeichnis des Agenten bereitgestellt wird.

Zugehörige Verweise

[fteCreateBridgeAgent \(MFT-Protokollbridgeagenten erstellen und konfigurieren\)](#)

„[Berechtigungsnachweise für einen Dateiserver zuordnen](#)“ auf Seite 313

Die Zuordnung von Benutzerberechtigungen in Managed File Transfer zu den Benutzerberechtigungen auf dem Dateiserver kann über die Standardfunktion des Protokollbridgeagenten für Credential-Mapping oder über einen eigens geschriebenen Benutzerexit erfolgen. Managed File Transfer stellt einen Beispielbenutzerexit bereit, der die Zuordnung der Benutzerberechtigungs-nachweise durchführt.

[ProtocolBridgeCredentialExit.java-Schnittstelle](#)

Eigenschaften für Protokolldateiserver mit der ProtocolBridgeProperties.xml -Datei definieren

Definieren Sie die Eigenschaften eines oder mehrerer Protokolldateiserver, auf die Sie Dateien übertragen möchten, und verwenden Sie dazu die ProtocolBridgeProperties.xml -Datei, die von Managed File Transfer im Konfigurationsverzeichnis des Agenten bereitgestellt wird.

Informationen zu diesem Vorgang

Der Befehl **fteCreateBridgeAgent** erstellt die Datei ProtocolBridgeProperties.xml im Agentenkonfigurationsverzeichnis *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name*. Der Befehl erstellt außerdem einen Eintrag in der Datei für den Standardprotokolldateiserver, wenn bei der Ausführung des Befehls ein Standardwert angegeben wurde.

Die Nachricht BFGCL0392I gibt die Position der ProtocolBridgeProperties.xml -Datei an.

```
<?xml version="1.0" encoding="IBM-1047"?>
<!--
This ProtocolBridgeProperties.xml file determines the protocol servers that will be accessed by
the
MQMFT protocol bridge agent.

Each protocol server is defined using either a <tns:ftpServer>, <tns:ftpsServer>, or
<tns:sftpServer>
element - depending on the protocol used to communicate with the server. When the protocol
bridge agent participates in a managed file transfer it will determine which server to use
based on
the prefix (if any) present on the file path. For example a file path of 'server1:/home/user/
file.txt' would
be interpreted as a request to transfer /home/user/file.txt using 'server1'. The server name
is compared
to the 'name' attribute of each <tns:ftpServer>, <tns:ftpsServer> or <tns:sftpServer> element
in this
XML document and the first match is used to determine which protocol server the protocol bridge
agent will connect to. If no match is found then the managed file transfer operation will fail.

If a file path is not prefixed with a server name, for example '/home/user/file.txt' then this
XML
document can specify a default server to use for the managed file transfer. To specify a
default server use the <tns:defaultServer> element as the first element inside the <tns:server
Properties>
element. The default server will be used whenever the protocol bridge agent participates in
a managed file transfer for file names which do not specify a prefix.

An optional <tns:limits> element can be specified within each server definition. This element
contains
attributes that govern the amount of resources used by each defined server.

An optional <tns:credentialsFile> element can be specified within each server definition. This
element contains a path to a file containing credentials to be used when connecting to defined
servers.

An example ProtocolBridgeProperties.xml file is as follows:

<?xml version="1.0" encoding="UTF-8"?>
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties ProtocolBridgePropert
ies.xsd">
  <tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
  <tns:defaultServer name="myFTPserver" />
  <tns:ftpServer name="myFTPserver" host="windows.hursley.ibm.com" port="1234" platform="wind
ows"
    timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
    listFormat="unix" limitedWrite="false">
  <tns:limits maxListFileNames="100" maxListDirectoryLevels="999999999"

```

```

        maxReconnectRetry="2" reconnectWaitPeriod="10"
        maxSessions="60" socketTimeout="30" />

</tns:ftpServer>

<tns:ftpsServer name="myFTPSserver" host="unix.hursley.ibm.com" platform="unix"
    timeZone="Europe/London" locale="en_GB" fileEncoding="UTF8"
    listFormat="unix" limitedWrite="false" ftpsType="explicit"
    trustStore="C:\FTE\keystores\myFTPSserver\FTPSKeyStore.jks" trustStorePassword="pass
word">

    <tns:limits maxReconnectRetry="10" connectionTimeout="10"/>

</tns:ftpsServer>

<tns:sftpServer name="mySFTPserver" host="windows.hursley.ibm.com" platform="windows"
    timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
    limitedWrite="false">

    <tns:limits connectionTimeout="60"/>

</tns:sftpServer>
</tns:serverProperties>

```

This example shows the outermost <tns:serverProperties> element which must exist for the document to be valid, an optional <tns:defaultServer> element, as well as definitions for an FTP, FTPS and SFTP server.

The attributes of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements determine the characteristics of the connection established to the server. These attributes correspond to the command line parameters for the 'fteCreateBridgeAgent' command.

The following attributes are valid for all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: name, host, port, platform, fileEncoding, limitedWrite and controlEncoding.

The following attributes are valid for the <tns:ftpServer> and <tns:ftpsServer> elements: timeZone, locale, listFormat, listFileRecentDateFormat, listFileOldDateFormat, and monthShortNames.

The following attributes are valid for the <tns:ftpServer> element only: passiveMode

The following attributes are valid for the <tns:ftpsServer> element only: ftpsType, trustStore, trustStorePassword, trustStoreType, keyStore, keyStorePassword, keyStoreType, ccc, protFirst, auth, and connectTimeout.

The following attributes are valid for the <tns:limits> element within all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: maxListFileNames, maxListDirectoryLevels, maxReconnectRetry, reconnectWaitPeriod, maxSessions and socketTimeout

```

-->
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
    xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties ProtocolBridgeProperties.xsd">

    <!-- By default the location of the credentials file is in the home directory of the user
    that started the -->
    <!-- protocol bridge agent. If you wish to specify a different location use the credentials
    File element to -->
    <!-- do this. For example: -->
    <!-- <tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml" /> -->
    <!-- <tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml" /> -->

    <tns:defaultServer name="WINMVSCA.HURSLEY.IBM.COM" />
    <tns:ftpServer name="WINMVSCA.HURSLEY.IBM.COM" host="WINMVSCA.HURSLEY.IBM.COM" platform="unix"
        timeZone="Europe/London" locale="en-GB" fileEncoding="US-ASCII"
        listFormat="unix" limitedWrite="false" />

    <!-- Define servers here -->
</tns:serverProperties>

```

Der Befehl kann die folgende Nachricht erzeugen: BFGCL0532I :

Damit dieser Agent funktioniert, muss eine zusätzliche Berechtigungsnachweisdatei manuell erstellt werden. Standardmäßig wird diese Datei als ProtocolBridgeCredentials.xml bezeichnet und befindet sich im Ausgangsverzeichnis. Verzeichnis des Benutzers, der den Agenten startet. Beispiel: Wenn dieser Benutzer den Agenten gestartet hat Die Position wäre: \$HOME/ProtocolBridgeCredentials.xml

Wenn Sie eine Berechtigungsnachweisdatei verwenden:

1. Weitere Informationen zum Erstellen von einem finden Sie im folgenden Text.
2. Die Berechtigungsnachweisdatei muss sich in einem Verzeichnis mit eingeschränkten Berechtigungen befinden. Es darf z. B. kein Lesezugriff für andere Benutzer vorhanden sein.
3. Geben Sie die Position des Verzeichnisses für die Berechtigungsnachweisdatei in der \$HOME -Umgebungsvariablen für die Benutzer-ID des gestarteten Agenten an, oder bearbeiten Sie die Datei ProtocolBridgeProperties.xml und geben Sie die Position in an:

```
<tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml"/>
```

Wenn Sie weitere Nicht-Standardprotokollserver hinzufügen möchten, bearbeiten Sie diese Datei, um ihre Eigenschaften zu definieren. In diesem Beispiel wird ein zusätzlicher FTP-Server hinzugefügt.

Anmerkung: Der Protokoll-Bridge-Agent unterstützt keine Dateispernung. Dies liegt daran, dass Managed File Transfer den Dateispernungsmechanismus auf einem Dateiserver nicht unterstützt.

Vorgehensweise

1. Definieren Sie einen Protokolldateiserver, indem Sie die folgenden Zeilen als untergeordnetes Element von <tns:serverProperties>in die Datei einfügen:

```
<tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234"
platform="windows"
timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8"
listFormat="unix" limitedWrite="false" >
<tns:limits maxListFileNames="10" maxListDirectoryLevels="500"/>
```

2. Ändern Sie anschließend den Wert der Attribute:

- name ist der Name Ihres Protokolldateiservers.
- host ist der Hostname oder die IP-Adresse des Protokolldateiservers.
- port ist die Anschlussnummer des Protokolldateiservers.
- platform ist die Plattform, auf der der Protokolldateiserver ausgeführt wird.
- timeZone ist die Zeitzone, in der der Protokolldateiserver ausgeführt wird.
- locale ist die Sprache, die auf dem Protokolldateiserver verwendet wird.
- fileEncoding ist die Zeichencodierung des Protokolldateiservers.
- listFormat ist das Dateilistenformat, das vom Protokolldateiserver zurückgegeben wird.
- limitedWrite bestimmt, ob der Standardmodus beim Schreiben in einen Dateiserver verwendet werden soll. Dies ist die Erstellung einer temporären Datei und die Umbenennung dieser Datei, wenn die Übertragung abgeschlossen ist. Für einen Dateiserver, der nur als Schreibzugriff konfiguriert ist, wird die Datei direkt mit ihrem endgültigen Namen erstellt. Der Wert dieser Eigenschaft kann 'true' oder 'false' sein. Das Attribut limitedWrite und die Agenteneigenschaft doNotUseTempOutputFile werden bei Protokollbrückenagenten gemeinsam verwendet. Wenn Sie temporäre Dateien verwenden möchten, müssen Sie den Wert von doNotUseTempOutputFile nicht festlegen, und Sie müssen den Wert von limitedWrite auf "false" setzen. Jede andere Kombination von Einstellungen bedeutet, dass temporäre Dateien nicht verwendet werden.
- maxListFileNames ist die maximale Anzahl der Namen, die beim Durchsuchen eines Verzeichnisses auf dem Protokolldateiserver für Dateinamen erfasst werden.

- `maxListDirectoryLevels` ist die maximale Anzahl der Verzeichnisebenen, die beim Durchsuchen eines Verzeichnisses auf dem Protokolldateiserver für Dateinamen rekursiv ausgeführt werden können.

Weitere Informationen zu diesen Attributen, einschließlich der Angabe, ob sie erforderlich oder optional sind, und ihre Standardwerte finden Sie im Abschnitt [Format der Protokollbrückeneigenschaftendatei](#).

Zugehörige Verweise

[Format der Protokollbrückeneigenschaftendatei](#)

[Von MFT verwendete reguläre Ausdrücke](#)

Eigenschaften des Protokolldateiservers suchen: `ProtocolBridgePropertiesExit2`

Wenn Sie über eine große Anzahl von Protokolldateiservern verfügen, können Sie die `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` -Schnittstelle implementieren, um die Eigenschaften des Protokolldateiservers zu suchen, auf die in Übertragungen verwiesen wird. Sie können diese Schnittstelle bevorzugt implementieren, um eine `ProtocolBridgeProperties.xml` -Datei zu verwalten.

Informationen zu diesem Vorgang

Managed File Transfer stellt einen Beispielbenutzerexit bereit, der die Eigenschaften von Protokolldateiservern nachschlägt. Weitere Informationen finden Sie unter [„Beispielbenutzerexit für die Suche von Eigenschaften des Protokolldateiservers verwenden“](#) auf Seite 310.

Jeder Benutzerexit, der die Eigenschaften der Protokollbridge sucht, muss die Schnittstelle `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` implementieren. Weitere Informationen finden Sie im Abschnitt [ProtocolBridgePropertiesExit.java-Schnittstelle](#).

Sie können mehrere Protokoll-Server-Eigenschaften auf ähnliche Weise miteinander verketteten, dass andere Benutzerexits vorhanden sind. Die Exits werden in der Reihenfolge aufgerufen, in der sie mit der Eigenschaft `protocolBridgePropertiesExitClasses` in der Agenteneigenschaftendatei angegeben sind. Die Initialisierungsmethoden werden alle separat zurückgegeben, und wenn mindestens ein Wert 'false' zurückgegeben wird, wird der Agent nicht gestartet. Der Fehler wird im Ereignisprotokoll des Agenten dokumentiert.

Es wird nur ein Gesamtergebnis für die Methoden 'getProtocolServerProperties' aller Exits zurückgegeben. Wenn die Methode ein Eigenschaftenobjekt als Ergebniscode zurückgibt, ist dieser Wert das zurückgegebene Ergebnis, und die Methoden `getProtocolServerProperties` der nachfolgenden Exits werden nicht aufgerufen. Wenn die Methode den Wert null als Ergebniscode zurückgibt, wird die Methode "getProtocolServerProperties" des nächsten Exits aufgerufen. Wenn kein nachfolgender Exit vorhanden ist, wird das Nulresultat zurückgegeben. Ein allgemeiner Ergebniscode von null wird als Lookup-Fehler durch den Protokollbrückenagenten betrachtet.

Inzwischen wird zwar die Verwendung der Schnittstelle `ProtocolBridgePropertiesExit2.java` empfohlen, nähere Informationen zur Schnittstelle `ProtocolBridgePropertiesExit.java` finden Sie aber nach wie vor unter [„Eigenschaften des Protokolldateiservers suchen: ProtocolBridgePropertiesExit“](#) auf Seite 311.

Führen Sie die folgenden Schritte aus, um den Exit auszuführen:

Vorgehensweise

1. Kompilieren Sie den Benutzerexit für die Protokollservereigenschaften.
2. Erstellen Sie eine Java-Archivdatei (JAR-Datei) mit dem kompilierten Exit und seiner Paketstruktur.
3. Setzen Sie die JAR-Datei, die die Exitklasse enthält, im Verzeichnis `exits` des Protokollbridgeagenten. Dieses Verzeichnis befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.

4. Bearbeiten Sie die Eigenschaftendatei des Protokollbrückenagenten, um die Eigenschaft `protocolBridgePropertiesExitClasses` einzuschließen. Geben Sie für den Wert dieser Eigenschaft eine durch Kommas getrennte Liste von Klassen an, die einen Benutzerexit für die Eigenschaften eines Protokoll-Bridge-Servers implementieren. Die Exitklassen werden in der Reihenfolge aufgerufen, in der sie in dieser Liste angegeben sind. Weitere Informationen finden Sie unter [Die MFT agent.properties-Datei](#).
5. Sie können die Eigenschaft `protocolBridgePropertiesConfiguration` optional angeben. Der Wert, den Sie für diese Eigenschaft angeben, wird als Zeichenfolge an die Methode "initialize ()" der von `protocolBridgePropertiesExitClasses` angegebenen Exitklassen übergeben. Weitere Informationen finden Sie unter [Die MFT agent.properties-Datei](#).

Beispielbenutzerexit für die Suche von Eigenschaften des Protokolldateiservers verwenden

Managed File Transfer stellt einen Beispielbenutzerexit bereit, der die Eigenschaften von Protokolldateiservern nachschlägt.

Informationen zu diesem Vorgang

Ein Beispielbenutzerexit, der Protokollbrückeeigenschaften sucht, wird im `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge`-Verzeichnis und im Thema [Benutzerexit für Beispielprotokollbrückeeigenschaften](#) bereitgestellt.

Der Exit `SamplePropertiesExit2.java` liest eine Eigenschaftendatei, die Eigenschaften für Protokollserver enthält. Das Format der einzelnen Einträge in der Eigenschaftendatei lautet wie folgt:

```
serverName=type://host:port
```

Die Position der Eigenschaftendatei wird aus der Eigenschaft `protocolBridgePropertiesConfiguration` des Protokollbrückenagenten übernommen.

Führen Sie die folgenden Schritte aus, um den Musterbenutzerexit auszuführen:

Vorgehensweise

1. Kompilieren Sie die `SamplePropertiesExit2.java`-Datei.
2. Erstellen Sie eine JAR-Datei, die den kompilierten Exit und die zugehörige Paketstruktur enthält.
3. Setzen Sie die JAR-Datei in das Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/exits`.
4. Bearbeiten Sie die `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties`-Datei so, dass sie die folgende Zeile enthält:

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. Erstellen Sie eine Eigenschaftendatei für die Protokollbrücke, z. B. `protocol_bridge_properties.properties`, im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent`. Bearbeiten Sie diese Datei, um Einträge in das Format aufzunehmen:

```
serverName=type://host:port
```

6. Bearbeiten Sie die `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties`-Datei so, dass sie die folgende Zeile enthält:

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

Sie müssen den absoluten Pfad zur `protocol_bridge_properties.properties`-Datei verwenden.

7. Starten Sie den Protokollbridgeagenten mit dem Befehl **fteStartAgent** .

Zugehörige Konzepte

[„Die Protokollbridge“ auf Seite 303](#)

Die Protokollbridge ermöglicht in Ihrem Managed File Transfer- bzw. MFT-Netz den Zugriff auf Dateien, die in Ihrer lokalen Domäne oder remote auf einem Dateiserver außerhalb Ihres MFT-Netztes gespeichert sind. Dieser Dateiserver kann die FTP-, FTPS- oder SFTP-Netzprotokolle verwenden. Für jeden Dateiserver ist mindestens ein dedizierter Agent erforderlich. Der dedizierte Agent wird als Protokoll-Bridge-Agent bezeichnet. Ein Bridge-Agent kann mit mehreren Dateiservern interagieren.

Zugehörige Verweise

[ProtocolBridgePropertiesExit.java-Schnittstelle](#)

[Benutzerexit für Eigenschaften der Protokollbrückeneigenschaften](#)

[Die MFT agent.properties-Datei](#)

[fteCreateBridgeAgent \(MFT-Protokollbridgeagenten erstellen und konfigurieren\)](#)

Eigenschaften des Protokolldateiservers suchen: ProtocolBridgePropertiesExit

Wenn Sie über eine große Anzahl von Protokolldateiservern verfügen, können Sie die `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` -Schnittstelle implementieren, um die Eigenschaften des Protokolldateiservers zu suchen, auf die in Übertragungen verwiesen wird.

Informationen zu diesem Vorgang

Sie können die Schnittstelle `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` implementieren, statt die Datei `ProtocolBridgeProperties.xml` zu verwalten. Verwenden Sie die Schnittstelle `ProtocolBridgePropertiesExit2.java` . Die Methode `getCredentialLocation` in `ProtocolBridgePropertiesExit2.java` verwendet die Standardposition der Datei `ProtocolBridgeCredentials.xml` , die Ihr Ausgangsverzeichnis ist.

Jeder Benutzerexit, der Protokollbrückeneigenschaften sucht, muss die Schnittstelle `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` implementieren:

```
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;
import java.util.Properties;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will be
 * invoked by a protocol bridge agent to look up properties for protocol servers
 * that are referenced in transfers.
 * <p>
 * There will be one instance of each implementation class for each protocol
 * bridge agent. The methods can be called from different threads so the methods
 * must be synchronised.
 */
public interface ProtocolBridgePropertiesExit {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to
     * initialize any resources that are required by the exit.
     *
     * @param bridgeProperties
     *        The values of properties defined for the protocol bridge.
     *        These values can only be read, they cannot be updated by the
     *        implementation.
     * @return {@code true} if the initialization is successful and {@code
     *         false} if unsuccessful. If {@code false} is returned from an exit
     *         the protocol bridge agent will not start.
     */
    public boolean initialize(final Map<String, String> bridgeProperties);

    /**
     * Obtains a set of properties for the specified protocol server name.
     * <p>
     * The returned {@link Properties} must contain entries with key names

```

```

* corresponding to the constants defined in
* {@link ProtocolServerPropertyConstants} and in particular must include an
* entry for all appropriate constants described as required.
*
* @param protocolServerName
*         The name of the protocol server whose properties are to be
*         returned. If a null or a blank value is specified, properties
*         for the default protocol server are to be returned.
* @return The {@link Properties} for the specified protocol server, or null
*         if the server cannot be found.
*/
public Properties getProtocolServerProperties(
    final String protocolServerName);

/**
* Invoked once when a protocol bridge agent is shut down. It is intended to
* release any resources that were allocated by the exit.
*
* @param bridgeProperties
*         The values of properties defined for the protocol bridge.
*         These values can only be read, they cannot be updated by the
*         implementation.
*/
public void shutdown(final Map<String, String> bridgeProperties);
}

```

Sie können mehrere Protokoll-Server-Eigenschaften auf ähnliche Weise miteinander verketteten, dass andere Benutzerexits vorhanden sind. Die Exits werden in der Reihenfolge aufgerufen, in der sie mit der Eigenschaft `protocolBridgePropertiesExitClasses` in der Agenteneigenschaftendatei angegeben sind. Die Initialisierungsmethoden werden alle separat zurückgegeben, und wenn mindestens ein Wert 'false' zurückgegeben wird, wird der Agent nicht gestartet. Der Fehler wird im Ereignisprotokoll des Agenten dokumentiert.

Es wird nur ein Gesamtergebnis für die Methoden 'getProtocolServerProperties' aller Exits zurückgegeben. Wenn die Methode ein Eigenschaftenobjekt als Ergebniscode zurückgibt, ist dieser Wert das zurückgegebene Ergebnis, und die Methoden `getProtocolServerProperties` der nachfolgenden Exits werden nicht aufgerufen. Wenn die Methode den Wert null als Ergebniscode zurückgibt, wird die Methode "getProtocolServerProperties" des nächsten Exits aufgerufen. Wenn kein nachfolgender Exit vorhanden ist, wird das Nulresultat zurückgegeben. Ein allgemeiner Ergebniscode von null wird als Lookup-Fehler durch den Protokollbrückenagenten betrachtet.

Vorgehensweise

Führen Sie die folgenden Schritte aus, um den Exit auszuführen:

1. Kompilieren Sie den Benutzerexit für die Protokollservereigenschaften.
2. Erstellen Sie eine Java-Archivdatei (JAR-Datei) mit dem kompilierten Exit und seiner Paketstruktur.
3. Setzen Sie die JAR-Datei, die die Exitklasse enthält, im Verzeichnis `exits` des Protokollbridgeagenten. Dieses Verzeichnis befindet sich im `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name` Verzeichnis.
4. Bearbeiten Sie die Eigenschaftendatei des Protokollbrückenagenten, um die Eigenschaft `protocolBridgePropertiesExitClasses` einzuschließen. Geben Sie für den Wert dieser Eigenschaft eine durch Kommas getrennte Liste von Klassen an, die einen Benutzerexit für die Eigenschaften eines Protokoll-Bridge-Servers implementieren. Die Exitklassen werden in der Reihenfolge aufgerufen, in der sie in dieser Liste angegeben sind. Weitere Informationen finden Sie unter [Die MFT agent.properties -Datei](#).
5. Sie können die Eigenschaft `protocolBridgePropertiesConfiguration optional` angeben. Der Wert, den Sie für diese Eigenschaft angeben, wird als Zeichenfolge an die Methode "initialize ()" der von `protocolBridgePropertiesExitClasses` angegebenen Exitklassen übergeben. Weitere Informationen finden Sie unter [Die MFT agent.properties -Datei](#).

Berechtigungsnachweise für einen Dateiserver zuordnen

Die Zuordnung von Benutzerberechtigungen in Managed File Transfer zu den Benutzerberechtigungen auf dem Dateiserver kann über die Standardfunktion des Protokollbridgeagenten für Credential-Mapping oder über einen eigens geschriebenen Benutzerexit erfolgen. Managed File Transfer stellt einen Beispielbenutzerexit bereit, der die Zuordnung der Benutzerberechtigungen durchführt.

Zugehörige Konzepte

[„Die Protokollbridge“ auf Seite 303](#)

Die Protokollbridge ermöglicht in Ihrem Managed File Transfer- bzw. MFT-Netz den Zugriff auf Dateien, die in Ihrer lokalen Domäne oder remote auf einem Dateiserver außerhalb Ihres MFT-Netzes gespeichert sind. Dieser Dateiserver kann die FTP-, FTPS- oder SFTP-Netzprotokolle verwenden. Für jeden Dateiserver ist mindestens ein dedizierter Agent erforderlich. Der dedizierte Agent wird als Protokoll-Bridge-Agent bezeichnet. Ein Bridge-Agent kann mit mehreren Dateiservern interagieren.

Zugehörige Tasks

[„Berechtigungsnachweise für einen Dateiserver unter Verwendung der ProtocolBridgeCredentials.xml -Datei zuordnen“ auf Seite 313](#)

Die Zuordnung von Benutzerberechtigungen in Managed File Transfer zu den Benutzerberechtigungen auf dem Dateiserver kann über die Standardfunktion des Protokollbridgeagenten für Credential-Mapping erfolgen. In Managed File Transfer ist eine XML-Datei bereitgestellt, in der Sie die Berechtigungsinformationen eingeben können.

[„Berechtigungsnachweise für einen Dateiserver mittels Exitklassen zuordnen“ auf Seite 315](#)

Wenn Sie zur Zuordnung der Benutzerberechtigungen in Managed File Transfer zu den Benutzerberechtigungen auf dem Dateiserver nicht die Standardfunktion des Protokollbridgeagenten für Credential-Mapping verwenden möchten, können Sie hierfür einen eigenen Benutzerexit schreiben. Wenn Sie die Benutzerexits der Berechtigungsnachweiszuoordnung konfigurieren, nehmen sie die Position der standardmäßigen Berechtigungsnachweiszuoordnungsfunktion an.

[„Beispiel: Protokollbridgeagenten für die Verwendung von privaten Schlüsselberechtigungen für einen UNIX-SFTP-Server konfigurieren“ auf Seite 318](#)

Dieses Beispiel veranschaulicht, wie Sie die `ProtocolBridgeCredentials.xml`-Datei generieren und konfigurieren können. Hierbei handelt es sich um ein typisches Beispiel, die Details können also je nach Plattform variieren, die Prinzipien bleiben jedoch identisch.

Zugehörige Verweise

[ProtocolBridgeCredentialExit.java-Schnittstelle](#)

[Musterbenutzerexit für Protokoll-Bridge-Berechtigungsnachweis](#)

[Die MFT agent.properties-Datei](#)

Berechtigungsnachweise für einen Dateiserver unter Verwendung der ProtocolBridgeCredentials.xml -Datei zuordnen

Die Zuordnung von Benutzerberechtigungen in Managed File Transfer zu den Benutzerberechtigungen auf dem Dateiserver kann über die Standardfunktion des Protokollbridgeagenten für Credential-Mapping erfolgen. In Managed File Transfer ist eine XML-Datei bereitgestellt, in der Sie die Berechtigungsinformationen eingeben können.

Informationen zu diesem Vorgang

Die Datei `ProtocolBridgeCredentials.xml` muss manuell vom Benutzer erstellt werden. Standardmäßig ist die Position dieser Datei das Ausgangsverzeichnis des Benutzers, der den Protokoll-Bridge-Agenten gestartet hat. Dies kann jedoch an einer beliebigen Position im Dateisystem gespeichert werden, auf die der Agent zugreifen kann. Wenn Sie eine andere Position angeben möchten, fügen Sie das `<credentialsFile>`Element zur Datei `ProtocolBridgeProperties.xml` hinzu. Beispiel:

```
<tns:credentialsFile path="/example/path/to/ProtocolBridgeCredentials.xml"/>
```

Bevor Sie einen Protokoll-Bridge-Agenten verwenden können, konfigurieren Sie die Berechtigungsnachweiszuoordnung, indem Sie diese Datei editieren, um Host-, Benutzer- und Berechtigungsinformationen

einzuschließen. Weitere Informationen und Beispiele finden Sie im Abschnitt Format der Datei mit den Berechtigungsnachweisen für Protokollbrücken.

Vorgehensweise

1. Bearbeiten Sie die Zeile `<tns:server name="server name">`, um den Wert des Namensattributs in den Servernamen in der Datei `ProtocolBridgeProperties.xml` zu ändern.

Sie können das Musterattribut verwenden, um anzugeben, dass Sie einen Servernamen verwendet haben, der Platzhalterzeichen oder reguläre Ausdrücke enthält. Beispiel:

```
<tns:server name="serverA*" pattern="wildcard">
```

2. Fügen Sie Benutzer-ID und Berechtigungsnachweisdaten als untergeordnete Elemente von `<tns:server>` in die Datei ein.

Sie können ein oder mehrere der folgenden Elemente in die Datei einfügen:

- Wenn es sich bei dem Protokolldateiserver um einen FTP-, FTPS- oder SFTP-Server handelt, können Sie Kennwörter verwenden, um den Benutzer zu authentifizieren, der die Übertragung anfordert. Fügen Sie die folgenden Zeilen in die Datei ein:

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  serverPassword="Server Password">
</tns:user>
```

Ändern Sie anschließend den Wert der Attribute.

- `name` ist ein regulärer Java-Ausdruck, mit dem die MQMD-Benutzer-ID verglichen wird, die der MFT-Übertragungsanforderung zugeordnet ist.
- `serverUserId` ist der Wert, der als Anmeldebenutzer-ID an den Protokolldateiserver übergeben wird. Wird das Attribut `serverUserId` nicht angegeben, wird stattdessen die MQMD-Benutzer-ID der MFT-Übertragungsanforderung verwendet.
- `serverPassword` ist das Kennwort, das dem `serverUserId` zugeordnet ist.

Das Attribut `name` kann einen regulären Java-Ausdruck enthalten. Der Credential Mapper vergleicht die MQMD-Benutzer-ID der MFT-Übertragungsanforderung mit diesem regulären Ausdruck. Der Protokollbridgeagent versucht, die MQMD-Benutzer-ID mit dem regulären Ausdruck im Namensattribut der `<tns:user>`-Elemente in der Reihenfolge abzugleichen, in der die Elemente in der Datei vorhanden sind. Wenn eine Übereinstimmung gefunden wird, wird der Protokoll-Bridge-Agent nicht nach weiteren Übereinstimmungen suchen. Wenn eine Übereinstimmung gefunden wird, werden die entsprechenden Werte für `serverUserId` und `serverPassword` als Anmeldebenutzer-ID und Kennwort an den Protokolldateiserver übergeben. Bei den MQMD-Benutzer-ID-IDs muss die Groß-/Kleinschreibung beachtet werden.

- Wenn es sich bei dem Protokolldateiserver um einen SFTP-Server handelt, können Sie öffentliche und private Schlüssel verwenden, um den Benutzer zu authentifizieren, der die Übertragung anfordert. Fügen Sie die folgenden Zeilen in die Datei ein, und ändern Sie den Wert der Attribute. Das Element `<tns:user>` kann ein oder mehrere `<tns:privateKey>`-Elemente enthalten.

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  hostKey="Host Key">
  <tns:privateKey associationName="association"
    keyPassword="Private key password">
    Private key file text
  </tns:privateKey>
</tns:user>
```

- `name` ist ein regulärer Java-Ausdruck, mit dem die MQMD-Benutzer-ID verglichen wird, die der MFT-Übertragungsanforderung zugeordnet ist.

- `serverUserId` ist der Wert, der als Anmeldebenutzer-ID an den Protokolldateiserver übergeben wird. Wird das Attribut `serverUserId` nicht angegeben, wird stattdessen die MQMD-Benutzer-ID der MFT-Übertragungsanforderung verwendet.
- `hostKey` ist der erwartete Schlüssel, der beim Anmelden vom Server zurückgegeben wird.
- `key` ist der private Schlüssel von `serverUserId`.
- `keyPassword` ist das Kennwort für den Schlüssel, um öffentliche Schlüssel zu generieren.
- `associationName` ist ein Wert, der zum Identifizieren von Trace- und Protokollierungszwecken verwendet wird.

Das Attribut `name` kann einen regulären Java-Ausdruck enthalten. Der Credential Mapper vergleicht die MQMD-Benutzer-ID der MFT-Übertragungsanforderung mit diesem regulären Ausdruck. Der Protokollbridgeagent versucht, die MQMD-Benutzer-ID mit dem regulären Ausdruck im Namensattribut der `<tns:user>`-Elemente in der Reihenfolge abzugleichen, in der die Elemente in der Datei vorhanden sind. Wenn eine Übereinstimmung gefunden wird, wird der Protokoll-Bridge-Agent nicht nach weiteren Übereinstimmungen suchen. Wird eine Übereinstimmung gefunden, werden die Werte für `serverUserId` und `key` für die Authentifizierung des MFT-Benutzers auf dem Protokolldateiserver verwendet. Bei den MQMD-Benutzer-ID-IDs muss die Groß-/Kleinschreibung beachtet werden.

Weitere Informationen zur Verwendung privater Schlüssel mit einem Protokollbridgeagenten finden Sie im Abschnitt [„Beispiel: Protokollbridgeagenten für die Verwendung von privaten Schlüsselberechtigungenachweisen für einen UNIX-SFTP-Server konfigurieren“](#) auf Seite 318.

Anmerkung:

Wenn die Übertragungsanforderung in die Befehlswarteschlange geschrieben wird und sich die Befehlswarteschlange des Agenten auf einem z/OS- oder IBM i-System befindet, wird die MQMD-Benutzer-ID unter Umständen in Großbuchstaben umgewandelt. Dies hat zur Folge, dass die MQMD-Benutzer-ID für denselben Ursprungsbenutzer im ursprünglichen Fall an den Exit für Berechtigungsnachweise angelangt oder in Abhängigkeit vom Quellenagenten, der in der Übertragungsanforderung angegeben ist, in Großbuchstaben konvertiert wird. Der Exit für die Berechtigungsnachweiszuordnung führt Groß-/Kleinschreibung für die angegebene MQMD-Benutzer-ID aus, die Sie möglicherweise in der Zuordnungsdatei zulassen müssen.

Zugehörige Verweise

[Dateiformat der Protokollbrückenberechtigungsachweise](#)

[Format der Protokollbrückeneigenschaftendatei](#)

[Von MFT verwendete reguläre Ausdrücke](#)

Berechtigungsachweise für einen Dateiserver mittels Exitklassen zuordnen

Wenn Sie zur Zuordnung der Benutzerberechtigungen in Managed File Transfer zu den Benutzerberechtigungen auf dem Dateiserver nicht die Standardfunktion des Protokollbridgeagenten für Credential-Mapping verwenden möchten, können Sie hierfür einen eigenen Benutzerexit schreiben. Wenn Sie die Benutzerexits der Berechtigungsachweiszuordnung konfigurieren, nehmen sie die Position der standardmäßigen Berechtigungsachweiszuordnungsfunktion an.

Informationen zu diesem Vorgang

Managed File Transfer stellt einen Beispielbenutzerexit bereit, der die Zuordnung der Benutzerberechtigungsachweise durchführt. Weitere Informationen finden Sie unter [„Beispielbenutzerexit für Protokollbridge-Berechtigungsachweise verwenden“](#) auf Seite 317.

Ein Benutzerexit für die Zuordnung von Protokoll-Bridge-Berechtigungsachweisen muss eine der folgenden Schnittstellen implementieren:

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`, mit dem ein Protokoll-Bridge-Agent Dateien an einen Standardprotokolldateiserver und von einem Standardprotokolldateiserver übertragen kann

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2` ermöglicht es Ihnen, Dateien an und von mehreren Endpunkten zu übertragen.

Die `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`-Schnittstelle enthält die gleiche Funktion wie `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` und enthält auch erweiterte Funktionen. Weitere Informationen finden Sie unter [ProtocolBridgeCredentialExit.java interface](#) und [ProtocolBridgeCredentialExit2.java interface](#).

Die Berechtigungsnachweisexits können in ähnlicher Weise wie andere Benutzerexits miteinander verketten werden. Die Exits werden in der Reihenfolge aufgerufen, in der sie mit der Eigenschaft `protocolBridgeCredentialConfiguration` in der Agenteneigenschaftendatei angegeben sind. Die Initialisierungsmethoden werden alle separat zurückgegeben, und wenn mindestens ein Wert 'false' zurückgegeben wird, wird der Agent nicht gestartet. Der Fehler wird im Ereignisprotokoll des Agenten dokumentiert.

Es wird nur ein Gesamtergebnis für die `mapMQUserId`-Methoden aller Exits wie folgt zurückgegeben:

- Wenn die Methode den Wert `USER_SUCCESSFULLY_MAPPED` oder `USER_DENIED_ACCESS` als Ergebniscode zurückgibt, ist dieser Wert das zurückgegebene Ergebnis, und die `mapMQUserId`-Methoden der nachfolgenden Exits werden nicht aufgerufen.
- Wenn die Methode den Wert `NO_MAPPING_FOUND` als Ergebniscode zurückgibt, wird die Methode 'mqMQUserId' des nächsten Ausgangs aufgerufen.
- Wenn kein weiterer Exit vorhanden ist, wird das Ergebnis `NO_MAPPING_FOUND` zurückgegeben.
- Ein allgemeiner Ergebniscode von `USER_DENIED_ACCESS` oder `NO_MAPPING_FOUND` wird als Übertragungsfehler durch den Bridge-Agenten betrachtet.

Führen Sie die folgenden Schritte aus, um den Exit auszuführen:

Vorgehensweise

1. Kompilieren Sie den Benutzerexit für die Berechtigungsnachweisfunktion für Protokollbrücken.
2. Erstellen Sie eine Java-Archivdatei (JAR-Datei) mit dem kompilierten Exit und seiner Paketstruktur.
3. Stellen Sie die JAR-Datei, die die Exitklasse enthält, in das Verzeichnis `exits` des Bridgeagenten. Das Verzeichnis befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.
4. Bearbeiten Sie die Eigenschaftendatei des Protokollbrückenagenten, um die Eigenschaft `protocolBridgeCredentialExitClasses` einzuschließen. Geben Sie für den Wert dieser Eigenschaft eine durch Kommas getrennte Liste von Klassen an, die eine Exitroutine für die Protokollbrückenberechtigungsnachweise implementieren. Die Exitklassen werden in der Reihenfolge aufgerufen, in der sie in dieser Liste angegeben sind. Weitere Informationen finden Sie unter [Die MFT agent.properties-Datei](#).
5. Bearbeiten Sie die Eigenschaftendatei des Protokollbrückenagenten, um Folgendes zu berücksichtigen:

```
exitClassPath=IBM MQ installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

Die `agent.properties`-Datei für einen Agenten befindet sich in Ihrem `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_agent_name`-Verzeichnis.

Wenn Sie die `agent.properties`-Datei ändern, müssen Sie den Agenten erneut starten, um die Änderungen zu übernehmen.

6. Sie können die Eigenschaft `protocolBridgeCredentialConfiguration optional` angeben. Der Wert, den Sie für diese Eigenschaft angeben, wird als Zeichenfolgeobjekt an die Methode "initialize()" der von `protocolBridgeCredentialExitClasses` angegebenen Exitklassen übergeben. Weitere Informationen finden Sie unter [Die MFT agent.properties-Datei](#).
7. Starten Sie den Protokollbridgeagenten mit dem Befehl **fteStartAgent**.

Beispielbenutzerexit für Protokollbridge-Berechtigungsachweise verwenden

Managed File Transfer stellt einen Beispielbenutzerexit bereit, der die Zuordnung der Benutzerberechtigungsachweise durchführt.

Informationen zu diesem Vorgang

Im `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` -Verzeichnis und im Thema [Beispielbenutzerexit für Berechtigungsachweise für Protokollbridge-Bridgewird ein Beispiexit für die Berechtigungsachweisbrücke für Protokollbridge bereitgestellt](#). Dieses Beispiel basiert auf der `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` -Schnittstelle.

Der `Exit SampleCredentialExit.java` liest eine Eigenschaftendatei, in der die MQMD-Benutzer-IDs, die Übertragungsanforderungen zugeordnet sind, Serverbenutzer-IDs und Serverkennwörtern zugeordnet werden. Die Position der Eigenschaftendatei wird aus der Eigenschaft `protocolBridgeCredentialConfiguration` des Protokollbrückenagenten übernommen.

Führen Sie die folgenden Schritte aus, um den Musterbenutzerexit auszuführen:

Vorgehensweise

1. Kompilieren Sie die `SampleCredentialExit.java` -Datei.
2. Erstellen Sie eine JAR-Datei, die den kompilierten Exit und die zugehörige Paketstruktur enthält.
3. Stellen Sie die JAR-Datei in das Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/exits`.
4. Bearbeiten Sie die `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` -Datei so, dass sie die folgende Zeile enthält:

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

5. Bearbeiten Sie die Eigenschaftendatei des Protokollbrückenagenten, um Folgendes zu berücksichtigen:

```
exitClassPath=IBM MQ installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

Die `agent.properties` -Datei für einen Agenten befindet sich in Ihrem `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` -Verzeichnis.

Wenn Sie die `agent.properties` -Datei ändern, müssen Sie den Agenten erneut starten, um die Änderungen zu übernehmen.

6. Erstellen Sie eine Eigenschaftendatei für Berechtigungsachweise (`credentials.properties`) im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` und bearbeiten Sie sie, um Einträge in das Format einzuschließen:

```
mqUserId=serverUserId,serverPassword
```

7. Bearbeiten Sie die `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` -Datei so, dass sie die folgende Zeile enthält:

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

Sie müssen den absoluten Pfad zur `credentials.properties` -Datei verwenden.

8. Starten Sie den Protokollbridgeagenten mit dem Befehl **fteStartAgent**.

Zugehörige Konzepte

„Die Protokollbridge“ auf Seite 303

Die Protokollbridge ermöglicht in Ihrem Managed File Transfer- bzw. MFT-Netz den Zugriff auf Dateien, die in Ihrer lokalen Domäne oder remote auf einem Dateiserver außerhalb Ihres MFT-Netztes gespeichert sind. Dieser Dateiserver kann die FTP-, FTPS- oder SFTP-Netzprotokolle verwenden. Für jeden Dateiserver ist mindestens ein dedizierter Agent erforderlich. Der dedizierte Agent wird als Protokoll-Bridge-Agent bezeichnet. Ein Bridge-Agent kann mit mehreren Dateiservern interagieren.

Zugehörige Verweise

[ProtocolBridgeCredentialExit.java-Schnittstelle](#)

[ProtocolBridgeCredentialExit2.java \(Schnittstelle\)](#)

[Musterbenutzerexit für Protokoll-Bridge-Berechtigungs-nachweis](#)

Die MFT `agent.properties`-Datei

[fteCreateBridgeAgent \(MFT-Protokollbridgeagenten erstellen und konfigurieren\)](#)

Beispiel: Protokollbridgeagenten für die Verwendung von privaten Schlüsselberechtigungs-nachweisen für einen UNIX-SFTP-Server konfigurieren

Dieses Beispiel veranschaulicht, wie Sie die `ProtocolBridgeCredentials.xml`-Datei generieren und konfigurieren können. Hierbei handelt es sich um ein typisches Beispiel, die Details können also je nach Plattform variieren, die Prinzipien bleiben jedoch identisch.

Informationen zu diesem Vorgang

Vorgehensweise

1. Generieren Sie einen öffentlichen und privaten Schlüssel für die Authentifizierung beim SFTP-Server.

Auf einem Linux -Hostsystem können Sie beispielsweise das als Teil des Pakets 'openssh' bereitgestellte Tool **ssh-keygen** verwenden, um das Paar aus öffentlichem und privatem Schlüssel zu erstellen.

Standardmäßig fordert der Befehl **ssh-keygen** ohne Argumente eine Position und eine Kennphrase für die beiden Schlüsseldateien an, wobei standardmäßig die folgenden Namen verwendet werden:

```
id_rsa      <-- Private key
id_rsa.pub  <-- Public key
```



Achtung: Wenn Sie den Befehl **ssh-keygen** aus einer aktuellen Version von OpenSSH verwenden, z. B. aus der mit RHEL 8 bereitgestellten Version, ist das verwendete Schlüsselformat nicht mit dem Protokollbridgeagenten kompatibel und Übertragungsversuche zum SFTP-Server schlagen mit der folgenden Nachricht fehl:

```
BFGBR0216E: Authentication to protocol server 'sftp.host.address' failed because of invalid private key.
```

Um einen kompatiblen privaten Schlüssel mit diesen neueren Versionen von OpenSSH zu erstellen, geben Sie das Schlüsselformat mit dem folgenden Argument für den Befehl **ssh-keygen** an:

```
ssh-keygen -m PEM
```

Der Inhalt des privaten `id_rsa`-Schlüssels enthält dann die erste und letzte Zeile von:

```
-----BEGIN RSA PRIVATE KEY-----
.....
-----END RSA PRIVATE KEY-----
```

die mit dem Protokollbridgeagenten kompatibel ist.

2. Kopieren Sie den gesamten Inhalt der Datei `id_rsa.pub` in die Datei `~/.ssh/authorized_keys` des SFTP-Benutzers auf dem SFTP-Server.

Stellen Sie sicher, dass die Dateiberechtigungen für diese Datei und das Verzeichnis `~/ .ssh` für den SFTP-Server entsprechend festgelegt wurden, um die Schlüsselauthentifizierung zuzulassen. Diese Berechtigungen sind in der Regel:

```
~/ .ssh          Mode 700
~/ .ssh/authorized_keys  Mode 600
```

3. Für Managed File Transfer muss ein SSH-Fingerabdruck des Host mit dem MD5-Algorithmus generiert werden. Führen Sie einen der folgenden Befehle aus, um den SSH-Fingerabdruck des Host vom SFTP-Server anzufordern.

- Für Red Hat® Enterprise Linux Version 6.x und früher und Linux Ubuntu 14.04 führen Sie den folgenden Befehl aus:

```
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
```

- Ab Red Hat Enterprise Linux 7.x, Linux, Ubuntu 16.04 und SuSE Linux 12.4 generiert der Befehl `ssh-keygen` standardmäßig den ssh-Fingerabdruck mit dem SHA256-Algorithmus. Führen Sie den folgenden Befehl aus, um den SSH-Fingerabdruck mit dem MD5-Algorithmus zu generieren:

```
ssh-keygen -l -E MD5 -f /etc/ssh/ssh_host_rsa_key.pub
```

Die Ausgabe des Befehls ähnelt dem folgenden Beispiel:

```
2048 MD5:64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 no comment (RSA)
```

Extrahieren Sie den hexadezimalen Teil nur der Ausgabe, die als Host-Key in der `ProtocolBridgeCredentials.xml`-Datei verwendet werden soll (siehe Schritt „4“ auf Seite 319). Daher würden Sie in diesem Beispiel `64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27` extrahieren.

4. Bearbeiten Sie auf dem Protokoll-Bridge-Agenten-System die `ProtocolBridgeCredentials.xml`-Datei. Ersetzen Sie die im folgenden Beispiel kursiv dargestellten Werte durch ihre eigenen Werte:

```
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
ProtocolBridgeCredentials.xsd ">

<tns:agent name="Agent_name">

<tns:server name="SFTP_name">

<tns:user name="mq_User_ID" serverUserId="SFTP_user_ID"
  hostKey="ssh_host_finger">
<tns:privateKey associationName="name" keyPassword="pass_phrase">
  Complete contents of the id_rsa file including the entries
  -----BEGIN RSA PRIVATE KEY-----

  -----END RSA PRIVATE KEY-----
</tns:privateKey>
</tns:user>

</tns:server>
</tns:agent>
</tns:credentials>
```

Dabei gilt:

- *Agentenname* steht für den Namen des Protokollbridgeagenten.
- *SFTP_hostname* ist der Name des SFTP-Servers, der in der `ProtocolBridgeProperties.xml`-Datei angezeigt wird.
- *MQ-Benutzer-ID* ist die der Übertragungsanforderung zugeordnete MQMD-Benutzer-ID.
- *SFTP_benutzer-id* ist die SFTP-Benutzer-ID, die in Schritt 2 verwendet wird. Es ist der Wert, der an den SFTP-Service übergeben wird, der als Anmeldebenutzer-ID verwendet wird.
- *SSH-Hostfingerabdruck* ist der in Schritt 3 erfasste Fingerabdruck.

- *Name* ist ein Name, den Sie für Tracing- und Protokollierungszwecke angeben können.
- *Kennphrase* ist die im Befehl 'ssh-keygen' in Schritt 1 angegebene Kennphrase.
- *Vollständiger Inhalt der id_rsa-Datei* ist der vollständige Inhalt der generierten id_rsa -Datei aus Schritt 1. Um einen Verbindungsfehler zu verhindern, stellen Sie sicher, dass Sie beide der folgenden Einträge enthalten:

```
-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----
```

Sie können zusätzliche Schlüssel hinzufügen, indem Sie das Element <tns:privatekey> duplizieren.

5. Starten Sie den Protokollbridgeagenten, sofern er noch nicht gestartet ist. Alternativ dazu fragt der Protokollbridgeagent in regelmäßigen Abständen die ProtocolBridgeCredentials.xml -Datei ab und holt die Änderungen ab.

Zugehörige Verweise

[Dateiformat der Protokollbrückenberechtigungsnachweise](#)

[fteCreateBridgeAgent \(MFT-Protokollbridgeagenten erstellen und konfigurieren\)](#)

Die MFT agent.properties-Datei

[„Berechtigungsnachweise für einen Dateiserver zuordnen“ auf Seite 313](#)

Die Zuordnung von Benutzerberechtigungen in Managed File Transfer zu den Benutzerberechtigungen auf dem Dateiserver kann über die Standardfunktion des Protokollbridgeagenten für Credential-Mapping oder über einen eigens geschriebenen Benutzerexit erfolgen. Managed File Transfer stellt einen Beispielbenutzerexit bereit, der die Zuordnung der Benutzerberechtigungs-nachweise durchführt.

Protokollbridge für einen FTPS-Server konfigurieren

Konfigurieren Sie einen FTPS-Server auf ähnliche Weise wie einen FTP-Server: Erstellen Sie einen Bridge-Agenten für den Server, definieren Sie die Servereigenschaften und ordnen Sie die Benutzerberechtigungs-nachweise zu.

Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen FTPS-Server zu konfigurieren:

Vorgehensweise

1. Erstellen Sie mit dem Befehl **fteCreateBridgeAgent** einen Protokollbridgeagenten für den FTPS-Server. Die Parameter, die auf FTP anwendbar sind, gelten auch für FTPS, aber es gibt auch drei erforderliche Parameter für FTPS:
 - a) Der Parameter **-bt** Geben Sie FTPS als Wert für diesen Parameter an.
 - b) Der Parameter **-bts** für die Truststore-Datei. Der Befehl setzt voraus, dass nur die Serverauthentifizierung erforderlich ist, und Sie müssen die Position der Truststore-Datei angeben.

Die explizite Form des FTPS-Protokolls wird standardmäßig mit dem Befehl **fteCreateBridgeAgent** konfiguriert, aber Sie können die implizite Form konfigurieren, indem Sie die Eigenschaftendatei der Protokollbridge ändern. Die Protokoll-Bridge stellt immer eine Verbindung zu FTPS-Servern im passiven Modus her.

Weitere Informationen zum Befehl **fteCreateBridgeAgent** finden Sie im Abschnitt [fteCreateBridgeAgent \(MFT-Protokollbridgeagenten erstellen und konfigurieren\)](#).

Wenn Sie Anweisungen zur Erstellung von Truststore-Dateien benötigen, lesen Sie die Informationen zum Keytool in der [Dokumentation zu Oracle keytool](#).

2. Definieren Sie die FTPS-Servereigenschaften in einem Element <ftpsServer> in der Eigenschaftendatei für die Protokollbridge: ProtocolBridgeProperties.xml. Weitere Informationen finden Sie unter [„Eigenschaften für Protokolldateiserver mit der ProtocolBridgeProperties.xml -Datei definieren“ auf](#)

Seite 306. Sie können die Clientauthentifizierung auch aktivieren, indem Sie die Eigenschaftendatei der Protokoll-Bridge bearbeiten. Ausführliche Informationen zu allen Konfigurationsoptionen finden Sie im Abschnitt [Dateiformat der Protokollbrückeneigenschaften](#).

3. Die Zuordnung von Benutzerberechtigungen in Managed File Transfer zu Benutzerberechtigungen auf dem FTPS-Server kann über die Standardfunktion für Credential-Mapping des Protokollbridgeagenten oder über einen benutzerdefinierten Benutzerexit erfolgen. Weitere Informationen finden Sie unter [„Berechtigungsnachweise für einen Dateiserver zuordnen“](#) auf Seite 313.
4. Standardmäßig ist die Truststore-Datei mit dem JKS-Format konfiguriert. Wenn Sie das Format ändern möchten, bearbeiten Sie die Eigenschaftendatei der Protokoll-Bridge.

Beispiel

Im Folgenden sehen Sie einen Beispieleintrag für einen FTPS-Server in der Eigenschaftendatei der Protokoll-Bridge:

```
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">
  <tns:defaultServer name="ftpserver.mycompany.com" />

  <tns:ftpsServer name="ftpserver.mycompany.com" host="ftpserver.mycompany.com" port="990" plat
form="windows"
  timeZone="Europe/London" locale="en_US" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false"
  trustStore="c:\mydirec\truststore.jks" />

  <!-- Define servers here -->
</tns:serverProperties>
```

Nächste Schritte

Informationen zu den Teilen des FTPS-Protokolls, die unterstützt werden und die nicht unterstützt werden, finden Sie in der [FTPS-Serverunterstützung durch die Protokollbrücke](#).

Zugehörige Konzepte

[„Die Protokollbridge“](#) auf Seite 303

Die Protokollbridge ermöglicht in Ihrem Managed File Transfer- bzw. MFT-Netz den Zugriff auf Dateien, die in Ihrer lokalen Domäne oder remote auf einem Dateiserver außerhalb Ihres MFT-Netztes gespeichert sind. Dieser Dateiserver kann die FTP-, FTPS- oder SFTP-Netzprotokolle verwenden. Für jeden Dateiserver ist mindestens ein dedizierter Agent erforderlich. Der dedizierte Agent wird als Protokoll-Bridge-Agent bezeichnet. Ein Bridge-Agent kann mit mehreren Dateiservern interagieren.

Zugehörige Tasks

[„Berechtigungsnachweise für einen Dateiserver unter Verwendung der ProtocolBridgeCredentials.xml -Datei zuordnen“](#) auf Seite 313

Die Zuordnung von Benutzerberechtigungen in Managed File Transfer zu den Benutzerberechtigungen auf dem Dateiserver kann über die Standardfunktion des Protokollbridgeagenten für Credential-Mapping erfolgen. In Managed File Transfer ist eine XML-Datei bereitgestellt, in der Sie die Berechtigungsinformationen eingeben können.

[„Eigenschaften für Protokolldateiserver mit der ProtocolBridgeProperties.xml -Datei definieren“](#) auf Seite 306

Definieren Sie die Eigenschaften eines oder mehrerer Protokolldateiserver, auf die Sie Dateien übertragen möchten, und verwenden Sie dazu die `ProtocolBridgeProperties.xml` -Datei, die von Managed File Transfer im Konfigurationsverzeichnis des Agenten bereitgestellt wird.

Zugehörige Verweise

[fteCreateBridgeAgent \(MFT-Protokollbridgeagenten erstellen und konfigurieren\)](#)

[Dateiformat der Protokollbrückenberechtigungs-nachweise](#)

[Format der Protokollbrückeneigenschaftendatei](#)

[FTPS-Serverunterstützung durch die Protokollbridge](#)

Szenarios und Beispiele für die Begrenzung der Anzahl der Dateiübertragungen auf einzelne Dateiserver

Wie der überarbeitete Protokollbridge-Agent mit den Attributen **maxActiveDestinationTransfers** und **failTransferWhenCapacityReached** arbeitet, zusammen mit einigen Beispielen.

Szenarios, in denen das Arbeiten des Protokollbridge-Agenten auf Basis des Werts **maxActiveDestinationTransfers** angezeigt wird

Szenario 1

Die Datei `ProtocolBridgeProperties.xml` für einen Protokollbridgeagenten enthält zwei Dateiserverdefinitionen:

- Sie haben das globale Attribut **maxActiveDestinationTransfers** nicht festgelegt.
- Sie haben das Attribut **maxActiveDestinationTransfers** nicht sowohl für `fileServerA` als auch für `fileServerB` festgelegt.
- Sie haben das Attribut **maxDestinationTransfers** des Protokollbridgeagenten auf den Standardwert gesetzt.

Wenn Sie das Attribut **maxDestinationTransfers** des Protokollbridgeagenten auf den Standardwert 25 gesetzt haben, gehen Sie wie folgt vor:

- Der Zielagent beginnt mit der Verarbeitung von zwei verwalteten Übertragungen an `fileServerA`.
- Die beiden Übertragungen werden abgeschlossen.

Zu diesem Zeitpunkt erkennt der Client, dass `fileServerA` fehlgeschlagen ist, und legt die folgenden Werte für `fileServerA` in der Datei `ProtocolBridgeProperties.xml` fest:

```
maxActiveDestinationTransfers = 0  
failTransferWhenCapacityReached =true
```

- Eine weitere Übertragung wird für `fileServerA` und einige für `fileServerB` empfangen:
Basierend auf den Eigenschaften, die im vorherigen Schritt festgelegt wurden, wird die verwaltete Übertragung an `fileServerA` zurückgewiesen und als fehlgeschlagen markiert, während die Übertragungen für `fileServerB` im Standarddatenfluss bearbeitet werden.
- Nach einiger Zeit erkennt der Client, dass `fileServerA` erneut ausgeführt wird, so dass der Client den zuvor hinzugefügten Wert in der `ProtocolBridgeProperties.xml` entfernt oder auskommentiert. Eine neue verwaltete Übertragung kommt für `fileServerA` an und wird im Standarddatenfluss bearbeitet.

Szenario 2

- Sie haben das Attribut **maxActiveDestinationTransfers** für einen Dateiserver festgelegt und das Attribut **failTransferWhenCapacityReached** nicht festgelegt.
- Der Protokollbridge-Agent agiert als Zielagent für diese Anzahl von verwalteten Übertragungen an den Dateiserver.
- Der Wert des Attributs **maxActiveDestinationTransfers** wird um 1 reduziert.

Der Protokollbridge-Agent aktualisiert seine Konfiguration dynamisch und setzt **maxActiveDestinationTransfers** auf den neuen Wert, während er noch aktiv ist. Die in Bearbeitung befindlichen verwalteten Übertragungen sind von dieser Aktualisierung nicht betroffen und dürfen nicht abgeschlossen werden.

Szenario 3

Die Datei `ProtocolBridgeProperties.xml` für einen Protokollbridgeagenten enthält zwei Dateiserverdefinitionen:

- Sie haben das globale Attribut **maxActiveDestinationTransfers** nicht festgelegt.
- Sie haben das Attribut **failTransferWhenCapacityReached** nicht festgelegt.
- Sie haben **maxActiveDestinationTransfers** unter `fileServerA` auf 1 gesetzt.
- Sie haben das Attribut **maxActiveDestinationTransfers** nicht für `fileServerB` festgelegt.

Wenn für den Protokollbridgeagenten das Attribut **maxDestinationTransfers** auf 5 gesetzt ist:

- Die maximale Anzahl aktiver Zielübertragungen vom Protokollbridgeagenten auf `fileServerA` ist 1 (obwohl der Zielagent über 5 Zielübertragungsbereiche verfügt, kann nur einer für verwaltete Übertragungen an `fileServerA` verwendet werden).

Dies ist hilfreich, wenn `fileServerA` fehlschlägt. Sobald `fileServerA` erneut ausgeführt wird, kann der Wert von **maxActiveDestinationTransfers** auf 5 erhöht werden, um die volle Kapazität der zulässigen Zielübertragungen zu ermöglichen.

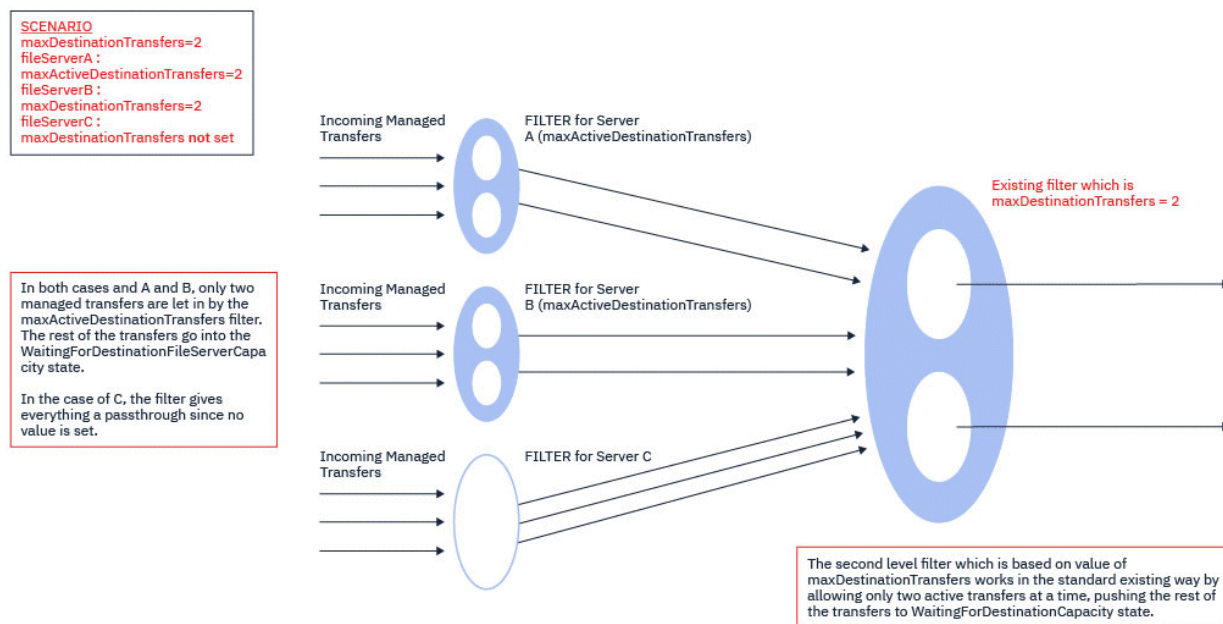
- Die maximale Anzahl aktiver Zielübertragungen vom Protokollbridgeagenten zu `fileServerB` ist 5.

Da **maxActiveDestinationTransfers** nicht für diesen Dateiserver festgelegt ist, kann der Protokollbridgeagent alle seiner 5 Zielübertragungsbereiche für verwaltete Transfers an ihn verwenden.

Szenario 4

Im folgenden Diagramm:

- Sie haben das Attribut **maxDestinationTransfers** in der Datei `agent.properties` auf 2 gesetzt.
- Sie haben **maxActiveDestinationTransfers** unter `fileServerA` auf 2 gesetzt.
- Sie haben das Attribut **maxActiveDestinationTransfers** unter `fileServerB` auf 2 gesetzt.
- Sie haben das Attribut **maxActiveDestinationTransfers** nicht für `fileServerC` festgelegt.



Wie das Diagramm zeigt, sind die Attribute **maxActiveDestinationTransfers** und **maxDestinationTransfers** unabhängig voneinander.

Die Werte für **maxActiveDestinationTransfers** für jeden der Server werden überprüft. Basierend auf diesem Wert können die Übertragungen entweder weiter fortgesetzt werden oder in den Status **WaitingForDestinationFileServerCapacity** verschoben werden.

Die Übertragungen, die zulässig sind, werden dann durch den vorhandenen Standardfluss der Prüfung für die **maxDestinationTransfers** übergeben.

Szenario 5



Achtung: Sie sollten vorsichtig sein, wenn Sie die Werte für die Attribute von **maxActiveDestinationTransfers** festlegen, da Sie den Wert des Attributs **maxDestinationTransfers** im Auge behalten müssen.

Wenn Sie dies nicht tun, kann eine Situation wie im folgenden Text beschrieben auftreten:

- Sie haben keinen Wert für das globale Attribut **maxActiveDestinationTransfers** festgelegt.
- Sie haben in der Datei `agent.properties` den Wert **maxDestinationTransfers**= 2 festgelegt.
- Sie haben den Wert **maxActiveDestinationTransfers**= 2 unter `fileServerA` festgelegt.
- Sie haben keinen Wert für **maxActiveDestinationTransfers** on `fileServerB` festgelegt.

Angenommen, die folgende Abfolge von Ereignissen tritt auf:

- Der Protokollbridge-Agent empfängt eine Anforderung zum Übertragen einer Datei an `fileServerA`. Der Protokollbridge-Agent führt derzeit keine Aktion aus, daher akzeptiert er diese verwaltete Übertragungsanforderung.

Die Übertragungsbereiche sehen jetzt wie folgt aus:

- Zielübertragungen: 1
 - Zielübertragungen für `fileServerA`: 1
 - Zielübertragungen für `fileServerB`: 0
- Jetzt empfängt der Protokollbridge-Agent eine weitere Anforderung, als Zielagent für eine verwaltete Übertragung mit `fileServerA` zu fungieren. Erneut akzeptiert er diese Anforderung, und so sehen die Übertragungsbereiche wie folgt aus:
 - Zielübertragungen: 2
 - Zielübertragungen für `fileServerA`: 2
 - Zielübertragungen für `fileServerB`: 0

Die beiden `Destination Transfer` -Slots im Agenten sind nun belegt, sodass der Agent erst an weiteren verwalteten Übertragungen teilnehmen kann, wenn eine der Übertragungen an `fileServerA` abgeschlossen ist.

- Eine kurze Zeit später schlägt `fileServerA` fehl, was dazu führt, dass die beiden verwalteten Übertragungen in die Wiederherstellung gehen. Die von diesen verwalteten Übertragungen verwendeten `Destination transfer` -Zeitfenster bleiben während dieser Zeit im Gebrauch.
- Als Nächstes empfängt der Protokollbridge-Agent eine Anforderung zum Übertragen einer Datei an `fileServerB`. Es gibt einen Platz für diese Übertragung in den `Destination Transfers for fileServerB` -Slots, aber alle `Destination Transfer` Slots für den Agenten werden verwendet, sodass die Übertragung in den Rückstand gestellt wird, sodass sie später wiederholt werden kann.

Daher wird die Übertragung an `fileServerB` blockiert, bis mindestens eine der Übertragungen an `fileServerA` abgeschlossen ist und den Slot `Destination Transfer` freigegeben hat.

Gehen Sie wie folgt vor, um diese Situation zu verhindern:

- Setzen Sie entweder den Wert von **maxActiveDestinationTransfers** auf den Dateiservern auf einen kleineren Wert als den Wert von **maxDestinationTransfers**, damit freie Bereiche verbleiben.
- Oder verteilen Sie den Wert des Attributs **maxActiveDestinationTransfers** gleichmäßig auf alle Endpunktserver.

Verhalten des Protokollbridgeagenten auf der Basis der Werte des Attributs **maxActiveDestinationTransfers**

Anmerkung: In allen in der folgenden Tabelle aufgeführten Fehlerfällen geht der Protokollbridge-Agent davon aus, dass dieses Attribut nicht gesetzt ist, wenn das Attribut **maxActiveDestinationTransfers** auf einen Wert gesetzt ist, der nicht gültig ist.

maxActiveDestinationTransfers	Beispielwert	Beschreibung
Nicht festgelegt	Nicht festgelegt	Die Übertragungen laufen wie üblich ab. Für die Anzahl der Übertragungen für den Endpunkt *ftp* gibt es keine Begrenzung.
Angegeben	0	Für diesen spezifischen *ftp*-Endpunkt sind keine Übertragungen zulässig.
Negativer Wert	-1	Fehler protokolliert in 'output0.log'. Wert -1 ist für eine nicht negative Ganzzahl nicht gültig. Der Protokollbridge-Agent geht davon aus, dass das Attribut nicht gesetzt ist.
Nicht ganzzahliger Wert	abc	Fehler protokolliert in 'output0.log'. Der Wert abc ist für eine Ganzzahl nicht gültig. Der Protokollbridge-Agent geht davon aus, dass das Attribut nicht gesetzt ist.
Leer	""	Der Wert ' ' des Attributs 'maxActiveDestinationTransfers' ist für eine nicht negative Ganzzahl nicht gültig.
Angegeben	5	Ermöglicht nur fünf aktive Übertragungen, die zu einem beliebigen Zeitpunkt für diesen *ftp*-Endpunkt ausgeführt werden. Die überhöhten Übertragungen werden basierend auf dem Wert des Attributs 'failTransferWhenCapacityReached' wiederholt oder zurückgewiesen.

Verhalten des Protokollbridgeagenten für die Kombination der Attribute **maxActiveDestinationTransfers** und **failTransferWhenCapacityReached**

Wert 'failTransferWhenCapacityReached'	Wert 'maxActiveDestinationTransfers'	Ergebnis
Falsch	3	Es sind drei aktive Übertragungen für diesen Endpunktserver zulässig. Alle weiteren Übertragungen werden wiederholt.
True	3	Es sind drei aktive Übertragungen für diesen Endpunktserver zulässig. Alle weiteren Übertragungen werden zurückgewiesen und als fehlgeschlagen markiert.
Nicht festgelegt	3	Der Standardwert false (falsch) für 'failTransferWhenCapacityReached' wird berücksichtigt. Das Ergebnis ist, dass drei aktive Übertragungen für diesen Endpunktserver zulässig sind. Alle weiteren Übertragungen werden wiederholt.

Wert 'failTransferWhenCapacityReached'	Wert 'maxActiveDestinationTransfers'	Ergebnis
Andere Werte als boolesche Werte	Angegeben	Fehler in 'output.log' protokolliert. Der für 'failTransferWhenCapacityReached' angegebene Wert ist kein boolescher Wert. Der Standardwert für 'failTransferEncoding CapacityReached' wird berücksichtigt.

Verhalten des Protokollbridgeagenten für die Kombination der Attribute **maxDestinationTransfers** und **failTransferWhenCapacityReached**

Wert 'failTransferWhenCapacityReached'	Wert 'maxDestinationTransfers'	Ergebnis
True	10	Wenn die Anzahl der gleichzeitig aktiven Übertragungen den Wert 10 erreicht, ist die verwaltete 11 ^t -Übertragung durch den Protokollbridgeagenten fehlgeschlagen.
Falsch	10	Bestehendes Verhalten. Wenn die Anzahl der gleichzeitig aktiven Übertragungen 10 erreicht, wird die 11 verwaltete Th -Übertragung in die Warteschlange gestellt und wartet darauf, dass ein Slot freigegeben wird.
Nicht festgelegt	10	Bestehendes Verhalten

Fehlermeldungen

Bestehende Nachricht:

BFGS0082I

Wird in der Ausgabedatei 'output0.log' des Quellenagenten protokolliert, wenn der Protokollbridge-Agent die Übertragung ablehnt und bereits die maximale Anzahl an Übertragungen ausführt, die im Attribut **maxDestinationTransfers** definiert sind.

Neue Nachrichten:

BFGSS0085I

Wird in der Ausgabedatei 'output0.log' des Quellenagenten protokolliert, wenn der Protokollbridge-Agent eine verwaltete Übertragung ablehnt und erneut versucht.

BFGSS0086I

Wird in der Ausgabedatei 'output0.log' des Quellenagenten protokolliert, wenn der Protokollbridge-Agent eine verwaltete Übertragung zurückweist und erneut versucht und das Zielelement nicht den Namen des Dateiservers enthält.

BFGSS0084E

Wird im Explorer und in der Datei 'audit.xml' protokolliert, wenn der Protokollbridge-Agent zurückweist, um die maximale Anzahl gleichzeitiger Übertragungen, die im Attribut **maxActiveDestina-**

tionTransfers angegeben sind, zu überschreiten, und markiert eine verwaltete Übertragung als fehlgeschlagen.

BFGSS0087E

Wird im Explorer und in der Datei 'audit.xml' protokolliert, wenn der Protokollbridge-Agent zurückweist, um die maximale Anzahl an Zielübertragungen, die im Attribut **maxActiveDestinationTransfers** angegeben sind, zu überschreiten, und markiert eine verwaltete Übertragung als fehlgeschlagen.

BFGSS0088W

Wird in der Datei 'output0.log' protokolliert, wenn der Wert des Attributs **maxActiveDestinationTransfers** den Wert des Attributs **maxDestinationTransfers** überschreitet.

BFGSS0089I

Wird in der Datei 'output0.log' des Zielprotokollbridge-Agenten protokolliert, wenn er einen Quellenagenten verwendet, der nicht auf IBM MQ 9.3.0 oder höher ausgeführt wird.

Zugehörige Konzepte

„Die Protokollbridge“ auf Seite 303

Die Protokollbridge ermöglicht in Ihrem Managed File Transfer- bzw. MFT-Netz den Zugriff auf Dateien, die in Ihrer lokalen Domäne oder remote auf einem Dateiserver außerhalb Ihres MFT-Netztes gespeichert sind. Dieser Dateiserver kann die FTP-, FTPS- oder SFTP-Netzprotokolle verwenden. Für jeden Dateiserver ist mindestens ein dedizierter Agent erforderlich. Der dedizierte Agent wird als Protokoll-Bridge-Agent bezeichnet. Ein Bridge-Agent kann mit mehreren Dateiservern interagieren.

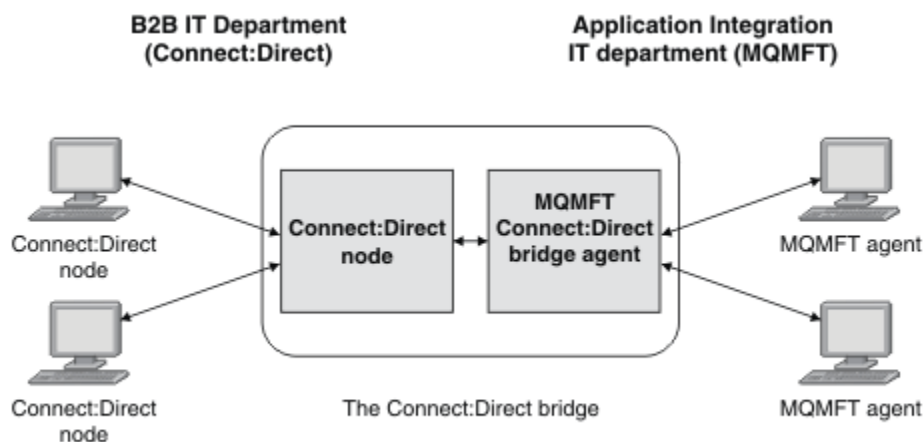
Zugehörige Tasks

„Eigenschaften für Protokolldateiserver mit der ProtocolBridgeProperties.xml -Datei definieren“ auf Seite 306

Definieren Sie die Eigenschaften eines oder mehrerer Protokolldateiserver, auf die Sie Dateien übertragen möchten, und verwenden Sie dazu die ProtocolBridgeProperties.xml -Datei, die von Managed File Transfer im Konfigurationsverzeichnis des Agenten bereitgestellt wird.

Connect:Direct-Bridge

Sie können Dateien in ein und aus einem IBM Sterling Connect:Direct-Netz übertragen. Mit der Connect:Direct-Bridge, einer Komponente von Managed File Transfer, können Sie Dateien zwischen MFT und IBM Sterling Connect:Direct übertragen.




Das Diagramm zeigt eine MFT Connect:Direct -Bridge zwischen zwei Abteilungen, der IT-Abteilung von B2B und der IT-Abteilung Application Integration. Die IT-Abteilung von B2B verwendet Connect:Direct, um Dateien von und von den Geschäftspartnern des Unternehmens zu übertragen. In der Application Integration-IT-Abteilung wird IBM MQ als Messaging-Infrastruktur eingesetzt, daher wurde Managed File Transfer als Lösung für Dateiübertragungen ausgewählt.

Über die MFT-Connect:Direct-Bridge können die beiden Abteilungen Dateien zwischen dem Connect:Direct-Netz in der B2B-IT-Abteilung und dem MFT-Netz in der Application Integration-IT-Abteilung austauschen. Die Connect:Direct-Bridge ist eine Komponente von Managed File Transfer, zu der ein MFT-Agent gehört, der mit einem Connect:Direct-Knoten kommuniziert. Der MFT-Agent ist für Übertragungen an den und aus dem Connect:Direct-Knoten bestimmt und wird als Connect:Direct-Bridgeagent bezeichnet.

Die Connect:Direct-Bridge steht als Bestandteil der Service- und der Agentenkomponente von Managed File Transfer zur Verfügung und kann für folgende Aufgaben eingesetzt werden:

1. Mit Managed File Transfer-Befehlen können Sie die Übertragung einer oder mehrerer Dateien von einem MFT-Agenten an einen Connect:Direct-Knoten starten.
2. Mit Managed File Transfer-Befehlen können Sie die Übertragung einer oder mehrerer Dateien von einem Connect:Direct-Knoten an einen MFT-Agenten starten.
3. Mit Managed File Transfer-Befehlen können Sie eine Dateiübertragung einleiten, bei der ein benutzerdefinierter Connect:Direct-Prozess gestartet wird.
4. Mit dem Connect:Direct-Prozess können Sie eine MFT-Dateiübertragungsanforderung übergeben.

Eine Connect:Direct-Bridge kann nur Dateien an oder von Connect:Direct-Knoten übertragen. Die Connect:Direct-Bridge kann Dateien nur dann zu oder von ihrem lokalen Dateisystem übertragen, wenn die Übertragung von einem Connect:Direct-Prozess übergeben wurde.

 Mit der Connect:Direct-Bridge können Sie Übertragungen in ein oder aus einem Dataset durchführen, das sich in einem Connect:Direct-Knoten auf einem z/OS-System befindet. Das Verhalten dieses Verfahrens unterscheidet sich in einigen Punkten von dem von Datasetübertragungen, bei denen nur Managed File Transfer-Agenten zum Einsatz kommen. Weitere Informationen finden Sie im Abschnitt

 [Datasets zu und von Connect:Direct-Knoten übertragen.](#)

Unterstützte Plattformen

Die Connect:Direct-Bridge setzt sich aus einem MFT Connect:Direct-Bridgeagenten und einem Connect:Direct-Knoten zusammen. Der Agent wird unter Windows und Linux für x86-64 unterstützt. Der Knoten wird auf den Plattformen unterstützt, die Unterstützung für IBM Sterling Connect:Direct for Windows und IBM Sterling Connect:Direct for UNIX bieten. Anweisungen zum Erstellen des Connect:Direct-Bridgeagenten und zur Konfiguration eines Connect:Direct-Knotens für den Agenten, mit dem Sie kommunizieren möchten, finden Sie im Abschnitt [Connect:Direct-Bridge konfigurieren](#).

Über die Connect:Direct-Bridge können Dateien in und aus Connect:Direct-Knoten übertragen werden, die Bestandteil einer Serviceinstallation unter Connect:Direct for Windows, Connect:Direct for UNIX oder Connect:Direct for z/OS sind. Details zu den unterstützten Connect:Direct-Versionen finden Sie auf der Webseite [System Requirements for IBM MQ](#).

Der Agent und der Knoten, die zusammen die Connect:Direct-Bridge bilden, müssen sich auf demselben System befinden oder Zugriff auf dasselbe Dateisystem (beispielsweise über einen gemeinsam genutzten NFS-Mount) haben. Dieses Dateisystem wird verwendet, um Dateien während Dateiübertragungen, die die Connect:Direct-Bridge einbeziehen, temporär in einem Verzeichnis zu speichern, das durch den Parameter **cdTmpDir** definiert ist. Der Connect:Direct-Bridgeagent und der Connect:Direct-Bridgeknoten müssen auf dieses Verzeichnis unter Verwendung desselben Pfadnamens zugreifen können. Wenn sich Agent und Knoten beispielsweise jeweils auf einem eigenen Windows-System befinden, muss zum Anhängen des gemeinsam genutzten Dateisystems für beide Systeme derselbe Laufwerksbuchstabe verwendet werden. Bei den folgenden Konfigurationen können der Agent und der Knoten denselben Pfadnamen verwenden:

- Agent und Knoten befinden sich auf demselben System, das unter Windows oder Linux for x86-64 betrieben wird.
- Der Agent befindet sich auf einem System mit Linux for x86-64, der Knoten auf einem AIX-System.
- Der Agent befindet sich auf einem Windows-System, der Knoten auf einem anderen Windows-System.

Bei den folgenden Konfigurationen können der Agent und der Knoten nicht denselben Pfadnamen verwenden:

- Der Agent befindet sich auf einem System mit Linux for x86-64, der Knoten auf einem Windows-System.
- Der Agent befindet sich auf einem Windows-System, der Knoten auf einem UNIX-System.

Diese Einschränkungen sollten Sie bei der Planung der Connect:Direct-Bridgeinstallation bedenken.

Zugehörige Konzepte

„Wiederherstellung und Neustart für Übertragungen an und aus Connect:Direct-Knoten“ auf Seite 336
 Unter Umständen kann Managed File Transfer während einer Übertragung keine Verbindung zum IBM Sterling Connect:Direct-Knoten herstellen (wenn beispielsweise der Knoten plötzlich nicht mehr zur Verfügung steht). In diesem Fall versucht Managed File Transfer, die Übertragung wiederherzustellen, oder die Übertragung schlägt mit einer Fehlernachricht fehl.

„Benutzerdefinierten Connect:Direct-Prozess aus einer Dateiübertragungsanforderung übergeben“ auf Seite 337

Sie können eine Dateiübertragungsanforderung für eine Übertragung über den Connect:Direct-Bridgeagenten übergeben, die als Teil der Dateiübertragung einen benutzerdefinierten Connect:Direct-Prozess aufruft.

„Connect:Direct-Prozesse zur Übergabe von Managed File Transfer-Übertragungsanforderungen verwenden“ auf Seite 342

Die Übergabe einer Übertragungsanforderung an den Connect:Direct-Bridgeagenten kann aus einem Connect:Direct-Prozess erfolgen. Hierzu bietet Managed File Transfer Befehle, die aus einer **RUN TASK**-Anweisung eines Connect:Direct-Prozesses aufgerufen werden können.

Zugehörige Tasks

Connect:Direct-Bridge konfigurieren

„Datei an einen Connect:Direct-Knoten übertragen“ auf Seite 330

Sie können eine Datei mithilfe der Connect:Direct -Bridge von einem Managed File Transfer -Agenten an einen Connect:Direct -Knoten übertragen. Geben Sie einen Connect:Direct-Knoten als Ziel der Übertragung an, indem Sie den Connect:Direct -Bridgeagenten als Zielagenten angeben und die Zieldatei in der Form `connect_direct_node_name:file_path` angeben.

„Datei aus einem Connect:Direct-Knoten übertragen“ auf Seite 331

Sie können eine Datei über die Connect:Direct -Bridge von einem Connect:Direct -Knoten an einen Managed File Transfer Agent übertragen. Sie können einen Connect:Direct-Knoten als Quelle für die Übertragung angeben, indem Sie den Connect:Direct-Bridgeagenten als Quellenagenten angeben und die Quellenspezifikation in der Form `connect_direct_node_name:file_path` angeben.

„Mehrere Dateien an einen Connect:Direct-Knoten übertragen“ auf Seite 332

Sie können mehrere Dateien über die Connect:Direct -Bridge von einem Managed File Transfer Agent auf einen Connect:Direct -Knoten übertragen. Wenn Sie einen Connect:Direct-Knoten als Ziel der mehrfachen Dateiübertragung verwenden möchten, geben Sie den Connect:Direct -Bridgeagenten als Zielagenten an und geben Sie das Zielverzeichnis in der Form `connect_direct_node_name:directory_path` an.

„Transferring multiple files from a Connect:Direct node“ auf Seite 334

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

„Mithilfe von Platzhalterzeichen mehrere Dateien an Connect:Direct übertragen“ auf Seite 335

Wenn Sie mehrere Dateien von einem Managed File Transfer -Agenten an einen Connect:Direct -Knoten übertragen möchten, verwenden Sie die Connect:Direct -Bridge. Sie können Platzhalterzeichen in der Quellenspezifikation verwenden, die Sie für den Befehl **fteCreateTransfer** angeben. Wie bei allen Managed File Transfer-Übertragungen unter Angabe von Platzhalterzeichen kann nur der letzte Teil des Dateipfads Platzhalterzeichen enthalten. Beispiel: `/abc/def*` ist ein gültiger Datenpfad und `/abc*/def` ist nicht gültig.

Fehlerbehebung bei der Connect:Direct-Bridge

Zugehörige Verweise

fteCreateCDAgent: Connect:Direct-Bridgeagenten erstellen

Datei an einen Connect:Direct-Knoten übertragen

Sie können eine Datei mithilfe der Connect:Direct -Bridge von einem Managed File Transfer -Agenten an einen Connect:Direct -Knoten übertragen. Geben Sie einen Connect:Direct-Knoten als Ziel der Übertragung an, indem Sie den Connect:Direct -Bridgeagenten als Zielagenten angeben und die Zieldatei in der Form `connect_direct_node_name:file_path` angeben.

Vorbereitende Schritte

Vor einer Dateiübertragung müssen Sie zunächst die Connect:Direct-Bridge, eine Komponente von Managed File Transfer, konfigurieren. Weitere Informationen finden Sie im Abschnitt [Connect:Direct-Bridge konfigurieren](#).

Informationen zu diesem Vorgang

In diesem Beispiel wird der Connect:Direct-Bridgeagent als 'CD_BRIDGE' bezeichnet. Der Quellenagent wird als FTE_AGENT bezeichnet und kann jede beliebige Version von WMQFTE sein. Der Connect:Direct-Zielknoten heißt CD_NODE1. Die Datei, die übertragen werden soll, befindet sich auf dem Dateipfad /home/helen/file.log auf dem System, auf dem sich FTE_AGENT befindet. Die Datei wird in den Dateipfad /files/data.log auf dem System übertragen, auf dem CD_NODE1 ausgeführt wird.

Vorgehensweise

1. Verwenden Sie den Übertragungsbefehl `fteCreate` mit dem Wert für den Parameter **-df** (Zieldatei) im Format `connect_direct_node_name:file_path` und dem Wert des Parameters **-da** (Zielagent), der als Name des Connect:Direct -Bridgeagenten angegeben ist.

Anmerkung: Der Connect:Direct-Knoten, der von `connect_direct_node_name` angegeben wird, ist der Knoten, an den die Datei übertragen werden soll, nicht der Connect:Direct -Knoten, der als Teil der Connect:Direct -Bridge arbeitet.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
-df CD_NODE1:/files/data.log /home/helen/file.log
```

Weitere Informationen finden Sie unter [fteCreateTransfer: Neue Dateiübertragung starten](#).

2. Der Quellenagent FTE_AGENT überträgt die Datei an den Connect:Direct-Bridgeagenten CD_BRIDGE. Die Datei wird vorübergehend auf dem System gespeichert, auf dem der Connect:Direct-Bridgeagent ausgeführt wird, und zwar an der Position, die über die Agenteneigenschaft 'cdTmpDir' definiert ist. Der Connect:Direct-Bridgeagent überträgt die Datei an den Connect:Direct-Knoten CD_NODE1.

Zugehörige Konzepte

[„Connect:Direct-Bridge“](#) auf Seite 327

Sie können Dateien in ein und aus einem IBM Sterling Connect:Direct-Netz übertragen. Mit der Connect:Direct-Bridge, einer Komponente von Managed File Transfer, können Sie Dateien zwischen MFT und IBM Sterling Connect:Direct übertragen.

Zugehörige Tasks

[„Datei aus einem Connect:Direct-Knoten übertragen“](#) auf Seite 331

Sie können eine Datei über die Connect:Direct -Bridge von einem Connect:Direct -Knoten an einen Managed File Transfer Agent übertragen. Sie können einen Connect:Direct-Knoten als Quelle für die Übertragung angeben, indem Sie den Connect:Direct-Bridgeagenten als Quellenagenten angeben und die Quellenspezifikation in der Form `connect_direct_node_name:file_path` angeben.

Zugehörige Verweise

[Die MFT agent.properties-Datei](#)

Datei aus einem Connect:Direct-Knoten übertragen

Sie können eine Datei über die Connect:Direct -Bridge von einem Connect:Direct -Knoten an einen Managed File Transfer Agent übertragen. Sie können einen Connect:Direct-Knoten als Quelle für die Übertragung angeben, indem Sie den Connect:Direct-Bridgeagenten als Quellenagenten angeben und die Quellenspezifikation in der Form `connect_direct_node_name:file_path` angeben.

Vorbereitende Schritte

Vor einer Dateiübertragung müssen Sie zunächst die Connect:Direct-Bridge, eine Komponente von Managed File Transfer, konfigurieren. Weitere Informationen finden Sie im Abschnitt [Connect:Direct-Bridge konfigurieren](#).

Informationen zu diesem Vorgang

In diesem Beispiel wird der Connect:Direct-Bridgeagent als 'CD_BRIDGE' bezeichnet. Der Zielagent hat den Namen 'FTE_AGENT' und kann in einer beliebigen Version von Managed File Transfer vorliegen. Der Connect:Direct-Quellknoten heißt 'CD_NODE1'. Die Datei, die übertragen werden soll, befindet sich auf dem Dateipfad `/home/brian/in.file` auf dem System, auf dem sich CD_NODE1 befindet. Die Datei wird in den Dateipfad `/files/out.file` auf dem System übertragen, auf dem FTE_AGENT ausgeführt wird.

Vorgehensweise

Verwenden Sie den Befehl **fteCreateTransfer** mit dem Wert für die Quellenspezifikation im Format `connect_direct_node_name:file_path` und dem Wert des Parameters **-sa**, der als Name des Connect:Direct -Bridgeagenten angegeben ist.

Anmerkung: Der Connect:Direct-Knoten, der von `connect_direct_node_name` angegeben wird, ist der Knoten, von dem die Datei übertragen werden soll, nicht der Connect:Direct -Knoten, der als Teil der Connect:Direct -Bridge arbeitet. For example:

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT
                 -df /files/out.file CD_NODE1:/home/brian/in.file
```

Weitere Informationen finden Sie unter [fteCreateTransfer: Neue Dateiübertragung starten](#).

Ergebnisse

Der Connect:Direct-Bridgeagent CD_BRIDGE fordert die Datei vom Connect:Direct-Knoten CD_NODE1 an. Der Connect:Direct-Knoten sendet die Datei an die Connect:Direct-Bridge. Während der Übertragung der Datei aus dem Connect:Direct-Knoten speichert die Connect:Direct-Bridge die Datei vorübergehend an der Speicherposition, die über die Agenteneigenschaft `cdTmpDir` angegeben ist. Sobald die Übertragung der Datei vom Connect:Direct-Knoten an die Connect:Direct-Bridge abgeschlossen ist, sendet die Connect:Direct-Bridge die Datei an den Zielagenten FTE_AGENT und löscht sie dann aus dem temporären Verzeichnis.

Zugehörige Konzepte

„Connect:Direct-Bridge“ auf Seite 327

Sie können Dateien in ein und aus einem IBM Sterling Connect:Direct-Netz übertragen. Mit der Connect:Direct-Bridge, einer Komponente von Managed File Transfer, können Sie Dateien zwischen MFT und IBM Sterling Connect:Direct übertragen.

Zugehörige Verweise

Die [MFT agent.properties-Datei](#)

Dataset an einen Connect:Direct-Knoten unter z/OS übertragen

Sie können ein Dataset von einem Managed File Transfer -Agenten unter z/OS auf einen Connect:Direct -Knoten unter z/OS übertragen, indem Sie eine Connect:Direct -Bridge verwenden, die sich auf einem Windows -oder Linux -System befindet.

Vorbereitende Schritte

Vor einer Dateiübertragung müssen Sie zunächst die Connect:Direct-Bridge, eine Komponente von Managed File Transfer, konfigurieren. Weitere Informationen finden Sie im Abschnitt [Connect:Direct-Bridge konfigurieren](#).

Informationen zu diesem Vorgang

In diesem Beispiel wird der Parameter **-df** verwendet, um die Zieladresse der Übertragung anzugeben. Der Parameter **-df** kann bei einem Quellenagenten mit einer beliebigen Version von Managed File Transfer verwendet werden. Sie können stattdessen den Parameter **-ds** verwenden. Bei dem Quellenagenten handelt es sich um einen Managed File Transfer-Agenten mit der Bezeichnung 'FTE_ZOS1'. Der Connect:Direct-Bridgeagent mit der Bezeichnung 'CD_BRIDGE' befindet sich auf einem Linux-System. Der Connect:Direct-Zielknoten heißt 'CD_ZOS2'. Quellenagent und Connect:Direct-Zielknoten befinden sich beide auf z/OS-Systemen. Der zu übertragene Datensatz befindet sich in //FTEUSER.SOURCE.LIB auf dem System, auf dem sich FTE_ZOS1 befindet. Der Datensatz wird an die Datei //CDUSER.DEST.LIB auf dem System übertragen, auf dem sich CD_ZOS2 befindet.

Vorgehensweise

1. Verwenden Sie den Übertragungsbefehl `fteCreateTransfer` mit dem Wert des Parameters **-df** im Format `connect_direct_node_name:data_set_name;attributes` und dem Wert des Parameters **-da** (Zielagent), der als Name des Connect:Direct -Bridgeagenten angegeben ist.

Der Connect:Direct -Knoten, der von `connect_direct_node_name` angegeben wird, ist der Knoten, an den die Datei übertragen werden soll, nicht der Connect:Direct -Knoten, der als Teil der Connect:Direct -Bridge arbeitet.

Der von `data_set_name` angegebene Name der Dateigruppe muss absolut, nicht relativ sein. Connect:Direct stellt dem Dataset-Namen kein Benutzernamenspräfix voran.

```
fteCreateTransfer -sa FTE_ZOS1 -sm QM_ZOS
                 -da CD_BRIDGE -dm QM_BRIDGE
                 -df CD_ZOS2: //'CDUSER.DEST.LIB;BLKSIZE(8000);LRECL(80)'
                 //'FTEUSER.SOURCE.LIB'
```

Weitere Informationen finden Sie unter **fteCreateTransfer**: Neue Dateiübertragung starten.

2. Der Quellenagent FTE_ZOS1 überträgt die Daten des Dataset an den Connect:Direct-Bridgeagenten CD_BRIDGE. Die Daten werden vorübergehend als Flatfile auf dem System gespeichert, auf dem der Connect:Direct-Bridgeagent ausgeführt wird, und zwar an der durch die Agenteneigenschaft 'cdTmpDir' definierten Position. Der Connect:Direct-Bridgeagent überträgt die Daten an den Connect:Direct-Knoten CD_ZOS2. Nach Abschluss der Übertragung wird die Flatfile von dem System, auf dem der Connect:Direct-Bridgeagent ausgeführt wird, gelöscht.

Zugehörige Konzepte

„Connect:Direct-Bridge“ auf Seite 327

Sie können Dateien in ein und aus einem IBM Sterling Connect:Direct-Netz übertragen. Mit der Connect:Direct-Bridge, einer Komponente von Managed File Transfer, können Sie Dateien zwischen MFT und IBM Sterling Connect:Direct übertragen.

Zugehörige Tasks

[Übertragen von Datensätzen an und von Connect: Direct-Knoten](#)

Zugehörige Verweise

[BPXWDYN-Eigenschaften, die Sie nicht mit MFT verwenden dürfen](#)

Mehrere Dateien an einen Connect:Direct-Knoten übertragen

Sie können mehrere Dateien über die Connect:Direct -Bridge von einem Managed File Transfer Agent auf einen Connect:Direct -Knoten übertragen. Wenn Sie einen Connect:Direct-Knoten als Ziel der mehrfachen Dateiübertragung verwenden möchten, geben Sie den Connect:Direct -Bridgeagenten als Zielagenten an und geben Sie das Zielverzeichnis in der Form `connect_direct_node_name:directory_path`

Vorbereitende Schritte

Vor einer Dateiübertragung müssen Sie die Connect:Direct-Bridge, eine Komponente von Managed File Transfer, konfigurieren. Weitere Informationen finden Sie im Abschnitt [Connect:Direct-Bridge konfigurieren](#).

Informationen zu diesem Vorgang

In diesem Beispiel wird der Quellenagent FTE_AGENT genannt. Der Connect:Direct-Bridgeagent wird heißt 'CD_BRIDGE'. Der Connect:Direct-Zielknoten heißt CD_NODE1. Die Daten, die übertragen werden sollen, sind /home/jack/data.log, /logs/log1.txt, and /results/latest auf dem System, wo sich FTE_AGENT befindet. Die Dateien werden in das Verzeichnis /in/files auf dem System übertragen, auf dem CD_NODE1 ausgeführt wird.

Vorgehensweise

Verwenden Sie den Befehl `fteCreateTransfer` mit dem Wert für den Parameter **-dd** (Zielverzeichnis) in der Form `connect_direct_node_name:directory_path`. Geben Sie den Wert des Parameters **-da** (Zielagent) als Name des Connect:Direct-Bridgeagenten an.

Anmerkung: Der Connect:Direct Knoten angegeben durch `connect_direct_node_name` ist der Knoten zu dem die Dateien übertragen werden sollen, nicht der Connect:Direct Knoten, der als Teil der Connect:Direct Bridge funktioniert.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/in/files /home/jack/data.log
                  /logs/log1.txt /results/latest
```

Weitere Informationen finden Sie unter [fteCreateTransfer: Neue Dateiübertragung starten](#).

Ergebnisse

Der Quellenagent FTE_AGENT überträgt die erste Datei an den Connect:Direct-Bridgeagenten CD_BRIDGE. Der Connect:Direct -Bridgeagent speichert die Datei vorübergehend an der Position, die durch die Eigenschaft 'cdTmpDir' definiert wird. Sobald die Datei vollständig vom Quellenagenten an die Connect:Direct-Bridge übertragen wurde, sendet der Connect:Direct-Bridgeagent die Datei an den Connect:Direct-Knoten, der über die Agenteneigenschaft 'cdNode' definiert wird. Dieser Knoten sendet die Datei an den Connect:Direct-Zielknoten CD_NODE1. Der Connect:Direct-Bridgeagent löscht die Datei aus dem temporären Verzeichnis, sobald die Übertragung zwischen den beiden Connect:Direct-Knoten abgeschlossen ist. Dieser Prozess wird für jede angegebene Quellendatei wiederholt.

Zugehörige Konzepte

[„Connect:Direct-Bridge“ auf Seite 327](#)

Sie können Dateien in ein und aus einem IBM Sterling Connect:Direct-Netz übertragen. Mit der Connect:Direct-Bridge, einer Komponente von Managed File Transfer, können Sie Dateien zwischen MFT und IBM Sterling Connect:Direct übertragen.

Zugehörige Tasks

[„Datei an einen Connect:Direct-Knoten übertragen“ auf Seite 330](#)

Sie können eine Datei mithilfe der Connect:Direct -Bridge von einem Managed File Transfer -Agenten an einen Connect:Direct -Knoten übertragen. Geben Sie einen Connect:Direct-Knoten als Ziel der Übertragung an, indem Sie den Connect:Direct -Bridgeagenten als Zielagenten angeben und die Zieldatei in der Form `connect_direct_node_name:file_path` angeben.

[„Mithilfe von Platzhalterzeichen mehrere Dateien an Connect:Direct übertragen“ auf Seite 335](#)

Wenn Sie mehrere Dateien von einem Managed File Transfer -Agenten an einen Connect:Direct -Knoten übertragen möchten, verwenden Sie die Connect:Direct -Bridge. Sie können Platzhalterzeichen in der Quellenspezifikation verwenden, die Sie für den Befehl **fteCreateTransfer** angeben. Wie bei allen Managed File Transfer-Übertragungen unter Angabe von Platzhalterzeichen kann nur der letzte Teil des Dateipfads Platzhalterzeichen enthalten. Beispiel: /abc/def* ist ein gültiger Datenpfad und /abc*/def ist nicht gültig.

[„Datei aus einem Connect:Direct-Knoten übertragen“ auf Seite 331](#)

Sie können eine Datei über die Connect:Direct -Bridge von einem Connect:Direct -Knoten an einen Managed File Transfer Agent übertragen. Sie können einen Connect:Direct-Knoten als Quelle für die Übertragung angeben, indem Sie den Connect:Direct-Bridgeagenten als Quellenagenten angeben und die Quellenspezifikation in der Form `connect_direct_node_name:file_path` angeben.

„Transferring multiple files from a Connect:Direct node“ auf Seite 334

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

Zugehörige Verweise

Die `MFT agent.properties`-Datei

Transferring multiple files from a Connect:Direct node

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

Before you begin

Before transferring a file, you must configure the Connect:Direct bridge, which is a component of Managed File Transfer. See [Configuring the Connect:Direct bridge](#).

About this task

In this example, the Connect:Direct bridge agent is called `CD_BRIDGE`. The destination agent is called `FTE_Z`, and is running on a z/OS system. The source Connect:Direct node is called `CD_NODE1`. The files to be transferred are located at the file paths `/in/file1`, `/in/file2`, and `/in/file3` on the system where `CD_NODE1` is located. The files are transferred to the partitioned data set `//OBJECT.LIB` on the system where `FTE_Z` is running.

Procedure

Use the `fteCreateTransfer` command with the values for the source specifications in the form `connect_direct_node_name:file_path` and the value of the `-sa` parameter specified as the name of the Connect:Direct bridge agent.

Note: The Connect:Direct node specified by `connect_direct_node_name` is the node that you want the files to be transferred from, not the Connect:Direct node that operates as part of the Connect:Direct bridge.

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_Z
                  -dp '//OBJECT.LIB' CD_NODE1:/in/file1
                  CD_NODE1:/in/file2 CD_NODE1:/in/file3
```

For more information, see [fteCreateTransfer: start a new file transfer](#).

Results

The Connect:Direct bridge agent `CD_BRIDGE` requests the first file from the Connect:Direct node `CD_NODE1`. The Connect:Direct node sends the file to the Connect:Direct bridge. While the file is being transferred from the Connect:Direct node, the Connect:Direct bridge stores the file temporarily in the location defined by the `cdTmpDir` agent property. When the file has finished transferring from the Connect:Direct node to the Connect:Direct bridge, the Connect:Direct bridge sends the file to the destination agent `FTE_Z` and then deletes the file from the temporary location. This process is repeated for each specified source file.

Related concepts

[“Connect:Direct-Bridge” on page 327](#)

Sie können Dateien in ein und aus einem IBM Sterling Connect:Direct-Netz übertragen. Mit der Connect:Direct-Bridge, einer Komponente von Managed File Transfer, können Sie Dateien zwischen MFT und IBM Sterling Connect:Direct übertragen.

Related reference

[The MFT agent.properties file](#)

Mithilfe von Platzhalterzeichen mehrere Dateien an Connect:Direct übertragen

Wenn Sie mehrere Dateien von einem Managed File Transfer -Agenten an einen Connect:Direct -Knoten übertragen möchten, verwenden Sie die Connect:Direct -Bridge. Sie können Platzhalterzeichen in der Quellenspezifikation verwenden, die Sie für den Befehl **fteCreateTransfer** angeben. Wie bei allen Managed File Transfer-Übertragungen unter Angabe von Platzhalterzeichen kann nur der letzte Teil des Dateipfads Platzhalterzeichen enthalten. Beispiel: `/abc/def*` ist ein gültiger Datenpfad und `/abc*/def` ist nicht gültig.

Vorbereitende Schritte

Vor einer Dateiübertragung müssen Sie zunächst die Connect:Direct-Bridge, eine Komponente von Managed File Transfer, konfigurieren. Weitere Informationen finden Sie im Abschnitt [Connect:Direct-Bridge konfigurieren](#).

Informationen zu diesem Vorgang

In diesem Beispiel wird der Quellenagent als FTE_AGENT bezeichnet und der Connect:Direct-Bridgeagent heißt CD_BRIDGE. Der Connect:Direct-Zielknoten heißt CD_NODE1. Die Dateien, die übertragen werden sollen, befinden sich im Verzeichnis `/reports` auf dem System, auf dem sich FTE_AGENT befindet. Nur Dateien mit Namen, die mit `report` beginnen, gefolgt mit zwei Buchstaben und dem Suffix `.log`, werden übertragen. Beispiel: Die Datei `/reports/report01.log` wird übertragen, aber die Datei `/reports/report1.log` wird nicht übertragen. Die Dateien werden in das Verzeichnis `/home/fred` auf dem System übertragen, auf dem CD_NODE1 ausgeführt wird.

Vorgehensweise

1. Verwenden Sie den Befehl `fteCreateTransfer` mit dem Wert für den Parameter **-dd** (Zielverzeichnis) in der Form `connect_direct_node_name:directory_path`. Geben Sie für den Wert des Parameters **-da** (Zielagent) den Connect:Direct-Bridgeagenten an.

Anmerkung: Der Connect:Direct Knoten angegeben durch `connect_direct_node_name` ist der Knoten zu dem die Dateien übertragen werden sollen, nicht der Connect:Direct Knoten, der als Teil der Connect:Direct Bridge funktioniert.

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                 -dd CD_NODE1:/home/fred "/reports/report??.log"
```

Weitere Informationen finden Sie unter [fteCreateTransfer: Neue Dateiübertragung starten](#).

2. Der Quellenagent FTE_AGENT überträgt die erste Datei, die mit dem Muster `/reports/report??.log` übereinstimmt, an den Connect:Direct-Bridgeagenten CD_BRIDGE. Der Connect:Direct-Bridgeagent speichert die Datei vorübergehend an der Position, die durch die Eigenschaft 'cdTmpDir' definiert wird. Sobald die Datei vollständig vom Quellenagenten an die Connect:Direct-Bridge übertragen wurde, sendet der Connect:Direct-Bridgeagent die Datei an den Connect:Direct-Knoten, der über die Agenteneigenschaft 'cdNode' definiert wird. Dieser Knoten sendet die Datei an den Connect:Direct-Zielknoten CD_NODE1. Der Connect:Direct-Bridgeagent löscht die Datei aus dem temporären Verzeichnis, sobald die Übertragung zwischen den beiden Connect:Direct-Knoten abgeschlossen ist. Dieser Prozess wird für jede Quelldatei wiederholt, die mit dem Platzhaltermuster übereinstimmt. `/reports/report??.log`.

Anmerkung: Die Liste der Dateien, die mit dem Muster `/reports/report??.log` übereinstimmen, hängt von dem Betriebssystem des Systems ab, auf dem sich der Quellenagent FTE_AGENT befindet.

- Wenn sich der Quellenagent auf einem Windows-System befindet, wird die Groß-/Kleinschreibung beim Musterabgleich nicht beachtet. Das Muster stimmt mit allen Dateien im Verzeichnis `/reports` mit dem Dateinamen des Formulars `report` überein, gefolgt von zwei Zeichen und einem Suffix von `.log`, unabhängig davon, in welchem Fall die Buchstaben in der Datei enthalten sind. Beispiel: `Report99.log` ist eine Übereinstimmung.
- Wenn sich der Quellenagent auf einem Linux- oder UNIX-System befindet, wird die Groß-/Kleinschreibung beim Musterabgleich beachtet. Das Muster entspricht nur den Dateien im Verzeichnis `/reports` mit dem Dateinamen des Formulars `report`, gefolgt von zwei Zeichen und einem Suffix von `.log`. Beispiel: `reportAB.log` ist eine Übereinstimmung, aber `treportAB.LOG` und `Report99.log` stimmen nicht überein.

Zugehörige Konzepte

„[Connect:Direct-Bridge](#)“ auf Seite 327

Sie können Dateien in ein und aus einem IBM Sterling Connect:Direct-Netz übertragen. Mit der `Connect:Direct-Bridge`, einer Komponente von Managed File Transfer, können Sie Dateien zwischen MFT und IBM Sterling Connect:Direct übertragen.

Zugehörige Tasks

[Platzhalterzeichen mit MFT verwenden](#)

„[Datei an einen Connect:Direct-Knoten übertragen](#)“ auf Seite 330

Sie können eine Datei mithilfe der `Connect:Direct -Bridge` von einem Managed File Transfer -Agenten an einen `Connect:Direct -Knoten` übertragen. Geben Sie einen `Connect:Direct-Knoten` als Ziel der Übertragung an, indem Sie den `Connect:Direct -Bridgeagenten` als Zielagenten angeben und die Zieldatei in der Form `connect_direct_node_name:file_path` angeben.

„[Mehrere Dateien an einen Connect:Direct-Knoten übertragen](#)“ auf Seite 332

Sie können mehrere Dateien über die `Connect:Direct -Bridge` von einem Managed File Transfer Agent auf einen `Connect:Direct -Knoten` übertragen. Wenn Sie einen `Connect:Direct-Knoten` als Ziel der mehrfachen Dateiübertragung verwenden möchten, geben Sie den `Connect:Direct -Bridgeagenten` als Zielagenten an und geben Sie das Zielverzeichnis in der Form `connect_direct_node_name:directory_path` an.

„[Transferring multiple files from a Connect:Direct node](#)“ auf Seite 334

You can transfer multiple files from a `Connect:Direct node` to a Managed File Transfer Agent by using the `Connect:Direct bridge`. You can specify a `Connect:Direct node` as the source of the multiple file transfer by specifying the `Connect:Direct bridge agent` as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

Zugehörige Verweise

[Die MFT agent.properties-Datei](#)

Wiederherstellung und Neustart für Übertragungen an und aus Connect:Direct-Knoten

Unter Umständen kann Managed File Transfer während einer Übertragung keine Verbindung zum IBM Sterling Connect:Direct-Knoten herstellen (wenn beispielsweise der Knoten plötzlich nicht mehr zur Verfügung steht). In diesem Fall versucht Managed File Transfer, die Übertragung wiederherzustellen, oder die Übertragung schlägt mit einer Fehlermeldung fehl.

Wenn der Connect:Direct-Knoten nicht mehr verfügbar ist

Wenn der `Connect:Direct-Knoten` plötzlich nicht mehr verfügbar ist (beispielsweise aufgrund eines Netz- oder Stromausfalls), unternimmt Managed File Transfer folgende Versuche, eine Dateiübertragung wiederherzustellen:

- Wenn Managed File Transfer zuvor nicht erfolgreich eine Verbindung zum `Connect:Direct -Knoten` als Teil dieser Übertragungsanforderung hergestellt hat, wird die Übertragung für einen Zeitraum wiederholt, der durch die Werte von `cdMaxConnectionRetries` und `recoverableTransferRetryInterval properties` bestimmt wird. Diese Eigenschaften werden in der `agent.properties`-Datei für den `Connect:Direct -Bridgeagenten` angegeben. Die Übertragung schlägt fehl, und es wird eine Fehlermeldung ausgegeben, nachdem die Anzahl der fehlgeschlagenen Versuche den Wert von

cdMaxConnectionRetries property erreicht hat. Standardmäßig wird die Übertragung auf unbestimmte Zeit mit 60 Sekunden zwischen den Versuchen versucht.

- Konnte Managed File Transfer im Rahmen einer vorherigen Übertragungsanforderung eine Verbindung zum Connect:Direct-Knoten herstellen, wird über einen bestimmten Zeitraum hinweg versucht, die Übertragung zu wiederholen; die Länge dieses Zeitraums wird über die Eigenschaften **cdMaxPartialWorkConnectionRetries** und **recoverableTransferRetryInterval** festgelegt. Die Übertragung schlägt fehl, und es wird eine Fehlernachricht ausgegeben, nachdem die Anzahl der fehlgeschlagenen Versuche den Wert der Eigenschaft **cdMaxPartialWorkConnectionRetries** erreicht hat. Standardmäßig wird die Übertragung auf unbestimmte Zeit mit 60 Sekunden zwischen den Versuchen versucht.
- Bei bestimmten Typen von Connect:Direct -Knotenausfällen, z. B. wenn das Stoppen des Knotens erzwungen wird, werden Connect:Direct -Prozesse in den Status `HEld Due to Error (HE)` versetzt, wenn der Knoten wiederhergestellt wird. Nach der Wiederherstellung des Knotens werden von Managed File Transfer alle Connect:Direct-Prozesse wiederaufgenommen, die in Zusammenhang mit der Dateiübertragung stehen und den Status HE haben.
- Schlägt die Übertragung fehl, werden in dem System, auf dem sich die Connect:Direct-Bridge befindet, alle temporären Dateien gelöscht, die mit der Übertragung zusammenhängen. Die Position dieser temporären Dateien wird durch die Eigenschaft **cdTmpDir** definiert.
- Erfolgt die Übertragung aus Managed File Transfer an Connect:Direct und wurde für die Handhabung der Quelldateien der Wert 'delete' angegeben, werden die Quelldateien beim Fehlschlagen der Übertragung nicht gelöscht.

Wenn die Benutzerberechtigungen für den Connect:Direct-Knoten ungültig sind

Werden die Berechtigungsnachweise des Benutzers vom Knoten abgelehnt und kann Managed File Transfer daher keine Verbindung zum Connect:Direct-Knoten herstellen, schlägt die Übertragung mit einer Fehlernachricht fehl. Vergewissern Sie sich in diesem Fall, dass Sie über die richtigen Benutzerberechtigungen für den Connect:Direct-Knoten verfügen. Weitere Informationen finden Sie im Abschnitt [Berechtigungsnachweise für Connect:Direct zuordnen](#).

Wenn der Connect:Direct-Bridgeagent nicht mehr verfügbar ist

Ist der Connect:Direct-Bridgeagent nicht mehr verfügbar, werden alle aktiven Übertragungen wie normale Managed File Transfer-Übertragungen wiederhergestellt. Weitere Informationen finden Sie unter [„Wiederherstellung und Neustart von MFT“](#) auf Seite 344.

Zugehörige Konzepte

[„Connect:Direct-Bridge“](#) auf Seite 327

Sie können Dateien in ein und aus einem IBM Sterling Connect:Direct-Netz übertragen. Mit der Connect:Direct-Bridge, einer Komponente von Managed File Transfer, können Sie Dateien zwischen MFT und IBM Sterling Connect:Direct übertragen.

[„Wiederherstellung und Neustart von MFT“](#) auf Seite 344

Sollte aus irgendeinem Grund Ihr Agent oder Warteschlangenmanager z. B. aufgrund eines Strom- oder Netzausfalls nicht verfügbar sein, führt Managed File Transfer eine Wiederherstellung gemäß folgender Szenarios durch:

Zugehörige Tasks

[Connect:Direct-Bridge konfigurieren](#)

Zugehörige Verweise

[Die MFT agent.properties-Datei](#)

Benutzerdefinierten Connect:Direct-Prozess aus einer Dateiübertragungsanforderung übergeben

Sie können eine Dateiübertragungsanforderung für eine Übertragung über den Connect:Direct-Bridgeagenten übergeben, die als Teil der Dateiübertragung einen benutzerdefinierten Connect:Direct-Prozess aufruft.

Wenn Sie eine Dateiübertragungsanforderung für eine Übertragung über die Connect:Direct-Bridge übergeben, generiert der Connect:Direct-Bridgeagent standardmäßig den Connect:Direct-Prozess, der zur Übertragung der Datei zum oder vom fernen Connect:Direct-Knoten benötigt wird.

Sie können den Connect:Direct -Bridgeagenten jedoch so konfigurieren, dass er stattdessen einen benutzerdefinierten Connect:Direct -Prozess unter Verwendung der ConnectDirectProcessDefinition.xml -Datei aufrufen kann.

Die ConnectDirectProcessDefinition.xml -Datei

Der Befehl **fteCreateCDAgent** erstellt die Datei ConnectDirectProcessDefinitions.xml im Agentenkonfigurationsverzeichnis *MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name*. Bevor Sie über den Connect:Direct-Bridgeagenten benutzerdefinierte Connect:Direct-Prozesse aufrufen, müssen Sie in dieser Datei die Prozessdefinitionen festlegen.

In der Datei sind eine oder mehrere Prozessgruppen definiert, die die Speicherposition eines oder mehrerer Connect:Direct-Prozesse angibt, die im Zuge einer Übertragung aufgerufen werden können. Jede Prozessgruppe enthält eine Reihe von Bedingungen. Erfüllt die Übertragung alle Bedingungen der Prozessgruppe, werden mithilfe dieser Prozessgruppe die Connect:Direct-Prozesse angegeben, die bei der Übertragung aufgerufen werden. Weitere Informationen finden Sie unter [„Zu startenden Connect:Direct-Prozess in der Datei 'ConnectDirectProcessDefinition.xml' festlegen“](#) auf Seite 339.

Interne Symbolvariablen

Mithilfe der von Managed File Transfer definierten internen Symbolvariablen können Sie Werte in benutzerdefinierten Connect:Direct-Prozessen ersetzen. Entsprechend der Namenskonvention von Connect:Direct haben alle von Managed File Transfer verwendeten internen Symbolvariablen das Format %FTE gefolgt von fünf großgeschrieben, alphanumerischen Zeichen.

Wenn Sie einen Prozess zum Übertragen von Dateien von einem Connect:Direct-Knoten auf das Connect:Direct-Bridgesystem erstellen, müssen Sie die intrinsische Variable %FTETFILE im Connect:Direct-Prozess als Wert für TO FILE verwenden. Wenn Sie einen Prozess zum Übertragen von Dateien vom Connect:Direct-Bridgesystem auf einen Connect:Direct-Knoten erstellen, müssen Sie die intrinsische Variable %FTEFFILE im Connect:Direct-Prozess als Wert für FROM FILE verwenden. Diese Variablen enthalten die temporären Dateipfade, die der Connect:Direct-Bridgeagent für Übertragungen zum und vom Managed File Transfer-Netz verwendet.

Weitere Informationen zu internen Symbolvariablen finden Sie in der Connect:Direct-Produktdokumentation.

Connect:Direct-Beispielprozesse

Managed File Transfer stellt Connect:Direct-Beispielprozesse bereit. Diese Beispiele befinden sich im folgenden Verzeichnis: *MQ_INSTALLATION_PATH/mqft/samples/ConnectDirectProcessTemplates*.

Zugehörige Tasks

[„Zu startenden Connect:Direct-Prozess in der Datei 'ConnectDirectProcessDefinition.xml' festlegen“](#) auf Seite 339

Geben Sie den Connect:Direct-Prozess an, der als Teil einer Managed File Transfer-Übertragung gestartet werden soll. In Managed File Transfer ist eine XML-Datei bereitgestellt, in der Sie die Prozessdefinitionen angeben können.

[„Interne symbolische Variablen in Connect:Direct-Prozessen, die von Managed File Transfer aufgerufen werden“](#) auf Seite 340

Sie können einen benutzerdefinierten Connect:Direct -Prozess aus einer Managed File Transfer -Übertragung aufrufen und Informationen aus der Übertragung an den Connect:Direct -Prozess übergeben, indem Sie interne Symbolvariablen in der Prozessdefinition verwenden.

Zugehörige Verweise

[Format der Datei mit den Connect:Direct-Prozessdefinitionen](#)

[Substitutionsvariablen für benutzerdefinierte Connect:Direct-Prozesse](#)

Zu startenden Connect:Direct-Prozess in der Datei 'ConnectDirectProcessDefinition.xml' festlegen

Geben Sie den Connect:Direct-Prozess an, der als Teil einer Managed File Transfer-Übertragung gestartet werden soll. In Managed File Transfer ist eine XML-Datei bereitgestellt, in der Sie die Prozessdefinitionen angeben können.

Informationen zu diesem Vorgang

Der Befehl **fteCreateCDAgent** erstellt die Datei `ConnectDirectProcessDefinitions.xml` im Agentenkonfigurationsverzeichnis `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`. Bevor Sie über den Connect:Direct-Bridgeagenten benutzerdefinierte Connect:Direct-Prozesse aufrufen, müssen Sie in dieser Datei die Prozessdefinitionen festlegen.

Führen Sie für jeden Prozess, der im Zuge einer Übertragung über die Connect:Direct-Bridge aufgerufen werden soll, die folgenden Schritte aus:

Vorgehensweise

1. Definieren Sie den Connect:Direct-Prozess, den der Connect:Direct-Bridgeagent als Teil einer Übertragung aufrufen soll, und speichern Sie die Prozessvorlage in einer Datei.
2. Öffnen Sie die Datei `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` in einem Texteditor.
3. Erstellen Sie ein Element `<processSet>`.
4. Erstellen Sie innerhalb des `<processSet>`-Elements ein Element `<condition>`.
5. Erstellen Sie innerhalb des Elements `<condition>` ein oder mehrere Elemente, die eine Bedingung definieren, mit der die Übertragungsanforderung übereinstimmen muss, um den Connect:Direct -Prozess aufzurufen, den Sie in Schritt 1 definiert haben. Bei diesen Elementen kann es sich um `<match>`-Elemente oder um `<defined>`-Elemente handeln.
 - Verwenden Sie ein `<match>`-Element, um anzugeben, dass der Wert einer Variablen mit einem Muster übereinstimmen muss. Erstellen Sie das Element `<match>` mit den folgenden Attributen:
 - `variable` : Der Name der Variablen, deren Wert verglichen wird. Die Variable ist ein intrinsisches Symbol. Weitere Informationen finden Sie unter [Substitutionsvariablen für die Verwendung mit benutzerdefinierten Connect:Direct-Prozessen](#).
 - `value` : Das Muster, das mit dem Wert der angegebenen Variablen verglichen werden soll.
 - Optional: `pattern` -Der Typ des Musters, der durch den Wert des Attributs `value` verwendet wird. Dieser Mustertyp kann `wildcard` oder `regex` sein. Dieses Attribut ist optional, und der Standardwert ist `wildcard`.
 - Verwenden Sie ein `<defined>`-Element, um anzugeben, dass eine Variable einen definierten Wert haben muss. Erstellen Sie das Element `<defined>` mit dem folgenden Attribut:
 - `variable` : Der Name der Variablen, für die ein Wert definiert sein muss. Die Variable ist ein intrinsisches Symbol. Weitere Informationen finden Sie unter [Substitutionsvariablen für die Verwendung mit benutzerdefinierten Connect:Direct-Prozessen](#).

Die Bedingungen, die im Element `<condition>` angegeben sind, werden mit einem logischen UND kombiniert. Alle Bedingungen müssen erfüllt sein, damit der Connect:Direct -Bridgeagent den von diesem `<processSet>`-Element angegebenen Prozess aufrufen kann. Wenn Sie kein Element `<condition>` angeben, stimmt der Prozesssatz mit allen Übertragungen überein.

6. Erstellen Sie innerhalb des `<processSet>`-Elements ein Element `<process>`.
7. Erstellen Sie innerhalb des `<process>`-Elements ein Element `<transfer>`.

Das `transfer`-Element gibt den Connect:Direct-Prozess an, den der Connect:Direct-Bridgeagent als Teil der Übertragung aufruft. Erstellen Sie das Element `<transfer>` mit dem folgenden Attribut:

- process- -Die Position des Connect:Direct -Prozesses, den Sie in Schritt 1 definiert haben Die Position dieser Datei wird mit einem absoluten Pfad oder relativ zum Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` angegeben.

Ergebnisse

Bei der Suche nach einer übereinstimmenden Bedingung durchsucht der Connect:Direct-Bridgeagent die Datei von oben nach unten. Die erste gefunden, die gefunden wird, wird verwendet.

Zugehörige Tasks

[Connect:Direct-Bridge konfigurieren](#)

Zugehörige Verweise

[Format der Datei mit den Connect:Direct-Prozessdefinitionen](#)

[fteCreateCDAgent: Connect:Direct-Bridgeagenten erstellen](#)

Interne symbolische Variablen in Connect:Direct-Prozessen, die von Managed File Transfer aufgerufen werden

Sie können einen benutzerdefinierten Connect:Direct -Prozess aus einer Managed File Transfer -Übertragung aufrufen und Informationen aus der Übertragung an den Connect:Direct -Prozess übergeben, indem Sie interne Symbolvariablen in der Prozessdefinition verwenden.

Informationen zu diesem Vorgang

In diesem Beispiel werden mithilfe interner Symbolvariablen Informationen aus einer Managed File Transfer-Übertragung an einen benutzerdefinierten Connect:Direct-Prozess übergeben. Weitere Informationen zu intrinsischen Symbolvariablen, die von Managed File Transfer verwendet werden, finden Sie unter [Substitutionsvariablen für die Verwendung mit benutzerdefinierten Connect:Direct-Prozessen](#).

In diesem Beispiel wird die Datei aus einem Managed File Transfer Agent an einen Connect:Direct-Bridgeknoten übertragen. Der erste Teil der Übertragung wird von Managed File Transfer übernommen. Der zweite Teil wird von einem benutzerdefinierten Connect:Direct-Prozess durchgeführt.

Vorgehensweise

1. Erstellen Sie einen Connect:Direct-Prozess, der interne Symbolvariablen verwendet.

```
%FTEPNAME PROCESS
SNODE=%FTESNODE
PNODEID=(%FTEPUSER,%FTEPPASS)
SNODEID=(%FTESUSER,%FTESPASS)

COPY001 COPY
FROM (
  FILE=%FTEFFILE
  DISP=%FTEFDISP
)
TO (
  FILE=%FTETFILE
  DISP=%FTETDISP
)
PEND
```

2. Speichern Sie diesen Prozess in einer Textdatei an der folgenden Position: `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent/Example.cdp`
3. Bearbeiten Sie die `ConnectDirectProcessDefinition.xml` -Datei so, dass sie eine Regel enthält, die den Connect:Direct -Prozess aufruft, den Sie in Schritt 1 erstellt haben.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/
  ConnectDirectProcessDefinitions ConnectDirectProcessDefinitions.xsd">

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="TOBERMORY" pattern="wildcard" />
    </tns:condition>
  </tns:processSet>
</tns:cdprocess>
```

```

</tns:condition>
<tns:process>
  <tns:transfer process="Example.cdp" />
</tns:process>
</tns:processSet>
</tns:cdprocess>

```

Wenn in diesem Beispiel eine Übertragungsanforderung an den Connect:Direct -Bridgeagenten übergeben wird, der TOBERMORY als seinen Quellen- oder Zielknoten Connect:Direct hat, wird der Example.cdp Connect:Direct -Prozess aufgerufen.

- Übergeben Sie eine Dateiübertragungsanforderung, die die Bedingungen erfüllt, die Sie in der ConnectDirectProcessDefinition.xml -Datei in Schritt 3 definiert haben.

Beispiel:

```

fteCreateTransfer -sa ORINOCO -da CD_BRIDGE
                 -sm QM_WIMBLEDON -dm QM_COMMON
                 -de overwrite -df TOBERMORY:/home/bulgaria/destination.txt
                 -sd leave c:\bungo\source.txt

```

In diesem Beispiel heißt der Connect:Direct-Zielknoten TOBERMORY. Dieser Knoten ist der sekundäre Knoten in der Übertragung, und der Wert von %FTESNODE wird auf TOBERMORY gesetzt. Dieser Befehl stimmt mit der Bedingung überein, die in der ConnectDirectProcessDefinition.xml -Datei festgelegt ist.

- Managed File Transfer überträgt die Quelldatei in ein temporäres Verzeichnis auf dem System, auf dem sich der Connect:Direct-Bridgeagent befindet.
- Der Connect:Direct-Bridgeagent bestimmt die Werte der internen Symbolvariablen aus den Informationen der Übertragungsanforderung sowie aus den Konfigurationsinformationen.

Die intrinsischen Symbolvariablen werden auf die folgenden Werte gesetzt:

- %FTEPNAME=*Prozessname* - Dieser Wert ist ein achtstelliger Prozessname, der vom Connect:Direct-Bridgeagenten generiert wird.
 - %FTESNODE= tobermory-Dieser Wert wird über den Parameter **-df** des Befehls **fteCreateTransfer** festgelegt.
 - %FTEPUSER,=*primary_node_user* -Diese Informationen werden aus der ConnectDirectCredentials.xml -Datei entnommen.
 - %FTEPPASS=*primary_node_user_password* -Diese Informationen werden aus der ConnectDirectCredentials.xml -Datei entnommen.
 - %FTESUSER,=*secondary_node_user* -Diese Informationen werden aus der ConnectDirectCredentials.xml -Datei entnommen.
 - %FTESPASS=*secondary_node_user_password* -Diese Informationen werden aus der ConnectDirectCredentials.xml -Datei entnommen.
 - D%FTEFFILE =*temporäres_Verzeichnis* - Das temporäre Verzeichnis auf dem System des Connect:Direct-Bridgeagenten, in dem die Datei gespeichert wird.
 - %FTEFDISP=leave-Dieser Wert wird über den Parameter **-sd** des Befehls **fteCreateTransfer** festgelegt.
 - %FTETFILE=/home/bulgaria/destination.txt -Dieser Wert wird über den Parameter **-df** des Befehls **fteCreateTransfer** festgelegt.
 - %FTETDISP=overwrite-Dieser Wert wird über den Parameter **-de** des Befehls **fteCreateTransfer** festgelegt.
- Der Connect:Direct-Prozess wird auf dem Connect:Direct-Bridgeknoten gestartet. Connect:Direct überträgt die Datei aus dem temporären Verzeichnis des Connect:Direct-Bridgesystems an das Ziel /home/bulgaria/destination.txt auf dem System, auf dem der Connect:Direct-Knoten TOBERMORY ausgeführt wird.

Zugehörige Konzepte

[„Benutzerdefinierten Connect:Direct-Prozess aus einer Dateiübertragungsanforderung übergeben“ auf Seite 337](#)

Sie können eine Dateiübertragungsanforderung für eine Übertragung über den Connect:Direct-Bridgeagenten übergeben, die als Teil der Dateiübertragung einen benutzerdefinierten Connect:Direct-Prozess aufruft.

Zugehörige Verweise

[Substitutionsvariablen für benutzerdefinierte Connect:Direct-Prozesse](#)

Connect:Direct-Prozesse zur Übergabe von Managed File Transfer-Übertragungsanforderungen verwenden

Die Übergabe einer Übertragungsanforderung an den Connect:Direct-Bridgeagenten kann aus einem Connect:Direct-Prozess erfolgen. Hierzu bietet Managed File Transfer Befehle, die aus einer **RUN TASK**-Anweisung eines Connect:Direct-Prozesses aufgerufen werden können.

Managed File Transfer stellt folgende Befehle zur Verwendung in Connect:Direct-Prozessen bereit:

ftetag

Geben Sie diesen Befehl in einem Schritt vor dem Befehl **ftebxfer** oder **ftecxfer** an, um die erforderlichen Prüfinformationen für die Übertragung zu erstellen. Mit diesem Befehl wird die Quellspezifikation der Übertragung als Parameter verwendet. Informationen zum Format der Quellspezifikation finden Sie in [fteCreateTransfer: Neue Dateiübertragung starten](#).

ftebxfer

Geben Sie diesen Befehl an, um eine Dateiübertragungsanforderung zu erstellen, wenn sich der Warteschlangenmanager, an den die Übertragungsanforderung übergeben wird, auf demselben System befindet wie der Connect:Direct -Knoten, der den Befehl übergibt. Dieser Befehl verwendet dieselben Parameter wie der Befehl **fteCreateTransfer**. Informationen zu diesen Parametern finden Sie unter [fteCreateTransfer: Neue Dateiübertragung starten](#). Dieser Befehl verfügt auch über einen zusätzlichen Parameter:

-qmgrname

Erforderlich. Der Name des Warteschlangenmanagers, an den der Befehl übergeben werden soll.

ftecxfer

Führen Sie diesen Befehl zur Erstellung einer Dateiübertragungsanforderung aus, wenn sich der Warteschlangenmanager, an den die Übertragungsanforderung übergeben wird, auf einem anderen System befindet als der Connect:Direct-Knoten, auf dem der Befehl ausgegeben wird. Dieser Befehl verwendet dieselben Parameter wie der Befehl **fteCreateTransfer**. Informationen zu den Parametern finden Sie unter [fteCreateTransfer: Neue Dateiübertragung starten](#). Dieser Befehl verfügt außerdem über drei zusätzliche Parameter:

-qmgrname

Erforderlich. Der Name des Warteschlangenmanagers, an den der Befehl übergeben werden soll.

-conname

Erforderlich. Der Host und Port des Warteschlangenmanagers, an den der Befehl übergeben wird, angegeben im CONNAME-Format von IBM MQ. Beispiel: `host.example.com(1337)`.

-channelname

Optional. Der Name des Kanals, der für die Verbindung zum Warteschlangenmanager verwendet werden soll, um den Befehl an zu übergeben. Wenn dieser Parameter nicht angegeben wird, wird der Standardwert `SYSTEM.DEF.SVRCONN` verwendet.

Zugehörige Tasks

[„Erstellen und Übergeben eines Connect:Direct -Prozesses, der Managed File Transfer aufruft, mithilfe des Connect:Direct -Requesters“ auf Seite 343](#)

Bei dem Connect:Direct Requester handelt es sich um eine grafische Benutzerschnittstelle, mit der ein Connect:Direct-Prozess erstellt und übergeben werden kann, mit dem Managed File Transfer aufgerufen wird.

Zugehörige Verweise

Beispiel: [Eine Connect: Direct-Prozessdatei, die MFT-Befehle aufruft](#)

Erstellen und Übergeben eines Connect:Direct -Prozesses, der Managed File Transfer aufruft, mithilfe des Connect:Direct -Requesters

Bei dem Connect:Direct Requester handelt es sich um eine grafische Benutzerschnittstelle, mit der ein Connect:Direct-Prozess erstellt und übergeben werden kann, mit dem Managed File Transfer aufgerufen wird.

Informationen zu diesem Vorgang

In diesem Abschnitt wird beschrieben, wie ein Connect:Direct-Prozess erstellt wird, mit dem der Managed File Transfer-Befehl **ftecxfer** oder **ftebxfer** aufgerufen wird. Verwenden Sie den Befehl **ftecxfer**, wenn sich der Warteschlangenmanager, an den die Übertragungsanforderung übergeben wird, auf einem anderen System befindet als der Connect:Direct -Knoten, der den Befehl übergibt. Verwenden Sie den Befehl **ftebxfer**, wenn der Warteschlangenmanager, an den die Übertragungsanforderung übergeben wird, sich auf demselben System befindet wie der Connect:Direct-Knoten, der den Befehl übergibt. Mit dem Befehl **ftecxfer** wird eine Clientverbindung zum Agentenwarteschlangenmanager des Quellenagenten der Übertragung hergestellt. Vor dem Aufruf des Befehls **ftecxfer** müssen Sie den Befehl **ftetag** aufrufen und die Quellenspezifikationsinformationen übergeben. Auf diese Weise kann der Prozess protokolliert und in derselben Weise protokolliert werden wie Übertragungen, die von Managed File Transfer eingeleitet werden.

Vorgehensweise

1. Starten Sie den Connect:Direct Requester.
2. Wählen Sie auf der Registerkarte **Knoten** des Teilfensters den Connect:Direct-Knoten aus, der als Primärknoten des Prozesses verwendet wird.
3. Wählen Sie **Datei > Neu > Prozess** aus. Das Fenster **Prozesseigenschaften** wird geöffnet.
4. Geben Sie in das Feld **Name:** den Namen des Prozesses ein.
5. Wählen Sie aus der Liste **Snode > Name** (Name des Sekundärknotens) den Sekundärknoten aus.
6. Wählen Sie in der Liste **Snode > Operating System:** das Betriebssystem des Sekundärknotens aus.
7. Optional: Geben Sie alle weiteren Informationen in diesem Fenster an, die Sie benötigen.
8. Klicken Sie auf **OK**. Das Fenster **Prozesseigenschaften** wird geschlossen.
9. Erstellen Sie eine Anweisung, die den Managed File Transfer-Befehl **ftetag** ausführt.
 - a) Klicken Sie im Fenster **Prozess** auf der **Ende** -Anweisung auf "Right-click".
 - b) Wählen Sie **Insert > Run Task** (Einfügen > Task ausführen) aus. Das Fenster **Taskanweisung ausführen** wird geöffnet.
 - c) Geben Sie im Feld **Bezeichnung:** Tagein.
 - d) Geben Sie im Feld **Optionale Parameter oder Befehle** `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)` ein. Weitere Informationen zum Format von *Quellenspezifikation* finden Sie unter [fteCreateTransfer: Neue Dateiübertragung starten](#).
 - e) Klicken Sie auf **OK**. Das Fenster **Taskanweisung ausführen** wird geschlossen.
10. Erstellen Sie eine Anweisung, die den Managed File Transfer-Befehl **ftecxfer** oder **ftebxfer** ausführt.
 - a) Klicken Sie im Fenster **Prozess** auf der **Ende** -Anweisung auf "Right-click".
 - b) Wählen Sie **Insert > Run Task** (Einfügen > Task ausführen) aus. Das Fenster **Taskanweisung ausführen** wird geöffnet.
 - c) Geben Sie im Feld **Bezeichnung:** Transferein.
 - d) Geben Sie im Feld **Optionale Parameter oder Befehle** je nach ausgewähltem Befehl `pgm(MQ_INSTALLATION_PATH/bin/ftecxfer) args(parameters)` oder `pgm(MQ_INSTALLATION_PATH/bin/ftebxfer) args(parameters)` ein. Die von den Befehlen **ftecxfer** und

ftebxf verwendeten Parameter sind mit den Parametern identisch, die vom Befehl **fteCreateTransfer** verwendet werden, sowie einige zusätzliche Parameter, die speziell für **ftecxf** und **ftebxf** gelten. Weitere Informationen hierzu finden Sie unter **fteCreateTransfer: Neue Dateiübertragung starten** und „Connect:Direct-Prozesse zur Übergabe von Managed File Transfer-Übertragungsanforderungen verwenden“ auf Seite 342.

- e) Klicken Sie auf **OK**. Das Fenster **Taskanweisung ausführen** wird geschlossen.
11. Optional: Erstellen Sie alle zusätzlichen Anweisungen, die Sie benötigen.
12. Übergeben Sie den Prozess.
 - a) Klicken Sie im Fenster **Prozess** auf "Right-click".
 - b) Wählen Sie **Übergeben** aus. Das Fenster **Connect: Direct Attach** wird geöffnet.
 - c) Geben Sie den Benutzernamen und das Kennwort ein, die für die Ausführung des Prozesses verwendet werden sollen.
 - d) Klicken Sie auf **OK**.

Zugehörige Konzepte

„Connect:Direct-Prozesse zur Übergabe von Managed File Transfer-Übertragungsanforderungen verwenden“ auf Seite 342

Die Übergabe einer Übertragungsanforderung an den Connect:Direct-Bridgeagenten kann aus einem Connect:Direct-Prozess erfolgen. Hierzu bietet Managed File Transfer Befehle, die aus einer **RUN TASK**-Anweisung eines Connect:Direct-Prozesses aufgerufen werden können.

Mit MFT über IBM Integration Bus arbeiten

Unter Verwendung der FTEOutput- und FTEInput-Knoten können Sie mit Managed File Transfer über IBM Integration Bus arbeiten.

- Über den FTEInput-Knoten können Sie eine Datei mithilfe von Managed File Transfer in einem Netz übertragen und anschließend als Teil eines Integration Bus-Nachrichtenflusses verarbeiten.
- Mit dem FTEOutput-Knoten können Sie eine Datei übertragen, die von einem Integration Bus-Nachrichtenfluss an eine andere Position im Netz ausgegeben wurde.

Bei den Agenten, die Dateien an den bzw. aus dem Brokeragenten übertragen, spielt die Version von Managed File Transfer keine Rolle.

Weitere Informationen finden Sie in der [IBM Integration Bus-Produktdokumentation](#).

Wiederherstellung und Neustart von MFT

Sollte aus irgendeinem Grund Ihr Agent oder Warteschlangenmanager z. B. aufgrund eines Strom- oder Netzausfalls nicht verfügbar sein, führt Managed File Transfer eine Wiederherstellung gemäß folgender Szenarios durch:

- Wenn ein Fehler beim Übertragen einer Datei auftritt, führt Managed File Transfer normalerweise eine Wiederherstellung durch und startet nach Beheben des Fehlers diese Dateiübertragung erneut.
- Wird eine Datei im Verlauf der Übertragung gelöscht oder geändert, solange der Agent oder Warteschlangenmanager nicht verfügbar ist, schlägt die Übertragung fehl und Sie erhalten eine Nachricht mit Einzelheiten zu dem Fehler im Übertragungsprotokoll.
- Schlägt während einer Dateiübertragung ein Agentenprozess fehl, wird die Übertragung beim Neustart des Agenten fortgesetzt.
- Wenn die Verbindung zwischen einem Agenten und seinem Agenten-Warteschlangenmanager unterbrochen wird, wartet der Agent, während er versucht, die Verbindung zum Warteschlangenmanager wiederherzustellen. Wenn der Agent erfolgreich eine erneute Verbindung zu seinem Warteschlangenmanager herstellt, wird die aktuelle Übertragung fortgesetzt.
- Wird der Agent aus irgendeinem Grund gestoppt, stellen alle ihm zugeordneten Ressourcenüberwachungen die Ressourcenabfrage ein. Bei einer Wiederherstellung des Agenten werden die Überwachungen ebenfalls erneut gestartet und die Ressourcenabfrage wird wieder aufgenommen.

- Erfolgt bei einer Dateiübertragung mit der Quellendisposition `delete` eine Wiederherstellung, nachdem alle Daten von einem Quellenagenten zu einem Zielagenten gesendet wurden, so wird die Quellendatei entsperrt, bevor sie gelöscht wird. Aufgrund dieser Entsperrung könnte es theoretisch passieren, dass die Quellendatei geändert wird, bevor sie gelöscht wird. Aus diesem Grund wird ein Löschen der Quellendatei als unsicherer Vorgang betrachtet und folgende Warnung wird ausgegeben:

```
BFGTR0075W: The source file has not been deleted because it is possible that the source file was modified after the source file was transferred.
```

Überprüfen Sie in diesem Fall, ob der Inhalt der Quellendatei unverändert ist, und löschen Sie sie danach manuell.

Sie können den Status Ihrer Übertragungen in IBM MQ Explorer prüfen. Wenn Übertragungen als blockiert angezeigt werden (`Stalled`), müssen Sie möglicherweise Maßnahmen zur Fehlerbehebung ergreifen, da der Blockierungsstatus auf ein Problem im Zusammenhang mit dem Agenten oder zwischen den beiden Agenten hinweist, die an der Übertragung beteiligt sind.

Zugehörige Tasks

„Zeitlimit für die Wiederherstellung von blockierten Übertragungen festlegen“ auf Seite 345

Sie können ein Zeitlimit für die Übertragungswiederherstellung für blockierte Dateiübertragungen festlegen, die für alle Übertragungen eines Quellenagenten angewendet werden. Sie können auch ein Zeitlimit für die Übertragungswiederherstellung für eine einzelne Übertragung festlegen. Wenn Sie eine bestimmte Zeitdauer (in Sekunden) festlegen, während der ein Quellenagent versucht, eine blockierte Dateiübertragung wiederherzustellen, und die Übertragung noch nicht erfolgreich war, wenn der Agent dieses Zeitlimit erreicht, schlägt die Übertragung fehl.

Zeitlimit für die Wiederherstellung von blockierten Übertragungen festlegen

Sie können ein Zeitlimit für die Übertragungswiederherstellung für blockierte Dateiübertragungen festlegen, die für alle Übertragungen eines Quellenagenten angewendet werden. Sie können auch ein Zeitlimit für die Übertragungswiederherstellung für eine einzelne Übertragung festlegen. Wenn Sie eine bestimmte Zeitdauer (in Sekunden) festlegen, während der ein Quellenagent versucht, eine blockierte Dateiübertragung wiederherzustellen, und die Übertragung noch nicht erfolgreich war, wenn der Agent dieses Zeitlimit erreicht, schlägt die Übertragung fehl.

Informationen zu diesem Vorgang

Sie können ein Zeitlimit für die Übertragungswiederherstellung festlegen, das für alle Übertragungen für einen Quellenagenten gilt, indem Sie der Datei `agent.properties` des Agenten einen Parameter für das Zeitlimit für die Übertragungswiederherstellung hinzufügen. Sie können auch ein Zeitlimit für die Übertragungswiederherstellung für eine einzelne Übertragung aus der Befehlszeile, mit dem IBM MQ Explorer oder mithilfe von Apache Ant-Tasks festlegen. Wenn in der Datei `agent.properties` ein Wert für das Zeitlimit für die Übertragungswiederherstellung festgelegt ist, wird durch das Festlegen des Zeitlimits für die Übertragungswiederherstellung für eine einzelne Übertragung der in der Datei `agent.properties` festgelegte Wert überschrieben.

Es gibt drei Optionen für das Zeitlimit für die Übertragungswiederherstellung:

- Der Agent versucht solange die blockierte Übertragung wiederherzustellen, bis sie erfolgreich abgeschlossen wird. Dies entspricht dem Standardverhalten des Agenten, wenn kein Zeitlimit für die Übertragungswiederherstellung festgelegt ist.
- Der Agent markiert die Übertragung direkt nach dem Beginn der Wiederherstellung als fehlgeschlagen.
- Der Agent versucht, die blockierte Übertragung über eine angegebene Dauer wiederherzustellen, bevor die Übertragung als fehlgeschlagen markiert wird.

Das Festlegen eines Zeitlimits für die Übertragungswiederherstellung ist optional. Wenn Sie ihn nicht festlegen, folgen Übertragungen dem Standardverhalten, bei dem der Agent weiterhin versucht, eine blockierte Übertragung wiederherzustellen, bis er erfolgreich ist.

Zugehörige Konzepte

„Wiederherstellung und Neustart von MFT“ auf Seite 344

Sollte aus irgendeinem Grund Ihr Agent oder Warteschlangenmanager z. B. aufgrund eines Strom- oder Netzausfalls nicht verfügbar sein, führt Managed File Transfer eine Wiederherstellung gemäß folgender Szenarios durch:

Konzepte für das Zeitlimit für die Übertragungswiederholung

Sie können die Zeit (in Sekunden) festlegen, in der ein Quellenagent versucht, eine blockierte Dateiübertragung wiederherzustellen. Wenn die Übertragung nicht erfolgreich ist, wenn der Agent das Zeitlimit für das Wiederholungsintervall erreicht, schlägt die Übertragung fehl.

Vorrangstellung für Wiederherstellungszeitlimit

Ein Zeitlimitwert für die Übertragungswiederherstellung für eine einzelne Übertragung, der über die Befehle **fteCreateTransfer**, **fteCreateTemplate** oder **fteCreateMonitor** oder mithilfe von IBM MQ Explorer angegeben oder im verschachtelten Element **fte:filespec** angegeben wird, hat Vorrang vor dem Wert, der für den Parameter **transferRecoveryTimeout** in der Datei `agent.properties` für den Quellenagenten angegeben ist.

Wenn der Befehl **fteCreateTransfer** beispielsweise ohne den Parameter **-rt** und das Wertepaar gestartet wird, überprüft der Quellenagent AGENT1 die Datei `agent.properties` auf den Wert **transferRecoveryTimeout**, um das Verhalten des Wiederherstellungszeitlimits zu bestimmen:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

Wenn der Parameter **transferRecoveryTimeout** in der Datei `agent.properties` nicht festgelegt oder auf `-1` gesetzt ist, folgt der Agent dem Standardverhalten und versucht, die Übertragung herzustellen, bis sie erfolgreich ist.

Wenn der Befehl **fteCreateTransfer** allerdings den Parameter **-rt** enthält, hat der Wert dieses Parameters Vorrang vor dem Wert in der Datei `agent.properties` und wird als Einstellung für das Wiederherstellungszeitlimit verwendet:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

Zähler für das Wiederherstellungszeitlimit

Der Zähler für das Wiederherstellungszeitlimit wird gestartet, wenn die Übertragung in den Wiederherstellungsstatus wechselt. Eine Übertragungsprotokollnachricht wird in dem Thema SYSTEM.FTE mit der Themenzeichenfolge `Log/agent_name/transfer_ID` veröffentlicht, um anzuzeigen, dass der Übertragungsstatus in die Wiederherstellung geändert wird und die Uhrzeit der Quellenagent, bei der sich der Status geändert hat, geändert wird. Wenn die Übertragung innerhalb des festgelegten Wiederholungsintervalls fortgesetzt wird und nicht das Wiederherstellungszeitlimit erreicht (Zähler \leq Wiederherstellungszeitlimit), wird der Zähler auf 0 zurückgesetzt, wenn die Übertragung erneut gestartet wird, wenn die Übertragung in die Wiederherstellung eintritt.

Wenn der Zähler den Maximalwert für das Wiederherstellungszeitlimit erreicht (Zähler $==$ Wiederherstellungszeitlimit), wird die Wiederherstellung der Übertragung gestoppt, und der Quellenagent meldet die Übertragung als fehlgeschlagen. Dieser Typ eines Übertragungsfehlers, der durch die Tatsache verursacht wird, dass die Übertragung das Wiederherstellungszeitlimit erreicht hat, wird durch den Nachrichtencode `RECOVERY TIMEOUT (69)` angezeigt. Eine andere Übertragungsprotokollnachricht wird in dem Thema SYSTEM.FTE mit der Themenzeichenfolge `Log/agent_name/transfer_ID` veröffentlicht, um anzuzeigen, dass die Übertragung fehlgeschlagen ist und eine Nachricht, den Rückkehrcode und das Ereignisprotokoll des Quellenagenten enthält. Das Ereignisprotokoll des Quellenagenten wird mit einer Nachricht aktualisiert, wenn eines der folgenden Ereignisse während der Wiederherstellung auftritt:

- Wenn der Parameter für das Wiederherstellungszeitlimit auf einen Wert größer `-1` gesetzt wird, beginnt die Wiederherstellung der Übertragung. Das Ereignisprotokoll des Agenten wird aktualisiert, um den

Start des Zeitgebers für die Wiederherstellung für den **TransferId** und die Zeit anzugeben, die der Quellenagent wartet, bevor er die Wiederherstellungszeitlimitverarbeitung einleitet.

- Wenn die Wiederherstellung der Übertragung fortgesetzt wird, wird das Ereignisprotokoll des Quellenagenten aktualisiert und enthält nun eine neue Nachricht, die angibt, dass die Übertragung **TransferId**, für die eine Wiederherstellung vorgenommen wurde, fortgesetzt wird.
- Wenn das Zeitlimit bei der Wiederherstellung einer Übertragung überschritten wird, wird das Ereignisprotokoll des Quellenagenten aktualisiert und gibt an, dass die Übertragung **TransferId** aufgrund einer Überschreitung des Wiederherstellungszeitlimits während der versuchten Wiederherstellung fehlgeschlagen ist.

Diese Protokollnachrichten ermöglichen es den Benutzern (Subskribenten und Protokollfunktionen), die Übertragungen zu identifizieren, die aufgrund des Zeitlimits für die Übertragungswiederherstellung fehlgeschlagen sind.

Der Zähler für das Wiederherstellungszeitlimit ist immer auf dem Quellenagenten vorhanden. Wenn der Zielagent jedoch nicht rechtzeitig Informationen vom Quellenagenten empfängt, kann er eine Anforderung an den Quellenagenten senden, um die Übertragung in die Wiederherstellung zu versetzen. Bei einer Übertragung, bei der die Option für das Wiederherstellungszeitlimit festgelegt ist, startet der Quellenagent den Wiederherstellungszeitlimitzähler, wenn er die Anforderung von dem Zielagenten empfängt.

Für Übertragungen, die nicht das Zeitlimit für Wiederherstellung verwenden, die fehlgeschlagenen und teilweise vollständigen Übertragungen sind weiterhin manuelle Handhabung erforderlich.

Bei Übertragungsgruppen, bei denen eine einzelne Übertragungsanforderung für mehrere Dateien ausgegeben wird und einige der Dateien erfolgreich abgeschlossen wurden, die Übertragung jedoch nur teilweise abgeschlossen ist, wird die Übertragung immer noch als fehlgeschlagen markiert, da sie nicht wie erwartet abgeschlossen wurde. Der Quellenagent hat möglicherweise das zulässige Zeitlimit überschritten, während die teilweise abgeschlossene Datei übertragen wurde.

Stellen Sie sicher, dass der Zielagent und der Dateiserver bereit und in einem Status sind, um Dateiübertragungen zu akzeptieren.

Sie müssen die Übertragungsanforderung erneut für die gesamte Gruppe ausgeben. Da aber einige der Dateien vom ursprünglichen Übertragungsversuch auf dem Ziel verbleiben, können Sie zur Vermeidung von Problemen die neue Anforderung mit Angabe der Option **Überschreiben**, falls bereits vorhanden ausgeben. Dadurch wird sichergestellt, dass der unvollständige Satz von Dateien aus dem vorherigen Übertragungsversuch als Teil der neuen Übertragung bereinigt wird, bevor die Dateien wieder in das Ziel geschrieben werden.

Ab IBM MQ 9.1.5 müssen Teildateien in einem Ziel nicht mehr manuell entfernt werden, nachdem ein erster Übertragungsversuch fehlgeschlagen ist. Wenn für eine Übertragung ein Zeitlimit für die Übertragungswiederholung festgelegt ist und dies überschritten wird, verschiebt der Quellenagent die Übertragung in den Status **RecoveryTimedOut**. Nachdem die Übertragung resynchronisiert wurde, entfernt der Zielagent alle Teildateien, die während der Übertragung erstellt wurden, und sendet eine Beendigungsnachricht an den Quellenagenten.

Traces und Nachrichten

Trace-Punkte werden zu Diagnosezwecken verwendet. Das Wiederherstellungszeitlimitwert, den Beginn des Wiederholungsintervalls, den Beginn des Wiederaufnahmezeitraums und den Zähler zurücksetzen sowie die Angabe, ob die Übertragung das zulässige Zeitlimit überschritten hat und fehlgeschlagen ist, werden protokolliert. Wenn Probleme auftreten oder es zu nicht erwartetem Verhalten kommt, können Sie die Tracedateien und das Ausgabeprotokoll des Quellenagenten erfassen und auf Aufforderung an den IBM Support übergeben, um die Fehlerbehebung zu unterstützen.

In folgenden Fällen erhalten Sie eine Nachricht:

- Die Wiederherstellung einer Übertragung beginnt (**BFGTR0081I**)
- Eine Übertragung wird beendet, weil das Zeitlimit der Wiederherstellung überschritten wurde (**BFGSS0081E**)
- Eine Übertragung wird nach der Wiederherstellung fortgesetzt (**BFGTR0082I**)

Zugehörige Konzepte

„Wiederherstellung und Neustart von MFT“ auf Seite 344

Sollte aus irgendeinem Grund Ihr Agent oder Warteschlangenmanager z. B. aufgrund eines Strom- oder Netzausfalls nicht verfügbar sein, führt Managed File Transfer eine Wiederherstellung gemäß folgender Szenarios durch:

Zeitlimit für Übertragungswiederherstellung für alle Übertragungen für einen Quellenagenten festlegen

Sie können ein Zeitlimit für die Übertragungswiederherstellung festlegen, das für alle Übertragungen für einen Quellenagenten angewendet wird, indem Sie den Parameter **transferRecoveryTimeout** der Datei `agent.properties` hinzufügen.

Informationen zu diesem Vorgang

Um ein Zeitlimit für die Übertragungswiederherstellung festzulegen, das für alle Übertragungen für einen Quellenagenten angewendet wird, fügen Sie den Parameter **transferRecoveryTimeout** und ein Wertepaar für den Parameter der Datei `agent.properties` hinzu.

Es gibt drei Optionen für den Parameter **transferRecoveryTimeout**:

-1

Der Agent versucht weiterhin, die blockierte Übertragung wiederherzustellen, bis die Übertragung erfolgreich war. Diese Option entspricht dem Standardverhalten des Agenten, wenn die Eigenschaft nicht gesetzt wird.

0

Der Agent stoppt die Dateiübertragung, sobald die Wiederherstellung einsetzt.

>0

Der Agent wiederholt den Versuch, die blockierte Übertragung wiederherzustellen, bis der in Form eines positiven Integerwerts angegebene Zeitraum (in Sekunden) abgelaufen ist.

Alle Änderungen, die Sie an der `agent.properties`-Datei vornehmen, werden erst wirksam, nachdem der Agent erneut gestartet wurde.

Falls erforderlich, können Sie den Zeitlimitwert für die Übertragungswiederherstellung in der Datei `agent.properties` für eine individuelle Übertragung überschreiben. Weitere Informationen finden Sie unter [„Zeitlimit für Übertragungswiederherstellung für einzelne Übertragungen festlegen“](#) auf Seite 349.

Prozedur

- Wenn Sie angeben möchten, dass der Agent das Wiederherstellen der blockierten Übertragung fortsetzt, bis diese erfolgreich ausgeführt wird, legen Sie für das Zeitlimit für die Übertragungswiederherstellung den Wert `-1` fest, wie im folgenden Beispiel gezeigt wird:

```
transferRecoveryTimeout=-1
```

- Wenn Sie angeben möchten, dass der Agent die Übertragung direkt nach dem Eintreten der Wiederherstellung als fehlgeschlagen markiert, legen Sie für das Zeitlimit für die Übertragungswiederholung den Wert `0` fest, wie im folgenden Beispiel gezeigt wird:

```
transferRecoveryTimeout=0
```

- Wenn Sie angeben möchten, dass der Agent das Wiederholen einer blockierten Übertragung über einen bestimmten Zeitraum fortsetzt, bevor die Übertragung als fehlgeschlagen markiert wird, legen Sie für das Zeitlimit der Übertragungswiederherstellung den Wert für den Zeitraum in Sekunden fest, den Sie für den erneuten Versuch des Agenten verwenden möchten.

Wenn Sie beispielsweise den Wert 21600 als Zeitlimit für die Verbindungswiederherstellung festlegen, versucht der Agent, die Wiederherstellung der Übertragung bis 6 Stunden nach Beginn der Wiederherstellung durchzuführen:

```
transferRecoveryTimeout=21600
```

Der maximal zulässige Wert für diesen Parameter ist 999999999.

Zeitlimit für Übertragungswiederherstellung für einzelne Übertragungen festlegen

Sie können das Zeitlimit für die Übertragungswiederherstellung für eine einzelne Übertragung aus der Befehlszeile, mit dem IBM MQ Explorer oder mithilfe von Apache Ant-Tasks festlegen. Wenn in der `agent.properties`-Datei ein Zeitlimitwert für die Übertragungswiederherstellung festgelegt ist, überschreibt das Zeitlimit für die Übertragungswiederherstellung für eine Einzelübertragung den in der `agent.properties`-Datei festgelegten Wert.

Informationen zu diesem Vorgang

Sie können das Zeitlimit für die Übertragungswiederherstellung für eine einzelne Übertragung festlegen, wenn Sie die folgenden Schritte ausführen:

- Erstellen einer Übertragung mit dem Befehl **fteCreateTransfer** oder mit dem IBM MQ Explorer.
- Erstellen einer Übertragungsvorlage mit dem Befehl **fteCreateTemplate** oder mit dem IBM MQ Explorer.
- Erstellen einer Ressourcenüberwachung mit dem Befehl **fteCreateMonitor** oder mit dem IBM MQ Explorer.
- Kopieren oder Verschieben von Dateien mithilfe der `fte: filecopy` -oder `fte: filemove` Ant -Tasks.

Wenn Sie einen Zeitlimitwert für die Übertragungswiederherstellung für eine Einzelübertragung festlegen, überschreibt dieser Wert den Zeitlimitwert für die Übertragungswiederherstellung in der Datei `agent.properties` (siehe „[Zeitlimit für Übertragungswiederherstellung für alle Übertragungen für einen Quellenagenten festlegen](#)“ auf Seite 348).

Prozedur

- Wenn Sie den Zeitlimitwert für die Übertragungswiederherstellung mit dem Befehl **fteCreateTransfer** oder **fteCreateTemplate** festlegen möchten, geben Sie die entsprechende Option für den Parameter **-rt** an:

-1

Der Agent versucht weiterhin, die blockierte Übertragung wiederherzustellen, bis die Übertragung erfolgreich war. Diese Option entspricht dem Standardverhalten des Agenten, wenn die Eigenschaft nicht gesetzt wird.

0

Der Agent stoppt die Dateiübertragung, sobald die Wiederherstellung einsetzt.

>0

Der Agent wiederholt den Versuch, die blockierte Übertragung wiederaufzunehmen, bis der angegebene Zeitraum (in Sekunden) abgelaufen ist.

Beispiele für den Befehl **fteCreateTransfer**

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt -1 -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 0 -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt C:\export\originalfile.txt
```

Beispiele für den Befehl **fteCreateTemplate**

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt -1 -sa PAYROLL -sm QM_PAYROLL1 -da ACCOUNTS -dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 0 -sa PAYROLL -sm QM_PAYROLL1 -da ACCOUNTS -dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 21600 -sa PAYROLL -sm QM_PAYROLL1 -da ACCOUNTS -dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

Es gibt keinen Parameter **-rt** für den Befehl **fteCreateMonitor**. Wenn Sie den Parameter **-rt** mit dem Befehl **fteCreateTransfer** festlegen und den Parameter **-gt** ebenfalls festlegen, wird der Parameter für das Wiederherstellungszeitlimit mit der Übertragungsdefinition, die beim Ausführen des Befehls **fteCreateTransfer** generiert wurde, in das XML-Dokument eingeschlossen. Anschließend verwendet die Ressourcenüberwachung dieses XML-Dokument bei der Ausführung des Befehls **fteCreateMonitor**. Im folgenden Beispiel würden die Zeitlimitdetails für die Übertragungswiederherstellung in die `task.xml`-Datei aufgenommen:

```
fteCreateMonitor -ma AgentName -md C:\mqmft\monitors -mn Monitor_Name -mt task.xml -tr "fileSize>=5MB,*.zip"
```

- Wenn Sie den Zeitlimitwert für die Übertragungswiederherstellung auf der Seite mit dem IBM MQ Explorer-Assistenten für eine neue Übertragung, Überwachung oder Vorlage festlegen möchten, wählen Sie die erforderliche Option im Feld **Zeitlimit für Übertragungswiederherstellung** (in Sekunden) aus:

Als Quellenagent

Wenn Sie **Als Quellenagent** auswählen, wird der Wert des Parameters **transferRecoveryTimeout** aus der Datei `agent.properties` verwendet, falls dieser festgelegt ist. Andernfalls wird das Standardverhalten für das Zeitlimit für die Übertragungswiederherstellung angewendet.

Numerisches Listenfeld

Wenn Sie im numerischen Listenfeld einen Zeitwert in Sekunden angeben, versucht der Agent weiterhin, die blockierte Übertragung wiederaufzunehmen, bis der angegebene Zeitraum abgelaufen ist.

--

Wenn Sie **Keine** auswählen, wird kein Zeitlimit für die Übertragungswiederherstellung festgelegt, und der Agent versucht weiterhin, die blockierte Übertragung wiederherzustellen, bis sie erfolgreich war.

- Festlegen des Wiederherstellungszeitlimits mithilfe von Ant-Tasks. Schließen Sie die Option und den Wert **transferRecoveryTimeout** mit den Elementen **fte:filecopy** oder **fte:filemove** zum Verschieben oder Kopieren von Dateien ein, z. B.:

Beispiel für **fte:filecopy**

```
<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="copy.result" transferRecoveryTimeout="0">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filecopy>
```

Beispiel für **fte:filemove**

```
<fte:filemove cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
  src="agent1@qm1" dst="agent2@qm2"
  rcproperty="move.result" transferRecoveryTimeout="21600">
```

```
<fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filemove>
```

Zugehörige Konzepte

„Wiederherstellung und Neustart von MFT“ auf Seite 344

Sollte aus irgendeinem Grund Ihr Agent oder Warteschlangenmanager z. B. aufgrund eines Strom- oder Netzausfalls nicht verfügbar sein, führt Managed File Transfer eine Wiederherstellung gemäß folgender Szenarios durch:

Zugehörige Tasks

„Neue Dateiübertragung starten“ auf Seite 228

Sie können eine neue Dateiübertragung entweder in IBM MQ Explorer oder alternativ über die Befehlszeile starten. Sie können die Übertragung einer einzelnen Datei oder mehrerer Dateien in einer Gruppe auswählen.

„Dateiübertragungsvorlage mit IBM MQ Explorer erstellen“ auf Seite 274

Eine Dateiübertragungsvorlage können Sie in IBM MQ Explorer oder über die Befehlszeile erstellen. Sie können diese Vorlage verwenden, um mit den Vorlagendetails eine neue Dateiübertragung zu erstellen, oder um die Vorlage abzuschicken, um die Dateiübertragung zu starten.

„MFT-Ressourcen überwachen“ auf Seite 239

Sie können Managed File Transfer-Ressourcen überwachen, beispielsweise eine Warteschlange oder ein Verzeichnis. Sobald für diese Ressource eine festgelegte Bedingung eintritt, startet die Ressourcenüberwachung eine bestimmte Task, beispielsweise die Übertragung einer Datei. Ressourcenüberwachungen können mit dem Befehl **fteCreateMonitor** oder über die Ansicht **Überwachungen** im Managed File Transfer-Plug-in für IBM MQ Explorer erstellt werden.

„Status von Dateiübertragungen im Übertragungsprotokoll anzeigen“ auf Seite 237

Sie können im **Übertragungsprotokoll** in IBM MQ Explorer die Details zu Dateiübertragungen ansehen. Dabei spielt es keine Rolle, ob die Übertragungen über die Befehlszeile oder über IBM MQ Explorer gestartet wurden. Sie können den Inhalt des **Übertragungsprotokolls** anpassen.

Zugehörige Verweise

Die MFT `agent.properties`-Datei

fteCreateTransfer: Neue Dateiübertragung starten

fteCreateTemplate: Neue Dateiübertragungsschablone erstellen

fteCreateMonitor: MFT-Ressourcenüberwachung erstellen

`fte`: Ant-Task `filecopy`

`fte`: Ant-Task `'filemove'`

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 mithilfe von `MQSC`-Befehlen vollständig in der Befehlszeile verwaltet werden.

Die MQ Telemetry-Dokumentation enthält auch Beispielscripts, 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 Konzepte

[MQ Telemetry](#)

Zugehörige Verweise

[MQXR-Eigenschaften](#)

Linux

AIX

Warteschlangenmanager unter Linux und AIX für Telemetrie konfigurieren

Führen Sie die folgenden Schritte aus, um MQ Telemetry manuell zu konfigurieren. Wenn Sie nur eine einfache Konfiguration benötigen, die die Gastbenutzer-ID verwendet, können Sie stattdessen den MQ Telemetry Support Wizard in IBM MQ Explorer ausführen.

Vorbereitende Schritte

Wenn Sie nur eine einfache Konfiguration benötigen, ziehen Sie die Verwendung der MQ Telemetry -Unterstützung in IBM MQ Explorer in Betracht. Diese Unterstützung umfasst einen Assistenten und eine Beispielbefehlsprozedur `sampleMQM`. Diese Ressourcen 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.

Wenn Sie eine komplexere Konfiguration benötigen, die eine andere Authentifizierungsmethode verwendet, verwenden Sie die Schritte in dieser Task. Beginnen Sie mit den folgenden ersten Schritten:

1. Informationen zur Installation von IBM MQ und des Features MQ Telemetry finden Sie unter [Installationsvoraussetzungen für MQ Telemetry](#).
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

Führen Sie die Schritte in dieser Task aus, um MQ Telemetry unter Verwendung unterschiedlicher Berechtigungschemas manuell zu konfigurieren.

Vorgehensweise

1. Öffnen Sie ein Befehlsfenster auf dem Telemetriemusterverzeichnis.
Das Verzeichnis "telemetry samples" ist `/opt/mqm/mqxr/samples`.
2. Erstellen Sie die Übertragungswarteschlange für Telemetrie.

Wenn `SYSTEM.MQTT.TRANSMIT.QUEUE` nicht vorhanden ist, wird es automatisch erstellt, wenn der Telemetrieservice (MQXR) zum ersten Mal gestartet wird, und für die Verwendung der Gastbenutzer-ID festgelegt. Diese Task konfiguriert jedoch MQ Telemetry für die Verwendung eines anderen Berechtigungschemas. Für diese Task erstellen Sie `SYSTEM.MQTT.TRANSMIT.QUEUE` und konfigurieren den Zugriff darauf, bevor Sie den Telemetrieservice (MQXR) starten.

Führen Sie den folgenden Befehl aus:

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```


3. Legt die Standardübertragungswarteschlange fest.

Es ist einfacher, Nachrichten direkt an MQTT -Clients zu senden, wenn SYSTEM.MQTT.TRANSMIT.QUEUE die Standardübertragungswarteschlange ist. Andernfalls müssen Sie für jeden Client, der IBM MQ -Nachrichten empfängt (siehe [„Eine Nachricht direkt an einen Client senden“](#) auf Seite 357), eine Definition einer fernen Warteschlange hinzufügen. Beachten Sie, dass eine Änderung der Standardübertragungswarteschlange die vorhandene Konfiguration beeinträchtigen kann.

Wenn der Telemetrieservice (MQXR) zum ersten Mal gestartet wird, wird SYSTEM.MQTT.TRANSMIT.QUEUE nicht als Standardübertragungswarteschlange für den Warteschlangenmanager festgelegt. Zum Konfigurieren dieser Einstellung ändern Sie die Eigenschaft für die Standardübertragungswarteschlange. Verwenden Sie dazu entweder die IBM MQ Explorer oder führen Sie den folgenden Befehl aus:

```
echo "ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') " | runmqsc qMgr
```

4. Führen Sie eine Prozedur in [„MQTT-Clients für den Zugriff auf IBM MQ-Objekte berechtigen“](#) auf Seite 360 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. Bearbeiten Sie die Datei `installMQXRService_unix.mqsc`, um die Schlüsseldatei zu konfigurieren, die zum Verschlüsseln der Kennphrase für MQTT-TLS-Kanäle verwendet wird:

a) Öffnen Sie die Datei `WMQ program installation directory/mqxr/samples/installMQXRService_unix.mqsc`.

b) Suchen Sie die Zeile mit dem Parameter **STARTARG** und bearbeiten Sie die Option **-sf**, um die Position Ihrer Schlüsseldatei für Berechtigungsnachweise anzugeben.

Standardmäßig verwendet die Datei `installMQXRService_unix.mqsc` eine Standardschlüsseldatei namens [DEFAULT]. Die Standardschlüsseldatei ist für alle IBM MQ -Installationen identisch. Daher müssen Sie eine Schlüsseldatei angeben, die für Ihre Installation eindeutig ist, wenn Sie Kennphrasen verschlüsseln.

Siehe auch den Beispielcode in [„SYSTEM.MQXR.SERVICE erstellen“](#) auf Seite 354.

6. Installieren Sie den Telemetrieservice (MQXR), indem Sie den folgenden Befehl ausführen:

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc queue_manager
```

7. 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.

8. Konfigurieren Sie mithilfe von IBM MQ Explorer Telemetriedkanäle, sodass sie Verbindungen von MQTT-Clients akzeptieren.

Die Telemetriedkanäle müssen so konfiguriert werden, dass ihre Identitäten eine der in Schritt [„4“](#) auf Seite 353 definierten Benutzer-IDs sind.

Siehe auch [DEFINE CHANNEL \(MQTT\)](#).

9. Überprüfen Sie die Konfiguration, indem Sie den Beispielclient ausführen.

Damit der Beispielclient mit Ihrem Telemetriedkanal arbeiten kann, muss der Kanal den Client für die Veröffentlichung, Subskription und Empfang von Veröffentlichungen berechtigen. Der Beispielclient stellt die Verbindung zum Telemetriedkanal an Port 1883 standardmäßig her. Siehe auch [IBM MQ Telemetry Transport-Beispielprogramme](#).

SYSTEM.MQXR.SERVICE erstellen

Verwenden Sie den Befehl **runMQXRService** , um den SYSTEM.MQXR.SERVICE zu erstellen.

```
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+" -sf "/home/keyFileLocation/keyFile.txt"') +
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')
```

Windows Warteschlangenmanager unter Windows für Telemetrie konfigurieren

Führen Sie die folgenden Schritte aus, um MQ Telemetry manuell zu konfigurieren. Wenn Sie nur eine einfache Konfiguration benötigen, die die Gastbenutzer-ID verwendet, können Sie stattdessen den MQ Telemetry Support Wizard in IBM MQ Explorer ausführen.

Vorbereitende Schritte

Wenn Sie nur eine einfache Konfiguration benötigen, ziehen Sie die Verwendung der MQ Telemetry -Unterstützung in IBM MQ Explorer in Betracht. Diese Unterstützung umfasst einen Assistenten und eine Beispielbefehlsprozedur `sampleMQM`. Diese Ressourcen 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.

Wenn Sie eine komplexere Konfiguration benötigen, die eine andere Authentifizierungsmethode verwendet, verwenden Sie die Schritte in dieser Task. Beginnen Sie mit den folgenden ersten Schritten:

1. Informationen zur Installation von IBM MQ und des Features MQ Telemetry finden Sie unter [Installationsvoraussetzungen für MQ Telemetry](#) .
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

Führen Sie die Schritte in dieser Task aus, um MQ Telemetry unter Verwendung unterschiedlicher Berechtigungsschemas manuell zu konfigurieren.

Vorgehensweise

1. Öffnen Sie ein Befehlsfenster auf dem Telemetriemusterverzeichnis.

Das Verzeichnis "telemetry samples" ist `WMQ program installation directory\mqxr\samples`.

2. Erstellen Sie die Übertragungswarteschlange für Telemetrie.

Wenn SYSTEM.MQTT.TRANSMIT.QUEUE nicht vorhanden ist, wird es automatisch erstellt, wenn der Telemetrieservice (MQXR) zum ersten Mal gestartet wird, und für die Verwendung der Gastbenutzer-ID festgelegt. Diese Task konfiguriert jedoch MQ Telemetry für die Verwendung eines anderen Berechtigungschemas.

gungsschemas. Für diese Task erstellen Sie SYSTEM.MQTT.TRANSMIT.QUEUE und konfigurieren den Zugriff darauf, bevor Sie den Telemetrieservice (MQXR) starten.

Führen Sie den folgenden Befehl aus:

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```

3. Legt die Standardübertragungswarteschlange fest.

Es ist einfacher, Nachrichten direkt an MQTT -Clients zu senden, wenn SYSTEM.MQTT.TRANSMIT.QUEUE die Standardübertragungswarteschlange ist. Andernfalls müssen Sie für jeden Client, der IBM MQ -Nachrichten empfängt (siehe „Eine Nachricht direkt an einen Client senden“ auf Seite 357), eine Definition einer fernen Warteschlange hinzufügen. Beachten Sie, dass eine Änderung der Standardübertragungswarteschlange die vorhandene Konfiguration beeinträchtigen kann.

Wenn der Telemetrieservice (MQXR) zum ersten Mal gestartet wird, wird SYSTEM.MQTT.TRANSMIT.QUEUE nicht als Standardübertragungswarteschlange für den Warteschlangenmanager festgelegt. Zum Konfigurieren dieser Einstellung ändern Sie die Eigenschaft für die Standardübertragungswarteschlange. Verwenden Sie dazu entweder die IBM MQ Explorer oder führen Sie den folgenden Befehl aus:

```
echo ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

4. Führen Sie eine Prozedur in „MQTT-Clients für den Zugriff auf IBM MQ-Objekte berechtigen“ auf Seite 360 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. Bearbeiten Sie die Datei `installMQXRService_win.mqsc`, um die Schlüsseldatei zu konfigurieren, die zum Verschlüsseln der Kennphrase für MQTT-TLS-Kanäle verwendet wird:

a) Öffnen Sie die Datei `WMQ program installation directory\mqxr\samples\installMQXRService_win.mqsc`.

b) Suchen Sie die Zeile mit dem Parameter **STARTARG** und bearbeiten Sie die Option **-sf**, um die Position Ihrer Schlüsseldatei für Berechtigungsnachweise anzugeben.

Standardmäßig verwendet die Datei `installMQXRService_win.mqsc` eine Standardschlüsseldatei namens [DEFAULT]. Die Standardschlüsseldatei ist für alle IBM MQ -Installationen identisch. Daher müssen Sie eine Schlüsseldatei angeben, die für Ihre Installation eindeutig ist, wenn Sie Kennphrasen verschlüsseln.

Siehe auch den Beispielcode in „Wird erstellt SYSTEM.MQXR.SERVICE“ auf Seite 356.

6. Installieren Sie den Telemetrieservice (MQXR), indem Sie den folgenden Befehl ausführen:

```
type installMQXRService_win.mqsc | runmqsc qMgr
```

7. 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.

8. Konfigurieren Sie mithilfe von IBM MQ Explorer Telemetriekanäle, sodass sie Verbindungen von MQTT-Clients akzeptieren.

Die Telemetriekanäle müssen so konfiguriert werden, dass ihre Identitäten eine der in Schritt „4“ auf Seite 355 definierten Benutzer-IDs sind.

Siehe auch [DEFINE CHANNEL \(MQTT\)](#).

9. Überprüfen Sie die Konfiguration, indem Sie den Beispielclient ausführen.

Damit der Beispielclient mit Ihrem Telemetrikkanal arbeiten kann, muss der Kanal den Client für die Veröffentlichung, Subskription und Empfang von Veröffentlichungen berechtigen. Der Beispielclient stellt die Verbindung zum Telemetrikkanal an Port 1883 standardmäßig her. Siehe auch [IBM MQ Telemetry Transport-Beispielprogramme](#).

Wird erstellt SYSTEM.MQXR.SERVICE

Verwenden Sie den Befehl `runMQXRService`, um den SYSTEM.MQXR.SERVICE zu erstellen.

```
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+\" -sf "c:\keyFileLocati
on\keyFile.txt"') +
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')
```

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 Telemetri- eservice (MQXR) an den Client gesendet. Es gibt eine Reihe von Möglichkeiten, eine Nachricht in SYS- TEM.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 unter SYSTEM.MQTT.TRANSMIT.QUEUE an MQTT -Clients weiter und verwendet dabei den Namen des Zielwarteschlangenmanagers als Schlüssel für einen bestimmten Client.

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. Unter Verwendung der Standardobjektnamensauflösung von IBM MQ wird `ClientIdentifier` wie folgt aufgelöst (siehe [Tabelle 16 auf Seite 357](#)).

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. *ClientIdentifizier* stimmt mit einem WS-Manager-Aliasnamen mit dem Namen *ClientIdentifizier* überein

ClientIdentifizier ist ein Name des fernen Warteschlangenmanagers. Er stimmt mit dem Namen eines WS-Manager-Aliasnamens überein.

Der Aliasname des WS-Managers muss mit *ClientIdentifizier* 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.

Tabelle 16. Namensauflösung eines MQTT-Warteschlangenmanager-Aliasnamens					
	Eingabe		Ausgabe		
<i>ClientIdentifizier</i>	Name des Warteschlangenmanagers	Warteschlangenname	Name des Warteschlangenmanagers	Warteschlangenname	Übertragungswarteschlange
Stimmt mit nichts überein	<i>ClientIdentifizier</i>	<i>undefined</i>	<i>ClientIdentifizier</i>	<i>undefined</i>	Standardübertragungswarteschlange. SYSTEM.MQTT.TRANSMIT.QUEUE
Übereinstimmung mit einem WS-Manager-Aliasnamen mit dem Namen <i>ClientIdentifizier</i>	<i>ClientIdentifizier</i>	<i>undefined</i>	<i>ClientIdentifizier</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
```

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 (`ClientIdentifizier`) des MQTT v3-Clients kennen. Da MQTT v3 -Clients keine Warteschlangen haben, wird der Name der Zielwarteschlange als Topicname an die Methode MQTT v3 application client `messageArrived` übergeben. Erstellen Sie zum Beispiel in einem MQI-Programm einen Objektdeskriptor mit dem Client als `ObjectQmgrName` :

```
MQOD.ObjectQmgrName = ClientIdentifizier ;
MQOD.ObjectName = name ;
```

Wenn die Anwendung unter Verwendung von JMS geschrieben wird, erstellen Sie ein Punkt-zu-Punkt-Ziel; Beispiel:

```
JM 3.0
jakarta.jms.Destination jmsDestination =
(jakarta.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifizier/name");
```

```
JMS 2.0
javax.jms.Destination jmsDestination =
(javax.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifizier/name");
```

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 (`ClientIdentifizier`) des Clients aufgelöst werden. Die Übertragungswarteschlange muss in `SYSTEM.MQTT.TRANSMIT.QUEUE` aufgelöst werden (siehe [Tabelle 17 auf Seite 358](#)). 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	<i>ClientIdentifi- fier</i>	SYSTEM.MQTT. TRANSMIT.QUEUE

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 Sitzungen](#). 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 lichste` 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.

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 Clientidentität. Je nach Kontext wird der Client durch die Client-ID, Benutzernamen, eine vom Administrator erstellte gemeinsame Clientidentität 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 Benutzernamen 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 Clientidentität 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, zeitbegrenzttes 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, ist 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
```

Windows Linux AIX **MQTT-Clients für den Zugriff auf IBM MQ-Objekte berechtigen**

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. `ClientIdentifier`
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 Telemetrikkanäle mit jeweils einer anderen Benutzer-ID, die der Administrator mit IBM MQ Explorer zuordnet. Clients, die unter Verwendung unterschiedlicher TCP/IP-Port-

nummern eine Verbindung herstellen, sind verschiedenen Telemetriedatenkanälen zugeordnet und weisen unterschiedliche Identitäten auf.

2. Definieren Sie einen einzelnen Telemetriedatenkanal, aber lassen Sie jeden Client einen Benutzernamen aus einer kleinen Gruppe von Benutzer-IDs auswählen. Der Administrator konfiguriert den Telemetriedatenkanal, um den Client `Username` als Identität auszuwählen.

In dieser Task wird die Identität des Telemetriedatenkanals `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 361
- „Coarse-Grained-Zugriffssteuerung“ auf Seite 361
- „Mittlere Zugriffssteuerung“ auf Seite 361
- „Feinkörnige Zugriffssteuerung“ auf Seite 362

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 `mqm`-Gruppe hinzu; siehe [Benutzer einer Gruppe unter Windows hinzufügen](#) oder [Gruppen unter Linux erstellen und verwalten](#).

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*, Themen an MQTT-Clients zu senden.

```
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
```

Windows Linux AIX MQTT-Clientauthentifizierung über ein Kennwort

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 subskribieren.

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 zum Veröffentlichen und Subskribieren eines Themas:

```
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 für die Authentifizierung des Benutzernamens des Clients unter Verwendung von JAAS.

Windows

Linux

AIX

MQTT-Clientauthentifizierung 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.

Indem Sie dem Client ein privat signiertes digitales Zertifikat zur Verfügung stellen, können Sie den MQTT-Client authentifizieren, um IBM MQ . Der IBM MQ -Administrator kann MQTT-Clients zwingen, sich 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 Telemetriedkanals als vertrauenswürdiges Zertifikat hinzugefügt wurde.

Anmerkung: Der Telemetriedkanal verfügt über eine Kombination aus Keystore/Truststore, die sowohl private Schlüssel für einen oder mehrere Telemetriedkanäle 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 Telemetrikkanal und den Client konfigurieren.

Zugehörige Konzepte

„Konfiguration des Telemetrikkanals für MQTT-Clientauthentifizierung mit TLS“ auf Seite 364

Der IBM MQ Administrator konfiguriert Telemetrikkanäle 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.

[MQTT-Clientkonfiguration für Clientauthentifizierung mit TLS](#)

Konfiguration des Telemetrikkanals für MQTT-Clientauthentifizierung mit TLS

Der IBM MQ Administrator konfiguriert Telemetrikkanäle 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.

Legen Sie die Eigenschaft `com.ibm.mq.MQTT.ClientAuth` eines TLS-Telemetrikkanals 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 Telemetrikkanal-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 369

Zugehörige Konzepte

[„Telemetriekanal-Konfiguration für Kanalauthentifizierung mit TLS“](#) auf Seite 366

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.

[CipherSpecs und CipherSuites](#)

Zugehörige Verweise

[DEFINE CHANNEL \(MQTT\)](#)

[Kanal ändern \(MQTT\)](#)

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 vertrauenswürdiges Netz aufgebaut. MQTT-Clients können auch mit TCP/IP über das VPN mit Telemetriekanälen 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, indem er die vom Server gesendeten Zertifikate mit den in seinem Truststore oder in seiner JRE abgelegten Zertifikaten abgleicht. cacerts speichern.

Der JRE-Zertifikatspeicher 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-Zertifikatspeicher oder im Truststore des Clients speichern. Sie können jedoch nicht beide Speicher verwenden. Verwenden Sie den Client-Truststore, wenn Sie die öffentlichen Zertifikate, denen der Client vertraut, getrennt von den Zertifikaten anderer Java Anwendungen verwenden. Verwenden Sie den JRE-Zertifikatspeicher, wenn Sie einen gemeinsamen Zertifikatspeicher für alle Java auf dem Client ausgeführte Anwendungen. Wenn Sie sich für die Verwendung des JRE-Zertifikatspeichers 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 Telemetrikkanals mit TLS müssen Sie den Server und den Client konfigurieren.

Windows Linux AIX **Telemetrikkanal-Konfiguration für Kanalauthentifizierung mit TLS**

Der IBM MQ Administrator konfiguriert Telemetrikkanäle 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 Telemetrikkanal 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 Telemetrikkanal mit dem IBM MQ-Explorer für die Verwendung von TLS. Geben Sie den Pfad zum Keystore und den Verschlüsselungstext für den Zugriff auf den Schlüsselspeicher an. Wenn Sie die TCP/IP-Portnummer des Kanals nicht festlegen, nimmt die TLS-Telemetrikanschlussnummer 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 369

Wichtig:   Ab IBM MQ 9.4.0 werden CMS -Schlüsselrepositores und Stashdateien nicht mit AMQP- und MQTT-Kanälen unterstützt, die SSL/TLS verwenden. Verwenden Sie PKCS #12 -Schlüsselrepositores und schützen Sie die Kennwörter des Schlüsselrepositores mithilfe des IBM MQ -Kennwortschutzsystems. Sie können ein PKCS #12 -Schlüsselrepository mit folgendem Befehl erstellen:

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

Dieser Befehl erstellt eine PKCS #12 -Schlüsselrepositorydatei namens *filename.p12*, die mit dem angegebenen Kennwort gesichert wird.

Zugehörige Konzepte

[„Konfiguration des Telemetrikkanals für MQTT-Clientauthentifizierung mit TLS“](#) auf Seite 364

Der IBM MQ Administrator konfiguriert Telemetrikkanäle 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.

[CipherSpecs und CipherSuites](#)

Zugehörige Verweise

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)



Beispiel für eine MQTT-Kanalkonfiguration mit TLS-Authentifizierung

Dieses Beispiel führt Sie durch ein Beispiel für die Konfiguration eines MQTT -Kanals, der TLS-Authentifizierung verwendet.

Das Beispiel konfiguriert einen Kanal zwischen MQTT und Mosquitto.

Das Beispiel verwendet einen Docker -Container für IBM MQ unter Red Hat Enterprise Linux und Mosquitto unter CentOS, gilt jedoch für jeden Servertyp. (CentOS wurde aufgrund von Registry-Berechtigungen für Mosquitto verwendet).



IBM MQ -Keystore und -Kanal für unidirektionale TLS konfigurieren

Wichtig:   Ab IBM MQ 9.4.0 werden CMS -Schlüsselrepositorys und Stashdateien nicht mit AMQP- und MQTT-Kanälen unterstützt, die SSL/TLS verwenden. Verwenden Sie PKCS #12 -Schlüsselrepositorys und schützen Sie die Kennwörter des Schlüsselrepositorys mithilfe des IBM MQ -Kennwortschutzsystems.

Führen Sie die folgenden Schritte aus:

1.   Erstellen Sie den IBM MQ -Schlüsselspeicher:

```
runmqakm -keydb -create -db mqtt.p12 -pw "passw0rd" -type p12
```

2.   Erstellen Sie ein persönliches Zertifikat:

```
runmqakm -cert -create -db mqtt.p12 -pw "passw0rd" -size 2048 -dn "CN= mqm, OU=MQTest, O=MQSuppor, C=US" -sig_alg SHA256_WITH_RSA -label ibmwebspheremqmqm
```

Mit dem folgenden Befehl können Sie die Erstellung des Zertifikats bestätigen:

```
runmqakm -cert -list -v -db mqtt.p12 -pw "passw0rd"
```

3.   Erstellen Sie den MQTT -Kanal, indem Sie den folgenden Befehl an der runmqsc-Eingabeaufforderung eingeben:

```
DEFINE CHANNEL(MQTTDEMO) CHLTYPE(MQTT) BACKLOG(4096) PORT(8883) MCAUSER('mqm') PRO□  
TOCOL(MQTTV311,MQTTV3,HTTP) SSLCAUTH(OPTIONAL) SSLCIPH('SSL_RSA_WITH_AES_256_CBC_SHA256')  
SSLKEYR('/var/mqm/mqtt/mqtt.p12') SSLKEYP('passw0rd') TRPTYPE(TCP)
```

Beachten Sie, dass der Kanal Java -Verschlüsselungszuordnungen verwendet, siehe [TLS CipherSpecs](#) und [CipherSuites in IBM MQ Classes for JMS](#).

4. Extrahieren Sie das Zertifikat:

```
runmqakm -cert -extract -db mqtt.kdb -stashed -label ibmwebspheremqmqm -target serverCert.pem
```

Mosquitto unter CentOS in einem Docker -Container installieren

Führen Sie die folgenden Schritte aus, um einen Docker -Container mit Mosquitto unter CentOS zu erstellen:

1. `docker pull centos`
2. `docker run -it centos /bin/bash`
3. `yum -y install epel-release`
4. `yum -y install mosquitto`

Unterzeichnerzertifikat in Mosquitto verschieben

Führen Sie die folgenden Schritte aus, um das Zertifikat, das Sie in IBM MQ erstellt haben, in Mosquitto zu verschieben. Diese Schritte werden auf der Docker -Hostmaschine ausgeführt.

1. Zeigen Sie die Container-IDs in Docker an:

```
docker container ls
```

2. Kopieren Sie die Datei aus dem Docker-Container in Ihr Docker-System auf dem lokalen System.

```
cp MQ_Container_ID:/var/mqm/mqtt/serverCert.pem serverCert.pem
```

3. Kopieren Sie die Datei von Ihrer lokalen Maschine in das Stammverzeichnis auf der CentOS -Maschine:

```
docker cp serverCert.pem CentOS_ContainerID:/serverCert.pem
```

Mit Mosquitto veröffentlichen

Veröffentlichen Sie eine Testnachricht in Mosquitto mit dem folgenden Befehl:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d
```

Die Befehlsargumente haben folgende Bedeutung:

-h

Die IP-Adresse des Red Hat Enterprise Linux -Hosts (kann mit **nslookup** gefunden werden).

-- cafile

Die Datei, die das Unterzeichnerzertifikat enthält.

-- unsicher

Diese Option wird angegeben, da das Beispiel ein selbst signiertes Zertifikat verwendet. Verwenden Sie diese Option nicht, wenn Sie echte CA-Zertifikate verwenden.

-p

Portnummer.

-i

Client-ID.

-t

Das Thema, das veröffentlicht wird.

-m

Die Nachricht, die veröffentlicht wird.

-d

Debugnachrichten aktivieren.

Konfigurieren Sie den MQTT -Kanal für die gegenseitige TLS-Authentifizierung.

Geben Sie den folgenden Befehl ein, um den MQTT -Kanal als SSLCAUTH (REQUIRED) zu rekonfigurieren.

```
ALTER CHANNEL(MQTTDEMO) CHLTYPE(MQTT) SSLCAUTH(REQUIRED)
```

Erstellen Sie ein Schlüssel/Zertifikat-Paar auf dem Mosquitto-Server und fügen Sie IBM MQ hinzu.

Geben Sie die folgenden Befehle ein, um das Schlüssel/Zertifikat-Paar auf Mosquitto zu erstellen:

1. Verwenden Sie **openssl** , um das Schlüssel/Zertifikat-Paar für Mosquitto zu erstellen:

```
openssl req -x509 -newkey rsa:4096 -keyout mosquittoKey.pem -out mosquittoCert.pem -subj "/" CN=Mosquitto"
```

2. Listen Sie die Container-IDs für die Container auf:

```
docker container ls
```

3. Kopieren Sie das Mosquitto-Zertifikat in den Docker Ihres lokalen Systems:

```
docker cp CentOS_ContainerID:mosquittoCert.pem .
```

4. Kopieren Sie das Mosquitto-Zertifikat nach IBM MQ:

```
docker cp mosquittoCert.pem MQ_Container_ID:/var/mqm/mqtt
```

5. Fügen Sie das Zertifikat zum IBM MQ -Keystore hinzu:

```
runmqakm -cert -add -db mqtt.kdb -stashed -file mosquittoCert.pem
```

6. Starten Sie den MQTT -Kanal erneut.

Veröffentlichen mit Mosquitto und gegenseitiger Authentifizierung

Führen Sie die folgenden Schritte aus, um mit Mosquitto mit gegenseitiger Authentifizierung zu veröffentlichen.

1. Der folgende Befehl sollte eine Testnachricht erfolgreich veröffentlichen:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d --cert mosquittoCert.pem --key mosquittoKey.pem
```

2. Der folgende Befehl sollte eine Testnachricht nicht veröffentlichen und eine Fehlermeldung generieren, da er kein persönliches Zertifikat von Mosquitto sendet:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d /var/mqm/qmgrs/mqttDemoQM/errors/ mqxr_0.log
```

Zugehörige Informationen

[Schlüssel und Zertifikate verwalten](#)

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) enthält, ist die minimale Java-Version Java 7 aus 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

[Telemetrieservice \(MQXR\)](#)

„[Telemetriekanal-Konfiguration für Kanalauthentifizierung mit TLS](#)“ auf Seite 366

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 Verweise

[DEFINE CHANNEL \(MQTT\)](#)

[Kanal ändern \(MQTT\)](#)

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 365.

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

Telemetriekanälen

Konfigurieren Sie TLS, um den Telemetriekanal 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 Telemetriekanäle 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 Telemetriekanälen verbunden werden.

Sie können die Verbindung zwischen einem Java-MQTT-Client und einem Telemetriekanal 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 Telemetriekanal. Verschlüsseln Sie Nachrichten zwischen dem Client und dem Warteschlangenmanager (siehe „[MQTT-Clientauthentifizierung mit TLS](#)“ auf Seite 363).
2. Verbinden Sie einen MQTT-Client nur mit einem vertrauenswürdigen Telemetriekanal. Verschlüsseln Sie Nachrichten zwischen dem Client und dem Warteschlangenmanager; siehe „[Telemetriekanal-Authentifizierung mit TLS](#)“ auf Seite 365.
3. Verschlüsseln Sie Nachrichten zwischen dem Client und dem Warteschlangenmanager; siehe „[Datenschutz auf Telemetriekanälen](#)“ auf Seite 369.

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 Telemetriedatenkanalparameter 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 Telemetriedatenkanals 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", "clientId1");
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 authentifiziert gegen Zertifikate im Kanalschlüsselspeicher und im `JRE-cacerts` Speicher.

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 Eigenschaftsanzeige des Warteschlangenmanagers oder mithilfe von `runmqsc` `SSLFIPStest`. `SSLFIPStest` 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 Eigenschaftsanzeige des Warteschlangenmanagers oder mithilfe von `runmqsc` fest. `SSLCRLNL` gibt eine Namensliste mit Authentifizierungsinformationsobjekten an, die zum Angeben von Zertifikatswiderufspositionen 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](#).

Protokoll

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, denen Sie vertrauen, im JRE- `cacerts` Speicher platzieren.

Konfigurieren Sie den Truststore, wenn Sie einen anderen Truststore für den Client verwenden möchten. Möglicherweise müssen Sie den Truststore nicht konfigurieren, wenn der Server ein Zertifikat verwendet, das von einer bekannten Zertifizierungsstelle ausgestellt wurde, deren Stammzertifikat bereits in `cacerts` gespeichert ist.

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:

Tabelle 18. Dateipfade nach Plattform für JRE-TLS-Konfigurationsdateien	
Plattform	Dateipfad
Windows	WMQ Installation Directory\java\jre\lib\security
AIX and Linux-Plattformen	WMQ Installation Directory/java/jre64/jre/lib/security

Well-bekannte Zertifizierungsstellen

Die Datei cacerts enthält die Stammzertifikate namhafter Zertifizierungsstellen. Der cacerts wird standardmäßig verwendet, es sei denn, Sie geben einen Truststore an. Wenn Sie den cacerts-Speicher verwenden oder keinen Truststore bereitstellen, müssen Sie die Liste der Unterzeichner in cacerts überprüfen und bearbeiten, um die Sicherheitsanforderungen zu erfüllen.

 Mit dem Befehl **runmqktool** können Sie die cacerts-Zertifikatsdatei verwalten. cacerts ist eine JKS-Datei. Geben Sie den Parameter **-storetype jks** an, wenn der Befehl **runmqktool** zum Verwalten der Zertifikatsdatei verwendet wird.

 Das Standardkennwort für die Datei cacerts lautet changeit. Ändern Sie das Kennwort mit dem Befehl **runmqktool -storepasswd**, um die Datei zu schützen.

Sicherheitsklassen konfigurieren

Verwenden Sie die Datei `java.security`, um zusätzliche Sicherheitsprovider und andere Standardsicherheitseigenschaften zu registrieren.

Berechtigungen

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. Laden Sie bei Bedarf starke, aber eingeschränkte Standortinformationen von [IBM Developer Kits](#), Sicherheitsinformationen 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 16 auf Seite 374](#) 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 16. 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 17. *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 18. *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 19. *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\qmgrs\qMgrName\mqxr\jaas.config* definiert.

Die *jaas.config* -Datei wird nach dem JAAS -Konfigurationsnamen organisiert. Unter jedem Konfigurationsnamen befindet sich eine Liste mit Anmeldekongfigurationen; siehe „[Beispiel jaas.config -Datei](#)“ auf [Seite 376](#).

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 Telemetrieservice (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 Beispielm modul (`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 Telemetrieservice (MQXR) befinden. Stellen Sie Ihre Klassen nicht in IBM MQ-Verzeichnisse, die sich im Klassenpfad befinden. Erstellen Sie Ihre eigenen Verzeichnisse und definieren Sie den gesamten Klassenpfad für den Telemetrieservice (MQXR).

Sie können den Klassenpfad erweitern, der vom Telemetrieservice (MQXR) verwendet wird, indem Sie den Klassenpfad in der Datei `service.env` festlegen. `CLASSPATH` muss aktiviert sein, und die Klassenpfadanweisung kann nur Literale enthalten. Sie können keine Variablen im `CLASSPATH` verwenden, z. B. `CLASSPATH=%CLASSPATH%` ist falsch. Der Telemetrieservice (MQXR) legt seinen eigenen Klassenpfad fest. Der in `service.env` definierte `CLASSPATH` wird diesem 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 „[JAASLoginModule.Login\(\)-Beispielmethode](#)“ auf Seite 376 finden Sie ein Beispiel dafür, wie auf den Benutzernamen und das Kennwort zugegriffen wird.

Beispiel jaas.config-Datei

Ein Beispiel für eine JAAS-Konfigurationsdatei mit der 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";
};
```

JAASLoginModule.Login() -Beispielmethode

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;
}

```

Zugehörige Tasks

Problembehebung: [Das JAAS-Anmeldemodul wird vom Telemetrieservice nicht aufgerufen](#)

Zugehörige Verweise

[AuthCallback-MQXR-Klasse](#)

AMQP-Client verwalten

Sie können einen AMQP-Client über IBM MQ Explorer oder in einer Befehlszeile verwalten. Im Explorer können Sie Kanäle konfigurieren und AMQP-Clients überwachen, die mit IBM MQ verbunden sind. Konfigurieren Sie die Sicherheit von AMQP-Clients mit TLS und JAAS.

Vorbereitungen

Weitere Informationen zur Installation von AMQP auf Ihrer Plattform finden Sie unter [Zu installierende Installation](#).

Verwaltung mit IBM MQ Explorer

Mit dem Explorer können Sie AMQP-Kanäle konfigurieren und die AMQP-Clients überwachen, die mit IBM MQ verbunden sind. Sie können die Sicherheit von AMQP-Clients mit TLS und JAAS konfigurieren.

Verwaltung über die Befehlszeile

Sie können einen AMQP-Client über die Befehlszeile [mit MQSC-Befehlen](#) verwalten.

Windows V9.4.0 Linux V9.4.0 AIX **AMQP-Service wird beim Start des Warteschlangenmanagers nicht automatisch gestartet**

Ab IBM MQ 9.4.0 hat sich das Standardverhalten der Einstellung des Attributs **CONTROL** zum Starten des AMQP-Service geändert. Beim Erstellen und Starten eines neuen Warteschlangenmanagers wird der AMQP-Service nicht automatisch als Teil des Startprozesses des Warteschlangenmanagers gestartet.

Zwischen IBM MQ 9.0.4 und IBM MQ 9.4.0 ist das Standardverhalten der Einstellung des Attributs **CONTROL** zum Starten des AMQP-Service QMGR.

Wenn die AMQP-Komponente installiert wurde, wurde der AMQP-Service automatisch gestartet, auch wenn er nicht verwendet wurde. Um den Standardstart der AMQP Java Virtual Machine (JVM) zu vermeiden, haben Sie zwei Optionen:

- Die AMQP-Komponente wird nicht installiert oder
- Ändern Sie das AMQP-Serviceattribut **CONTROL** nach dem Start des Warteschlangenmanagers in MANUAL .

Ab IBM MQ 9.4.0 haben neu erstellte Warteschlangenmanager die Einstellung des Attributs **CONTROL** von SYSTEM.AMQP.SERVICE auf MANUAL, die Standardeinstellung vor IBM MQ 9.0.4.

Migrierte Warteschlangenmanager fahren bei Verwendung von AMQP mit dem automatischen Start des Service beim Start des Warteschlangenmanagers fort. Um festzustellen, ob AMQP verwendet wurde, werden folgende Punkte überprüft:

- Vorhandene AMQP-Kanäle
- Nachrichten zu Kanalstarts im AMQP-Fehlerprotokoll.



Achtung:

- Dies geschieht nur einmal; beim ersten Start des Warteschlangenmanagers nach einem Upgrade.
- Wenn das Attribut **CONTROL** während der Migration von QMGR in MANUAL geändert wird, wird eine Informationsnachricht im IBM MQ -Fehlerprotokoll aufgezeichnet, um die Änderung anzuzeigen. Weitere Informationen finden Sie unter Position von AMQP-Protokollen, Fehlerprotokollen und Konfigurationsdateien .

Wenn der AMQP-Service automatisch gestartet werden soll, ändern Sie das Attribut **CONTROL** des Service in QMGR und starten Sie den Warteschlangenmanager neu. Nachfolgende Neustarts des Warteschlangenmanagers starten den AMQP-Service.

Von AMQP-Clients verwendete IBM MQ-Objekte anzeigen

Sie können die verschiedenen IBM MQ-Ressourcen anzeigen, die von AMQP-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 Hconns-Pool wird verwendet, wenn AMQP-Clients Nachrichten veröffentlichen. Sie können die Hconns mit dem Befehl **DISPLAY CONN** anzeigen. For example:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (APPLDESC LK 'IBM 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 AMQP-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 AMQP-Client erstellt hat. Sie können den von einem bestimmten AMQP-Client verwendeten Hconn mit dem Befehl **DISPLAY CONN** anzeigen. For example:

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_abcd1234')
```

Subskriptionen, die von Clients erstellt werden

Wenn ein AMQP-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 lautet `private`, wenn der Client eine nicht gemeinsam genutzte Subskription erstellt hat, oder `share`, wenn der Client einer gemeinsam genutzten Subskription beigetreten ist.

Um die Subskriptionen anzuzeigen, die von einem bestimmten AMQP-Client verwendet werden, führen Sie den Befehl **DISPLAY SUB** aus und filtern Sie das 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 AMQP-Clients verwendet werden können, können Sie die Clients anzeigen, die derzeit Nachrichten von 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. For example:

```
DISPLAY SUB('/:private:recv_e298452:public') DEST
5 : DISPLAY SUB('/:private:recv_e298452:public') DEST
AMQ8096: WebSphere MQ subscription inquired.
SUBID(414D5120514D31202020202020202020707E0A565C2D0020)
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(414D5143514D31202020202020202020)
TYPE(HANDLE)

OBJNAME(SYSTEM.BASE.TOPIC)          OBJTYPE(TOPIC)

OBJNAME(SYSTEM.MANAGED.DURABLE.560A7E7020002961)
OBJTYPE(QUEUE)
```

3. Zeigen Sie für jede Kennung die AMQP-Client-ID an, für die die Kennung geöffnet ist:

```
DISPLAY CONN(707E0A56642B0020) CLIENTID
23 : DISPLAY CONN(707E0A56642B0020) CLIENTID

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_8f02c9d)
DISPLAY CONN(707E0A565F290020) CLIENTID
24 : DISPLAY CONN(707E0A565F290020) CLIENTID
AMQ8276: Display Connection details.
CONN(707E0A565F290020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_86d8888)
```

AMQP-Clientidentifikation, -Autorisierung 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 Kanalauthentifizierungsregeln 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.

AMQP-Clientidentität und -berechtigung

Verwenden Sie die AMQP-Client-ID, den AMQP-Benutzernamen oder eine allgemeine Clientidentität, die im Kanal oder in einer Kanalauthentifizierungsregel definiert ist, für die Berechtigung zum Zugreifen auf IBM MQ-Objekte.

Der Administrator kann bei der Definition oder Änderung des AMQP-Kanals die Auswahl treffen, indem er die Einstellung des Warteschlangenmanagers CONNAUTH oder die Kanalauthentifizierungsregeln 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 Kanalauthentifizierungsregel.

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. Die folgenden Befehle berechtigen beispielsweise eine Kanalidentität AMQPClient, die vom Administrator des Warteschlangenmanagers 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
```

AMQP-Clientauthentifizierung mit einem Kennwort

Authentifizieren Sie den Benutzernamen des AMQP-Clients mit dem Clientkennwort. 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 subskribieren.

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.

Das folgende Verfahren beschreibt beispielhaft die Schritte zur Authentifizierung einzelner Benutzer mit den lokalen Betriebssystembenutzern und -kennwörtern und, falls erfolgreich, zur Übernahme der gemeinsamen Identität AMQPUser:

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 zum Veröffentlichen und Subskribieren eines Themas:

```
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 CHCKCLNT (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.

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 Kanalverschlüsselung 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.

AMQP-Clients mit TLS konfigurieren

Sie können AMQP-Clients für die Verwendung von TLS konfigurieren, um Daten zu schützen, die über das Netz fließen, und um die Identität des Warteschlangenmanagers zu authentifizieren, zu dem der Client eine Verbindung herstellt.

Wenn Sie TLS für die Verbindung von einem AMQP-Client zu einem AMQP-Kanal verwenden möchten, 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

Mit dem Attribut SSLCAUTH wird angegeben, ob der Warteschlangenmanager einen AMQP-Client benötigen soll, um ein Clientzertifikat zur Verifizierung seiner Identität vorzulegen.

SSLCIPH

Das Attribut SSLCIPH gibt die Verschlüsselung an, die der Kanal zum Codieren von Daten in den TLS-Fluss verwenden soll.

V 9.4.0 Ab IBM MQ 9.4.0 unterstützen AMQP-Kanäle ANY* generische CipherSpecs. Weitere Informationen zu CipherSpecs finden Sie im Abschnitt [CipherSpecs aktivieren](#).

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 Schlüsselspeicher 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 AMQP-Client eine Verbindung zu einem AMQP-Kanal herstellt, der durch TLS geschützt ist, überprüft der Client die Identität des vom Warteschlangenmanager vorgelegten Zertifikats. Dazu müssen Sie Ihren AMQP-Client mit einem Truststore konfigurieren, der das Zertifikat des Warteschlangenmanagers enthält. Die dafür erforderlichen Schritte variieren je nach verwendetem AMQP-Client. Informationen zu den verschiedenen AMQP-Clients und -APIs finden Sie in der entsprechenden AMQP-Clientdokumentation.

Zugehörige Verweise

[DEFINE CHANNEL](#) (Definieren eines neuen Kanals)

[STOP SERVICE](#) (Service stoppen) unter [Multiplatforms](#)

[START SERVICE](#) (Service starten) unter [Multiplatforms](#)

Verbindung zwischen AMQP-Clients und dem Warteschlangenmanager trennen

Wenn Sie die Verbindung zwischen AMQP-Clients und dem Warteschlangenmanager trennen möchten, führen Sie entweder den Befehl PURGE CHANNEL aus oder stoppen Sie die Verbindung zum AMQP-Client.

- Führen Sie den Befehl **PURGE CHANNEL** aus. For example:

```
PURGE CHANNEL(MYAMQP) CLIENTID('recv_28dbb7e')
```

- Alternativ können Sie die Verbindung, die der AMQP-Client zum Trennen des Clients verwendet, stoppen, indem Sie die folgenden Schritte ausführen:

1. Suchen Sie die Verbindung, die der Client verwendet, indem Sie den Befehl **DISPLAY CONN** ausführen. For example:

```
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(414D5143514D312020202020202020)
TYPE (CONN)
CLIENTID(recv_28dbb7e)
```

2. Stoppen Sie die Verbindung. For example:

```
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:

```
amqssubc Sports QM1
```

Dabei sind *Sports* und *QM1* die gleichen wie in Schritt „2“ auf Seite 385.

4. Geben Sie `Hello world` in Befehlszeile 1 ein. Wenn der Port und die IP-Adresse, die im COMMINFO-Objekt angegeben sind, ordnungsgemäß konfiguriert sind, gibt das `amqssubc`-Beispiel, das am Port für Veröffentlichungen von der angegebenen Adresse empfangsbereit ist, `Hello world` in 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 Multicast-Bereich der IP-Adresse lautet 224.0.0.0 bis 239.255.255.255, aber einige dieser Adressen sind 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.

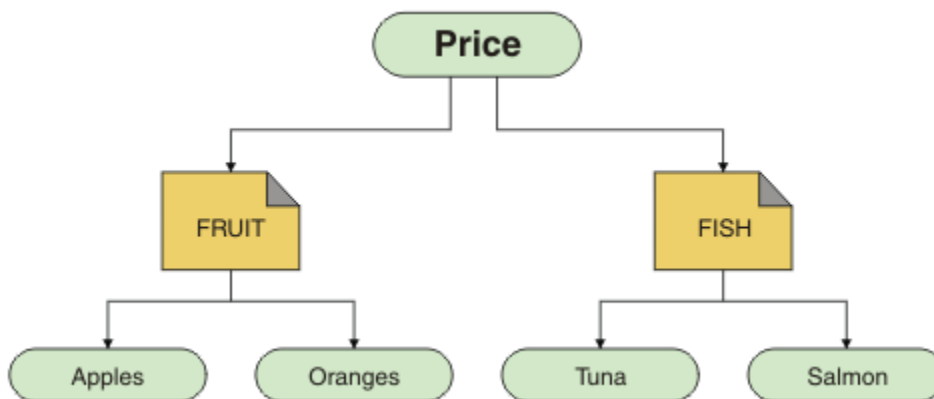
Im folgenden Diagramm gibt es zwei mögliche Multicastdatenströme:

```
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 des COMMINFO-Objekts MC2 definiert und der Price -Knoten hat keine Multicastdefinitionen.

In IBM MQ Multicast sind Themenzeichenfolgen auf 255 Zeichen beschränkt. Diese Einschränkung bedeutet, dass bei den Namen von Knoten und Blattknoten in der Baumstruktur Vorsicht geboten ist. Wenn die Namen von Knoten und Blattknoten zu lang sind, ist die Themenzeichenfolge möglicherweise länger als 255 Zeichen und gibt den Ursachencode 2425 (0979) (RC2425): MQRC_TOPIC_STRING_ERROR 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 Standard-MQMD-Werte sind nicht unbedingt identisch mit dem Wert für MQMD_DEFAULT und werden in [Tabelle 19 auf Seite 387](#) 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

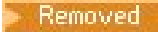
Es werden nur die Benutzereigenschaften übertragen.

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.

 XMS .NETMulticast-Messaging (mithilfeRMM) wurde veraltet abIBM MQ 9.2 und entfernt beiIBM MQ 9.3 .

Multicastnachrichten-Attribute

Nachrichtenattribute können aus verschiedenen Bereichen stammen, wie z. B. MQMD, die Felder in den MQRFH2- und Nachrichteneigenschaften.

Die folgende Tabelle zeigt, was passiert, wenn Nachrichten abhängig vom Wert von MCPROP (zuvor in diesem Abschnitt beschrieben) gesendet werden, und der Standardwert verwendet wird, wenn ein Attribut nicht gesendet wird.

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

Tabelle 19. Messaging-Attribute und ihre Beziehungen zu Multicasting (Forts.)

Attribut	Aktion bei Verwendung von Multicasting	Standardwert, wenn nicht übertragen
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
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](#)
[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 Multicast-Objekts **COMMINFO** verwendet werden, um die Codierung der Nachrichtenübertragung zu definieren, um die Effizienz zu erhöhen.

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 im Abschnitt [Zeilengruppe ClientExit](#) 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.

Mit dem MQSC-Befehl **DISPLAY CONN** können Sie Verbindungsinformationen zu den Anwendungen anzeigen, die mit dem Warteschlangenmanager verbunden sind. Weitere Informationen zum Befehl **DISPLAY CONN** finden Sie unter [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 Einreihung 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 Veröffentlichungen überträgt, die seit dem Zeitpunkt der Subskription erstellt wurden. **NONE** ist der Standardwert.

ALLE

Der Wert ALL bewirkt, dass der Sender den gesamten bekannten Verlauf eines Themas erneut überträgt. 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 Verweise

[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.

Eine zusätzliche IBM MQ-Multicast-Konfiguration kann in einer `ini`-Datei vorgenommen werden. Die Datei `ini`, die Sie verwenden müssen, hängt vom Typ der Anwendungen ab:

- Client: Konfigurieren Sie die Datei `MQ_DATA_PATH/mqclient.ini`.
- Warteschlangenmanager: Konfigurieren Sie die Datei `MQ_DATA_PATH/qmgrs/QMNAME/qm.ini`.

Dabei ist `MQ_DATA_PATH` die Position des IBM MQ-Datenverzeichnisses (`/var/mqm/mqclient.ini`) und `QMNAME` der Name des Warteschlangenmanagers, für den die Datei `.ini` 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.

FeedbackMode

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 erheblich kleiner als der Wert für **HeartbeatTimeout** sein, um falsche Zeitlimitereignisse für Überwachungssignale 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:

IBM MQEigenschaft	IBM MQ LLM-Eigenschaftstyp	LLM-Eigenschaftentyp	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

Tabelle 20. IBM MQ-Nachrichteneigenschaften für Eigenschaftszuordnung von IBM MQ LLM (Forts.)

IBM MQEigenschaft	IBM MQ LLM-Eigenschaftstyp	LLM-Eigenschaftenart	LLM-Eigen-schaftscode
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

CL-Befehle sind die bevorzugte Methode für die Verwaltung von IBM MQ auf IBM i. Sie können auch MQSC-Befehle, PCF-Befehle, Steuerbefehle und Fernverwaltung verwenden.

Informationen zu diesem Vorgang

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.

Zugehörige Konzepte

[Hinweise zu Namen von Warteschlangenmanagerbibliotheken von IBM MQ for IBM i](#)
[Installierbare Services und Komponenten unter IBM i](#)

Zugehörige Tasks

[IBM MQ -Konfigurationsinformationen auf Multiplatforms ändern](#)
[Sicherheit unter IBM i einrichten](#)

[„Steuerroutine der Warteschlange für nicht zustellbare Nachrichten unter IBM i aufrufen“ auf Seite 174](#)
 Unter IBM MQ for IBM i rufen Sie den DLQ-Handler mit dem Befehl **STRMQMDLQ** auf.

[Problembestimmung bei IBM MQ for IBM i-Anwendungen](#)

Zugehörige Verweise

[System-und Standardobjekte](#)

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.

Mit dem 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 „[IBM MQ mit MQSC-Befehlen verwalten](#)“ auf Seite 12 und „[IBM MQ Programmable Command Formats verwenden](#)“ auf Seite 28.

Zugehörige Verweise

[IBM MQ for IBM i CL-Befehlsreferenz](#)

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 468.

IBM i IBM MQ for IBM i-Objekte erstellen

Verwenden Sie diese Informationen, um die Methoden zum Erstellen von IBM MQ -Objekten für IBM i zu verstehen.

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 **CRTMQM**. Anweisungen zum Erstellen der Warteschlange finden Sie im Abschnitt „Erstellen einer lokalen Warteschlange mit dem Befehl `CRTMQM`“ auf Seite 396. 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 **WRKMQM-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 `CRTMQM 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)
MAXMSGL(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 399.

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 **CRTMQM** 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 (Steueroutine 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 unter [„Steueroutine der Warteschlange für nicht zustellbare Nachrichten unter IBM i aufrufen“](#) auf Seite 174.

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. For example:

```
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. For example:

```
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 397 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 **CHGMQMQ** :

```
CHGMQMQ QNAME('orange.local.queue') MQMNAME(MYQUEUEMANAGER) MAXMSGLEN(10000)
```

Mit diesem Befehl wird ein einzelnes Attribut geändert, das der maximalen Nachrichtenlänge entspricht. Alle anderen Attribute bleiben unverändert.

- Verwendung des Befehls **CRTMQMQ** mit der Option `REPLACE *YES`, z. B.:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL) MQMNAME(MYQUEUEMANAGER)  
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(MYQUEUEMANAGER)
```

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.

Die Produktinformation zu IBM i finden Sie unter [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.

IBM i

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 stellt und indirekt, wenn sie eine Nachricht in diese Warteschlange einreicht. 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 an 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:


```
CRTMQM QNAME('my.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
MQMNAME(MYQUEUEMANAGER)
```

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. For example:

```
CHGMQM QNAME('my.alias.queue') TGTQNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

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 */
CRTMQM QNAME('alphas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*YES) GETENBL(*NO) MQMNAME(MYQUEUEMANAGER)

/* This alias is put disabled and get enabled for application BETA */
CRTMQM QNAME('betas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*NO) GETENBL(*YES) MQMNAME(MYQUEUEMANAGER)
```

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. For example:

```
* Display the alias queue's attributes */
DSPMQM QNAME('alphas.alias.queue') MQMNAME(MYQUEUEMANAGER)

/* ALTER the base queue name, to which the alias resolves. */
/* FORCE = Force the change even if the queue is open. */
CHQMOM QNAME('alphas.alias.queue') TGTQNAME('orange.local.queue') FORCE(*YES)
MQMNAME(MYQUEUEMANAGER)
```

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 **WRKMQMQ-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). For example:

```
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 wurden, permanente dynamische Warteschlangen. Die Attribute, die nicht angegeben sind, werden automatisch aus der Standardwarteschlange `SYSTEM.DEFAULT.MODEL.QUEUE` kopiert.

Sie können das Attribut `REPLACE *YES` bei der Definition von Modellwarteschlangen auf dieselbe Weise wie bei lokalen Warteschlangen verwenden.

Andere Befehle mit Modellwarteschlangen verwenden

Sie können die entsprechenden Befehle verwenden, um die Attribute einer Modellwarteschlange anzuzeigen oder zu ändern. For example:

```
/* Display the model queue's attributes */
DSPMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('green.model.queue')

/* ALTER the model queue to enable puts on any */
/* dynamic queue created from this model. */
CHGMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('blue.model.queue') PUTENBL(*YES)
```

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 als Befehl **STRMQMTRM** bereitgestellte Auslösemonitor 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:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(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. Das Triggering 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)

Ein Auslöserereignis wird generiert, wenn die Anzahl der Nachrichten mit der erforderlichen Priorität (**TRGMSGPTY**) erreicht die in **TRGDEPTH** angegebene Anzahl.

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. Initialisierungswarteschlangen 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

Mit dem Befehl **CRTMQMPC** können Sie eine Prozessdefinition 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 **DSPMQMPCRC** 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

Mit dem Befehl **DSPMQMPCRC** können Sie die Ergebnisse Ihrer Definition untersuchen. For example:

```
MQMNAME(MYQUEUEMANAGER) DSPMQMPCRC('motor.insurance.quote.process')
```

Sie können auch den Befehl **CHGMQMPCRC** zum Ändern einer vorhandenen Prozessdefinition und den Befehl **DLTMQMPCRC** zum Löschen einer Prozessdefinition verwenden.

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 QMGR1 auf.

```
CRTMQM  MQMNAME(QMGR1) TEXT('System A - Queue +
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

2. Starten Sie diesen Warteschlangenmanager.

```
STRMQM  MQMNAME(QMGR1)
```

3. Definieren Sie die IBM MQ -Objekte unter SYSTEMA , die Sie zum Senden von Nachrichten an einen Warteschlangenmanager unter SYSTEMB benötigen.

```
/* Transmission queue */
CRTMQMQ QNAME(XMITQ.TO.QMGRB1) QTYPE(*LCL) +
MQMNAME(QMGR1) 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(QMGR1) 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)

```

- 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)

```

- Starten Sie den WS-Manager unter SYSTEMB.

```

STRMQM MQMNAME(QMGRB1)

```

- Definieren Sie die IBM MQ -Objekte, die Sie für den Empfang von Nachrichten vom Warteschlangenmanager unter 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')

```

- 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 mit dem 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:

```

WRKMQMMSG QNAME(TARGETB) MQMNAME(QMGRB1).

```

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, 2024. 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 */+
TGTQNAME('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 */

CRTMQMPCRC 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 verwenden, einschließlich MQSC-Befehle, PCF-Befehle, Steuerbefehle und Fernverwaltung.

Informationen zu diesem Vorgang

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 394.

Sie können auch MQSC-Befehle und PCF-Befehle verwenden, wie in den Unterabschnitten beschrieben, und Sie können Steuerbefehle verwenden, wie in [„Verwaltung von IBM MQ for Multiplatforms über Steuerbefehle“](#) auf Seite 10 beschrieben.

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.

Verwenden Sie eine der in den folgenden Unterabschnitten beschriebenen Verwaltungsmethoden als Alternative zur Verwendung von IBM i-CL-Befehlen:

IBM i Lokale und ferne Verwaltung unter IBM i

Sie verwalten IBM MQ for IBM i-Objekte lokal oder über Fernzugriff.

Informationen zu diesem Vorgang

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.

den 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 werden in den folgenden Schritten beschrieben.

Prozedur

- Ein Programm mit PCF-Nachrichten schreiben. Weitere Informationen finden Sie unter „[Verwaltung mithilfe von PCF-Befehlen unter IBM i](#)“ auf Seite 411.
- Schreiben Sie ein Programm mit der MQAI, die PCF-Nachrichten sendet. Weitere Informationen finden Sie unter „[Verwendung von MQAI zur Vereinfachung der Verwendung von PCFs](#)“ auf Seite 39.
- Verwenden Sie den IBM MQ Explorer, der mit IBM MQ for Windows verfügbar ist, um eine grafische Benutzerschnittstelle zu verwenden und die richtigen PCF-Nachrichten zu generieren. Siehe „[IBM MQ Explorer mit IBM MQ for IBM i verwenden](#)“ auf Seite 412.
- Verwenden Sie **STRMQMQSC**, um Befehle indirekt an einen fernen Warteschlangenmanager zu senden. Siehe „[MQSC-Befehle unter IBM i verwalten](#)“ auf Seite 410.

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.

MQSC-Befehle unter IBM i verwalten

Unter IBM i erstellen Sie eine Liste mit Befehlen in einer Scriptdatei und führen die Datei anschließend mit dem Befehl **STRMQMQSC** aus. Sie verwenden MQSC-Befehle zum Verwalten von WS-Manager-Objekten, einschließlich des Warteschlangenmanagers selbst, Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services, Themen und Authentifizierungsinformationsobjekte.

Informationen zu diesem Vorgang

IBM MQ -Scriptbefehle (MQSC) werden in lesbarer Form in EBCDIC geschrieben. MQSC-Befehle werden mit dem CL-Befehl **STRMQMQSC** 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 STRMQMQSC, da die Verwendung der Bibliothek QTEMP begrenzt ist. Sie müssen eine andere Bibliothek als Eingabedatei für den Befehl verwenden.

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

Die in MQSC angegebenen Objektattribute 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 allen IBM MQ -Plattformen verfügbar.

Eine Beschreibung der einzelnen MQSC-Befehle und ihrer Syntax finden Sie in [MQSC-Befehle](#).

Vorgehensweise

1. Erstellen Sie die QMQSC-Quellendatei.

IBM MQ for IBM i liefert 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')
```

2. Mit den Teildateien arbeiten.

Die MQSC-Quelle wird in Teildateien in dieser Quellendatei gehalten. Geben Sie den folgenden Befehl ein, um mit den Membern zu arbeiten:

```
WRKMBRPDM MYLIB/QMQSC
```

Sie können jetzt neue Mitglieder hinzufügen und vorhandene verwalten.

```
.
.
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 20. Auszug aus einer MQSC-Befehlsdatei, *myprog.in*, mit einem MQSC-Befehl (DEFINE QLOCAL) mit seinen Attributen.

Zugehörige Informationen

[IBM MQ mit MQSC-Befehlen verwalten](#)

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 39.

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 mit dem Namen IBM MQ Explorer bereit, um Verwaltungstasks alternativ zur Verwendung von CL-, Steuerungs- oder MQSC-Befehlen auszuführen.

The IBM MQ Explorer allows you to perform local or remote administration of your network from a computer running Windows (x86 platform), or Linux (x86 and x86-64 platforms), by pointing the IBM MQ Explorer at the queue managers and clusters you are interested in

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 **STRMQMSVR** gestartet wurde.
2. Für jeden fernen WS-Manager ist ein geeigneter TCP/IP-Listener vorhanden. Dies ist der IBM MQ-Listener, der mit dem 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.

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:

```
STRMQMSVR 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.

```
ENDMQMSVR 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.

```
ENDMQMSVR 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:

```
DSPMQMSVR MQMNAME('saturn.queue.manager')
```

Geben Sie diesen Befehl auf der Zielmaschine aus. Wenn der Befehlsserver aktiv ist, wird die in [Abbildung 21 auf Seite 414](#) gezeigte Anzeige aufgerufen:

```
Display MQM Command Server (DSPMQMSVR)
```

```
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 21. Anzeige 'MQM-Befehlsserver anzeigen'

Befehle der Webkonsole ausführen

Sie müssen Ihre Umgebung gemäß nachfolgender Beschreibung konfigurieren, damit die Qshell-Befehle mit Bezug auf die Webkonsole ordnungsgemäß unter IBM MQ for IBM i ausgeführt werden.

Informationen zu diesem Vorgang

Wenn die Qshell startet, initialisiert sie interne Tabellen für die Verarbeitung von Befehlen, die auf der CCSID des Jobs basieren. Damit die Qshell-Befehle mit Bezug auf die Webkonsole ordnungsgemäß unter IBM i ausgeführt werden, müssen Sie Ihre Umgebung konfigurieren.

Eine Ländereinstellung wird festgelegt, indem die Umgebungsvariable LANG auf den Pfadnamen für ein Objekt für die Ländereinstellung gesetzt wird. Wenn Sie beispielsweise die Ländereinstellung für amerikanisches Englisch festlegen möchten, wird die Umgebungsvariable LANG wie folgt festgelegt:

```
LANG=/QSYS.LIB/EN_US.LOCALE
```

In der Qshell können Sie die Einstellung nach ausgegebenem Befehlssatz überprüfen, um alle Umgebungsvariablen aufzulisten. Normalerweise handelt es sich um LANG, was Auswirkungen auf die Ländereinstellung für die Laufzeitumgebung haben kann. Es kann sich auch um LC_ALL handeln.

Damit Qshell-Befehle ordnungsgemäß ausgeführt werden, muss die Einstellung für die länderspezifische Umgebung mit Ihrer Jobeinstellung übereinstimmen.

Vorgehensweise

Verwenden Sie den CL-Befehl DSPJOB JOB(JobNumber/USERProfile/JobName)

- a) Wählen Sie Option 2 aus, um die Jobdefinitionsattribute anzuzeigen.

b) Die folgenden Attribute sollten mit der Umgebungseinstellung LANG oder LC_ALL übereinstimmen

- Sprachenkennung
- Landes- oder Regions-ID
- ID des codierten Zeichensatzes

Beispiel: Bei

```
LANG=/QSYS.LIB/FR_FR.LOCALE
```

sollten Ihre Jobattribute wie folgt lauten:

- Sprachen-ID FRA
- Land-oder Regions-ID FR
- ID des codierten Zeichensatzes. 297

Nächste Schritte

Weitere Informationen zur Unterstützung der Landessprache finden Sie beim IBM Documentation-Thema [Berücksichtigungen zur Unterstützung der Landessprache \(NLS\)](#).

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 21 auf Seite 416](#) 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 WRKMQMLSR anzeigen.

Jobname	Funktion
AMQZMUCO	Utility-Manager. Dieser Job führt kritische WS-Manager-Dienstprogramme aus, z. B. den Journalkettenmanager.
AMQZXMAO	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).
AMQZLAAO	WS-Manager-Agenten, die die meisten Arbeiten für Anwendungen ausführen, die mit MQCNO_STANDARD_BINDING eine Verbindung zum Warteschlangenmanager herstellen.
AMQZLSAO	Warteschlangenmanageragent.
AMQZMUFO	Utility-Manager
AMQZMGRO	Prozesscontroller. Dieser Job wird verwendet, um Empfangsprogramme und Services zu starten und zu verwalten.
AMQZMURO	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 Kanalantwörterin.
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).

IBM i 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 QMQMJOBID ausgeführt.

Die Arbeitsverwaltungsobjekte, die bei der Installation von IBM MQ bereitgestellt werden, sind in Tabelle 22 auf Seite 417 aufgelistet und die Objekte, die für einen Warteschlangenmanager erstellt wurden, sind in Tabelle 23 auf Seite 417 aufgelistet.

Anmerkung: Die Arbeitsverwaltungsobjekte können in der Bibliothek QMQM gefunden werden, und die WS-Manager-Objekte befinden sich in der WS-Manager-Bibliothek.

Name	Typ	Beschreibung
AMQZLAA0	*JOBID	Jobbeschreibung, die von den IBM MQ-Agentenprozessen verwendet wird
AMQZLSA0	*JOBID	Warteschlangenmanageragent für isolierte Bindungen
AMQZXMA0	*JOBID	Jobbeschreibung, die von IBM MQ-Ausführungscontrollern verwendet wird
QMQM	*SBSD	Subsystem, in dem alle IBM MQ-Jobs ausgeführt werden.
QMQM	*JOBQ	Die Jobwarteschlange, die dem angegebenen Subsystem zugeordnet ist.
QMQMJOBID	*JOBID	Die IBM MQ-Standardjobbeschreibung, die verwendet wird, wenn für einen Job keine bestimmte Jobbeschreibung angegeben ist.
QMQMMSG	*MSGQ	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

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.

Tabelle 23. Für einen Warteschlangenmanager erstellte Arbeitsverwaltungsobjekte (Forts.)		
Name	Typ	Beschreibung
AMQAJRNMSG	*MSGQ	Nachrichtenwarteschlange des lokalen Journals
AMQCRC6B	*PGM	Programm zum Starten der LU6.2-Verbindung
AMQRFOLD	*DATEI	Migrierte WS-Manager-Kanaldefinitionsdatei
QMQMMSG	*MSGQ	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. For example:

- 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 WS-Manager-Bibliothek finden Sie im Artikel [Bibliotheksnamen von IBM MQ for IBM i -Warteschlangenmanagern verstehen](#).
- 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 (QMQMRUN20, QMQMRUN35 oder QMQMRUN50) 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 21 auf Seite 416](#) 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 QMQMJOB0 exist in the queue manager library? Yes   No
Does job description AMQZXMA0 exist in the QMQM library?           Yes   No
Does job description QMQMJOB0 exist in the QMQM library?           Yes   No

```

Wenn Sie Nein zu all diesen Fragen beantworten, erstellen Sie eine globale Jobbeschreibung QMQMJOB0 in der QMQM-Bibliothek.

IBM MQ-Nachrichtenwarteschlange

Es wird in jeder Warteschlangenmanagerbibliothek eine IBM MQ-Nachrichtenwarteschlange mit dem Namen QMQMMSG 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 TESTQMAus.
2. IBM MQ durchsucht die Warteschlangenmanagerbibliothek QMTESTQM zunächst nach der Jobbeschreibung AMQZXMA0 und dann nach der Jobbeschreibung QMQMJOB0.

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.

IBM i Ablaufsteuerungsbeispiele unter IBM i konfigurieren

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. For example:

```
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 Leitwegfolgenummer angeben, aber die Werte müssen in einer sequenziellen Reihenfolge angegeben werden. Diese Folgenummer 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('QMQMRUN10')
```

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 QMQMJOBDB in der WS-Manager-Bibliothek, indem Sie den folgenden Befehl ausgeben:

```
CRTDUPOBJ OBJ(QMQMJOBDB) FROMLIB(QMQM) OBJTYPE(*JOBDB) TOLIB(QMlibraryname)
```

- f. Ändern Sie die Jobbeschreibung so, dass sie die neue Jobwarteschlange verwendet, indem Sie den folgenden Befehl ausgeben:

```
CHGJOBDB JOBDB(QMlibraryname/QMQMJOBDB) JOBQ(QMQM/QMQM2)
```

- g. Starten Sie das Subsystem, indem Sie den folgenden Befehl ausgeben:

```
STRSBS SBSDB(QMQM/QMQM2)
```

Anmerkung:

- 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.
- Alle WS-Manager-Jobs, die dem Warteschlangenmanager QMlibraryname zugeordnet sind, werden jetzt unter dem Subsystem QMQM2 ausgeführt.

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 Warteschlangenmanagersausfalls die Wiederherstellung zu beschleunigen.

In diesem Abschnitt wird beschrieben, wie IBM MQ für 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`, wobei `QM GRLIB` der Name der WS-Manager-Bibliothek und `A` ein Buchstabe ist, `A` im Fall eines Einzel-Instanz-Warteschlangenmanagers, 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 Warteschlangenmanagerbibliothek kann bei der Erstellung eines Warteschlangenmanagers mit dem Befehl **CRTMQM** angegeben werden.

Journale 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 WS-Manager mit **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 Folgendes:

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. `AMQA1nnnnn` wechselt zu `AMQA1(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 428).
 - b. Journaländerungen im letzten Journalempfänger aufzeichnen (**RCDMQMIMG**) und wiederholen Sie dann den vorherigen Schritt. Auf diese Weise können die alten Journalempfängeramen wieder verwendet 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 mqs.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 Details zur Konfiguration des Systems finden Sie unter [Ändern der IBM MQ -Konfigurationsdaten auf Multiplatforms](#) .

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 Folgendes:

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, sehen Sie sich das Schlüsselwort THRESHOLD im Befehl **CRTJRNRCV** an.

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 428.

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) DSPJRNTA(*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 22 auf Seite 425](#) zu sehen.

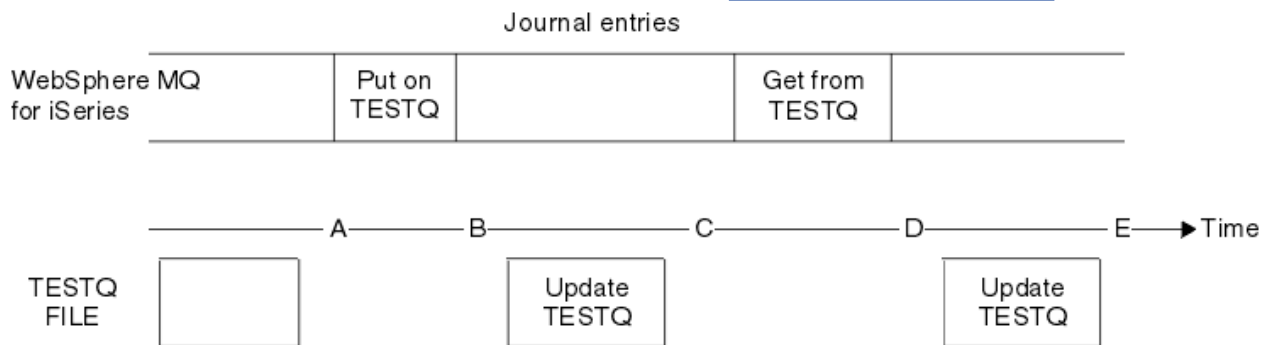


Abbildung 22. Reihenfolge von Ereignissen beim Aktualisieren von MQM-Objekten

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 422.

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 Fehlermeldung AMQ7462 in [„Journalverwaltung unter IBM i“](#) auf Seite 428)

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.

Anstatt das Datum des ältesten Datenträgerwiederherstellungseintrags für einen unnötig langen Zeitraum fortzusetzen, verwenden Sie den IBM MQ -Befehl RCDMQMIMG, mit dem Sie ein Image ausgewählter Objekte manuell 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.

Andere Objekte müssen mit dem IBM MQ-Befehl RCRMQMOBJ manuell wiederhergestellt werden. 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 Prüfpunkt-Thread im Prozess AMQZMUC0 ist für die Erfassung des Prüfpunkts an den folgenden Punkten zuständig:

- 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 unter [Changing IBM MQ configuration information on Multiplatforms](#).

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. Verwenden Sie den Befehl **RCDMQMIMG**, um ein MQM-Image für alle IBM MQ -Objekte aufzuzeichnen, und erzwingen Sie anschließend einen Prüfpunkt mit dem folgenden Befehl:

```
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 Ihr Warteschlangenmanager 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'))
```

Journalsicherung 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. Verwenden Sie den Befehl **RCDMQMIMG**, um ein MQM-Image für alle IBM MQ -Objekte aufzuzeichnen, und erzwingen Sie anschließend einen Prüfpunkt mit dem folgenden Befehl:

```
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 Journalsicherung durchzuführen. Dies hängt natürlich davon ab, wie Sie Ihre Sicherungsstrategie für Ihr Unternehmen eingerichtet haben.

IBM i **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 (RCRMQOBJ)

Bevor Sie einen Journalempfänger löschen, müssen Sie darauf achten, dass Sie eine Sicherungskopie haben und dass der Journalempfänger nicht mehr benötigt wird.

Journalempfänger 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 23 auf Seite 429](#) dargestellt.

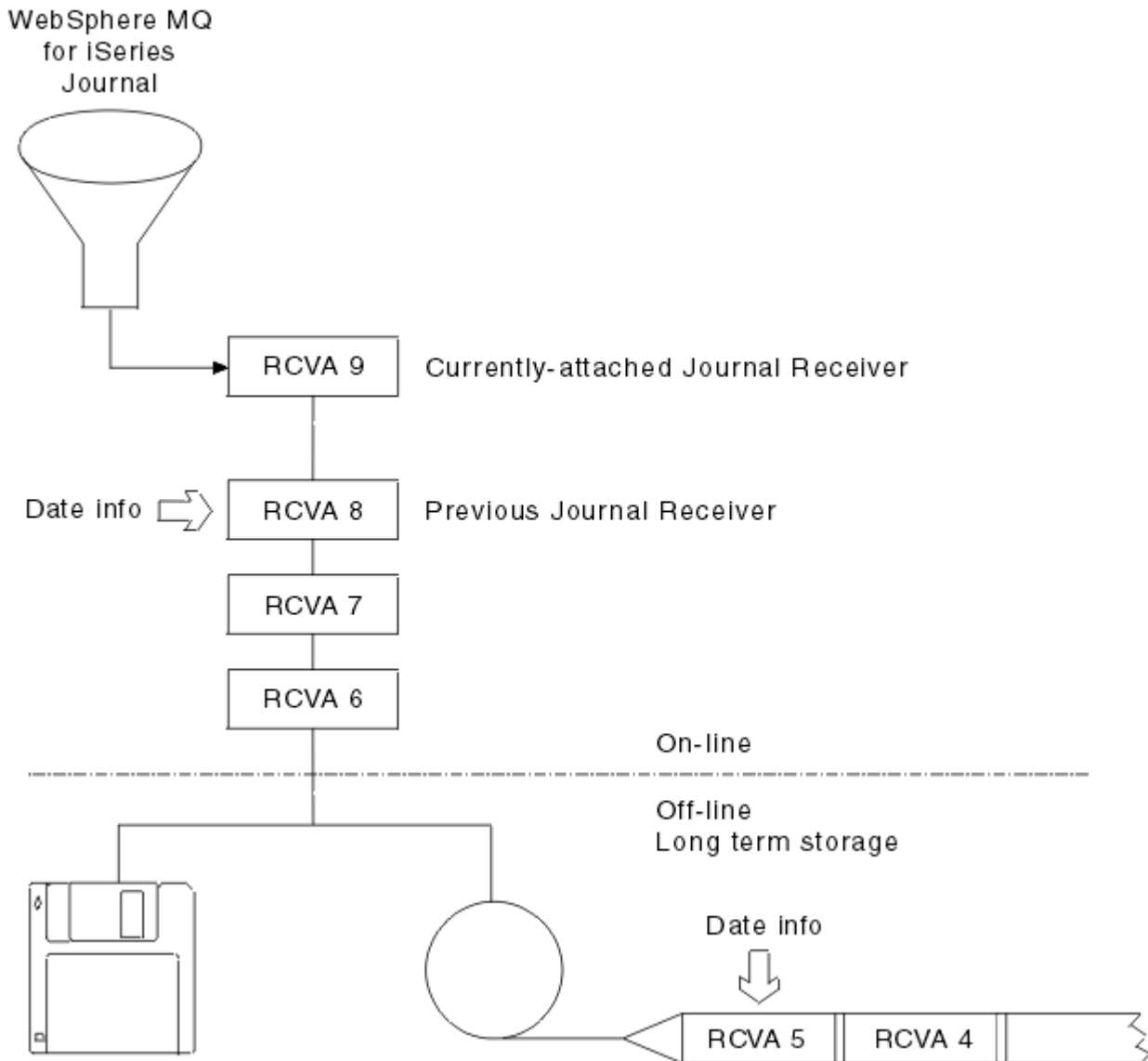


Abbildung 23. Journalführung in IBM i

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 (QMOMMSG 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:

```
JJJJJJJJJLLLLLLLLLLLLLLLLLLLLLLLLYYYYMMDDHHMMSSmmm
```

Dabei gilt:

JJJJJJJJJ

Ist der älteste Empfängername, den IBM MQ noch benötigt.

LLLLLLLLLLLL

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 (QMCMMSG) 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.

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(QMORLIB1) .....  
RSTLIB LIB(QMORLIB2) .....
```

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(*JRRCV) .....
```

Eine Reihe von Journalempfängern muss möglicherweise zurückgeschrieben werden, und nicht nur ein einzelner Empfänger. Beispiel: AMQA000007 ist der älteste Empfänger in den IBM MQ-Bibliotheken und sowohl AMQA000005 als auch AMQA000006 müssen wiederhergestellt 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 423):

```
CRTJRNRCV JRNRCV(QMQRLIB/AMQA900001) THRESHOLD(XXXXX)
```

2. Erstellen Sie das Journal manuell mit dem Journalempfänger:

```
CRTJRN JRN(QMGRLIB/AMQAJRN) MNGRCV(*SYSTEM) +  
JRNRCV(QMGRLIB/AMQA9000001) MSGQ(QMGRLIB/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 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 konsistenten 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 (Journalspiegelung) 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 den IBM i-Informationen 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 mithilfe von MQCONNX und geben 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](#).

 *Gemeinsam 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 Netzfreigabe in den Host-WS-Manager-Instanzen gehen.

Vorbereitende Schritte

- Für diese Aufgabe sind drei IBM i-Server erforderlich. Die Netzwerkfreigabe 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 gemeinsame Nutzung.

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) Give the share a name, WMQ for example.

- e) Setzen Sie die Zugriffssteuerung auf Read/Write.

- f) Wählen Sie den **Pfadnamen** aus, indem Sie in das zuvor erstellte /QIBM/UserData/mqm/qmgrs -Verzeichnis navigieren, und klicken Sie auf **OK**.

Das Fenster **IBM i NetServer-Dateifreigabe-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-Berechtigungen-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 /QIBM/UserData/mqm/qmgrsBerechtigungen erteilen möchten.

Sie können z. B. die Standardberechtigungen für Benutzerprofil (Public) Read und Execute für /QIBM/UserData/mqm/qmgrs angeben.

5. Stellen Sie sicher, dass alle Benutzerprofile, denen der Zugriff auf /QIBM/UserData/mqm/qmgrs auf GAMMA gewährt wird, dasselbe Kennwort haben wie auf den Servern, 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. Verwenden Sie alternativ die Befehle **CHFUSRPRF** und **CHGPWD** .

Ergebnisse

Überprüfen Sie, ob Sie mit der Freigabe auf GAMMA von anderen Servern zugreifen können. Wenn Sie die anderen Tasks ausführen, überprüfen Sie, ob Sie auf GAMMA von ALPHA und BETA auf den Pfad /QNTC/GAMMA/WMQ 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 449 oder „Einzel-Instanz-Warteschlangenmanager mithilfe von NetServer und Journalspiegelung unter IBM i in einen Multi-Instanz-Warteschlangenmanager konvertieren“ auf Seite 453 einen Warteschlangenmanager mit mehreren Instanzen.

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-Warteschlangenma-

nager 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 439)
2. Einen unabhängigen Zusatzspeicherpool, in den der Warteschlangenmanager verschoben wird (siehe [„Unabhängige Zusatzspeicherpools \(IASPs\)“](#) auf Seite 439)
3. Eine Clusterressourcengruppe (siehe [„Einheiten-Clusterressourcengruppen“](#) auf Seite 440), in der Sie Folgendes definieren:
 - Wiederherstellungsdomäne
 - IASP
 - Exitprogramm (siehe [„Exitprogramm für Einheiten-CRG“](#) auf Seite 440)

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.
- Ordzeichnet 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.

Das Beispiexitprogramm AMQSCRG4 ist unter IBM i verfügbar.

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 442 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 441 enthält Beispielcode, der Sie bei der Ausführung dieser Operation unterstützt.

Zugehörige Konzepte

„Unabhängige ASPs und hohe Verfügbarkeit“ auf Seite 462

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-Übernahmearadresse 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] *DEV *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] *DEV *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([MANAGER 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]

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 24 auf Seite 445 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 24 auf Seite 445 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 423 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.

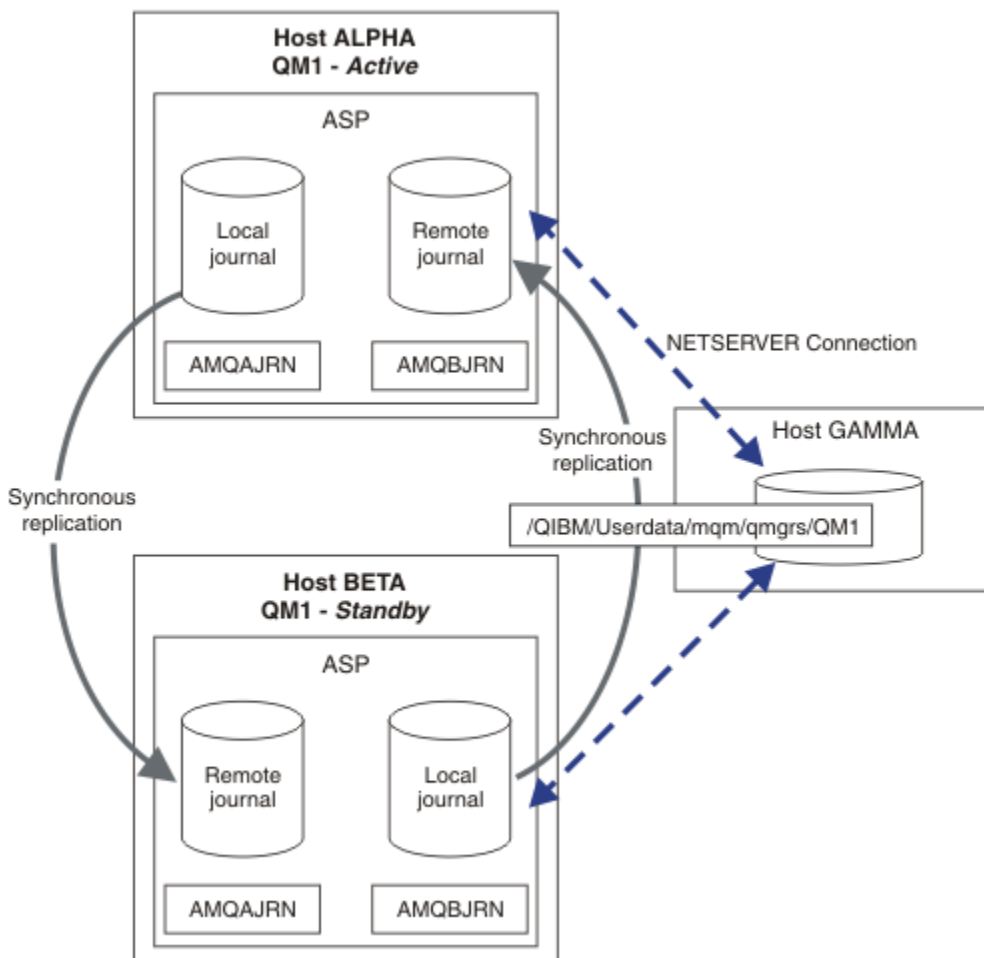


Abbildung 24. WS-Manager-Journal spiegeln

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 *ASync

IBM i-Journale werden entweder durch synchrone (*SYNC) oder asynchrone (*ASync) Journalführung gespiegelt (siehe [Verwaltung ferner Journale](#)).

Der Replikationsmodus in [Abbildung 24 auf Seite 445](#) ist *SYNC, nicht *ASync. *ASync 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 24 auf Seite 445](#) 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 *ASYNC ist ein Trade-off. In [Tabelle 24 auf Seite 446](#) sind die Unterschiede der Journalführung im *SYNC- und im *ASYNC-Modus zwischen zwei Warteschlangenmanagern aufgeführt.

Aktiv	Standby	*SYNC	*ASYNC
*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.
*ASYNC		<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/*ASYNC-Konfiguration zwischen einem Serverpaar hat den Nachteil einer Verzögerung beim Wiederaufnehmen der Standby-Instanz nach dem Failover. Für die Konfiguration *SYNC / *ASYNC ist dieses Problem nicht aufgetreten.

*SYNC / *ASYNC 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 / *ASYNC**

Die aktive WS-Manager-Instanz verwendet die *SYNC -Journalfunktion, und wenn die Standby-Warteschlangenmanagerinstanz gestartet wird, wird die *ASYNC -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 *ASYNC in *SYNC umschaltet, wird die aktive Instanz angehalten, wenn der Status des fernen Journals *ASYNCPEND 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 konsistent. Aus der Perspektive der Verwaltung von Warteschlangenmanagern mit mehreren Instanzen hat der IBM i -Systembediener in einer *SYNC / *ASYNC -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 *ASYNC 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 *SYNC gesetzt 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 *ASYNC oder *SYNC 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 **ADDQMJRN** 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. Das Produkt muss mindestens IBM WebSphere MQ 7.0.1 Fix Pack 1 aufweisen.
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 25](#) auf Seite 452 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 Warteschlangenmanager QM1 auf 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 **ADDQMJRN** 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 435 beschriebenen Schritt aus.

Als Ergebnis haben ALPHA und BETA einen Anteil, /QNTC/GAMMA/WMQ, der 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.

```
ADDQMJRN MQMNAME(QM1) RMTJRN RDB(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.

```
ADDQMQRN 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 450 auszuführen, können Sie auch mit dem folgenden Befehl die Journalführung von BETA auf ALPHA im asynchronen Modus ausführen:

```
ADDQMQRN MQMNAME (QM1) RMTJNRDB (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 443, 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:

```
WRKMQRN MQMNAME(QM1)
```

- b) Auf BETA:

```
WRKMQRN 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

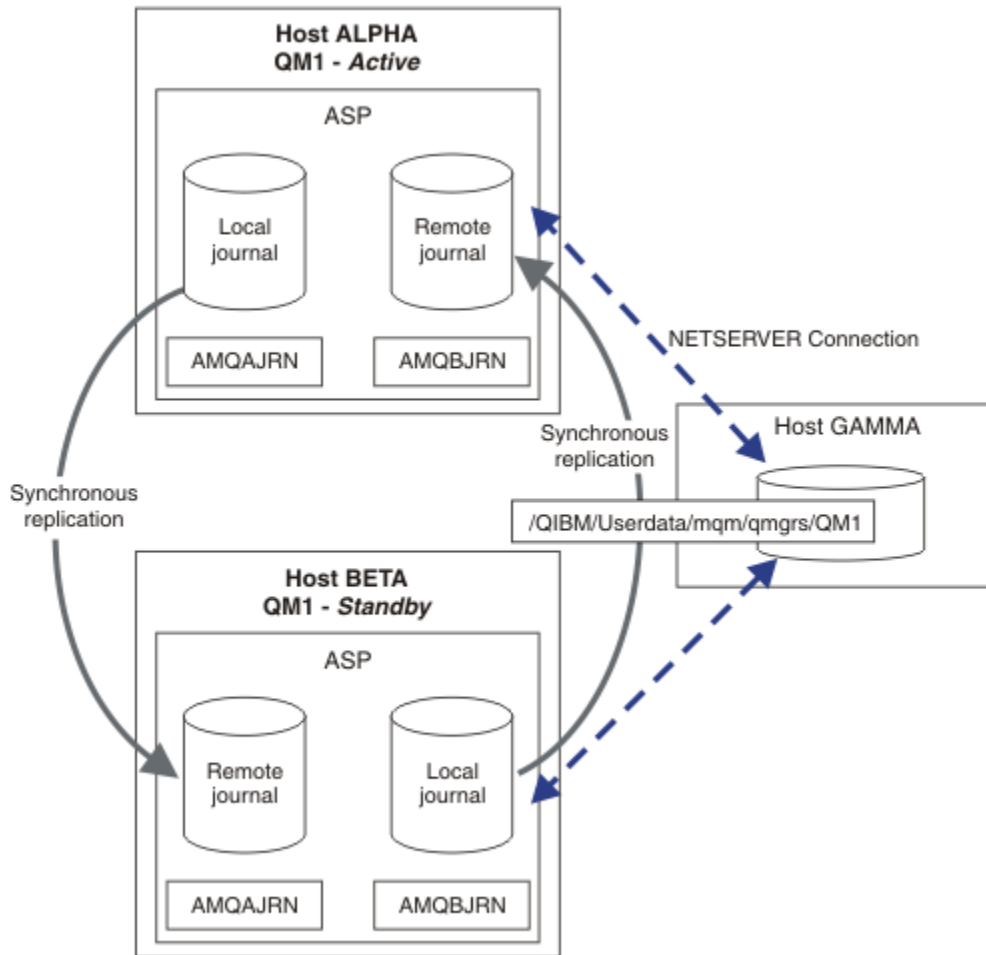


Abbildung 25. Gespiegelte Journalkonfiguration

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 Beispielpprogramme sind 'C'-Clients. Sie können sie von einer Windows- oder UNIX-Plattform aus ausführen.

1. Starten Sie die Beispielpprogramme 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 26 auf Seite 457](#) 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 das Verschieben der Warteschlangenmanagerverzeichnisse und -dateien in den gemeinsam genutzten Netzbereich mit dem Befehl **CPY**.

- 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 **ADDMQMJRN** vereinfacht das Hinzufügen der Journale und der Verbindungen.
- Starten Sie den Warteschlangenmanager, um eine Standby-Instanz zu ermöglichen.

Anmerkung:

In Schritt „4“ auf Seite 454 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

454 zunächst die WS-Manager-Bibliothek sichern. Im Anschluss an Schritt „5“ auf Seite 454 wird die WS-Manager-Bibliothek wiederhergestellt.

Anmerkung:

In Schritt „5“ auf Seite 454 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 435 beschriebenen Schritt aus.

Als Ergebnis haben ALPHA und BETA einen Anteil, /QNTC/GAMMA/WMQ, der 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.

```
STRMQMQSC 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.

```
ADDQMJRN 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.

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.

```
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.

11. Auf BETA ausführen. Der Befehl fügt ein lokales Journal auf BETA und ein fernes Journal auf ALPHA für hinzu.

```
ADDQMJRN 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 450 auszuführen, können Sie auch mit dem folgenden Befehl die Journalführung von BETA auf ALPHA im asynchronen Modus ausführen:

```
ADDQMJRN MQMNAME (QM1) RMTJNRDB (ALPHA) RMTJRN DLV (*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 443, 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

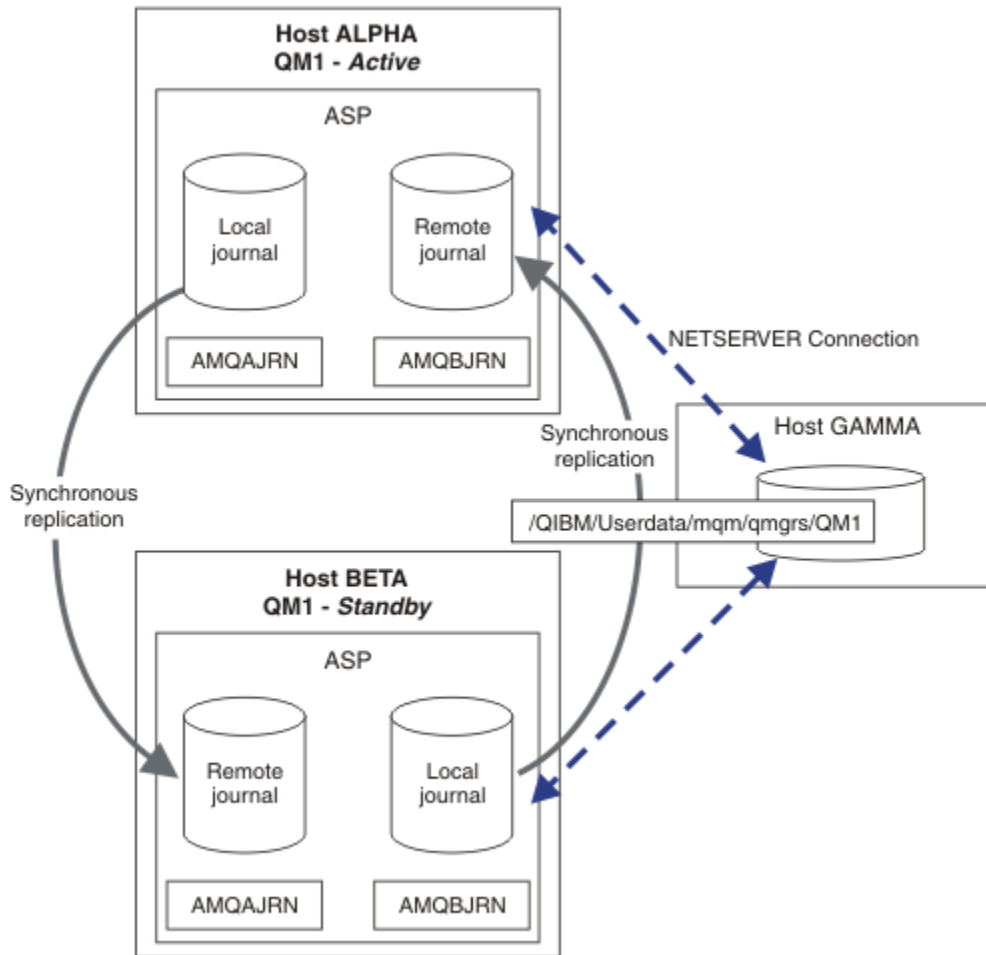


Abbildung 26. Gespiegelte Journalkonfiguration

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.

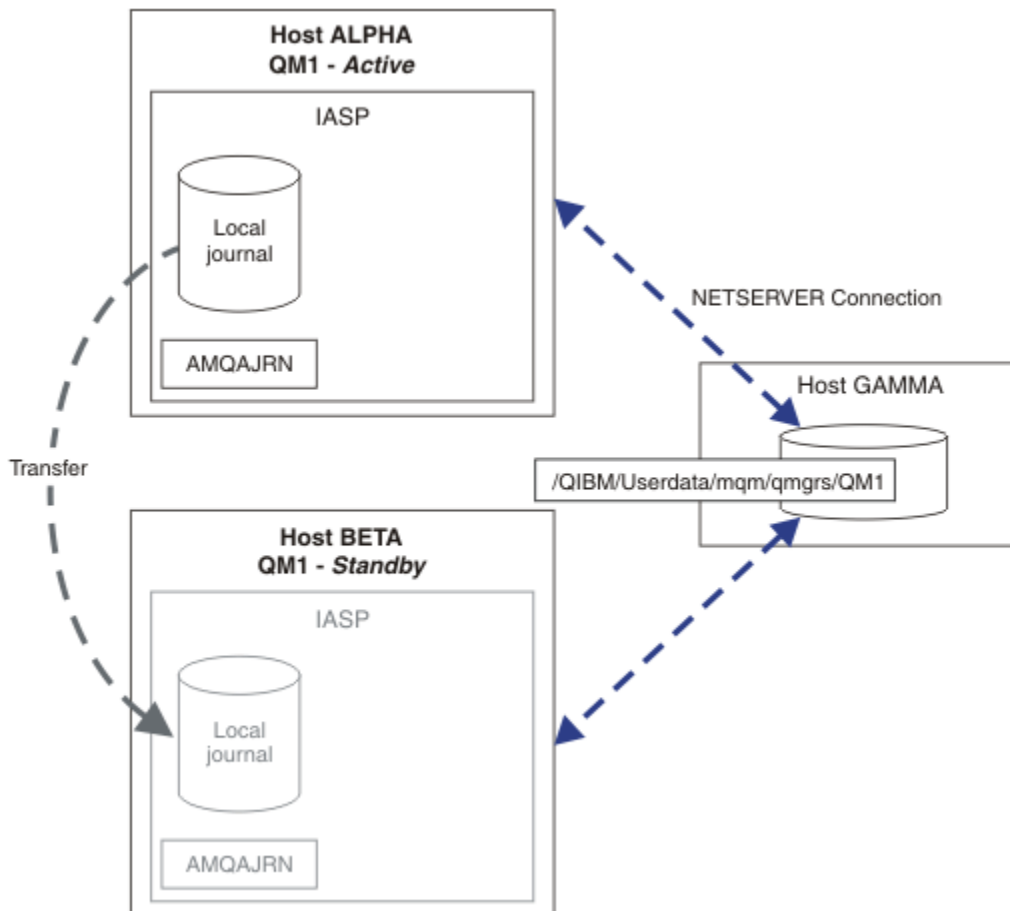


Abbildung 27. WS-Manager-Journal mit einem unabhängigen ASP übertragen

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 434).

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 462.

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.

Multi-Instanz-Warteschlangenmanager unter Verwendung von unabhängigem ASP 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 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. Das Produkt muss mindestens IBM WebSphere MQ 7.0.1 Fix Pack 1 aufweisen.
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 28](#) auf Seite 461 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 Warteschlangenmanager QM1 auf 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 435 beschriebenen Schritt aus.

Als Ergebnis haben ALPHA und BETA einen Anteil, /QNTC/GAMMA/WMQ, der 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) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIR(' /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=QMOM1
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(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.

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

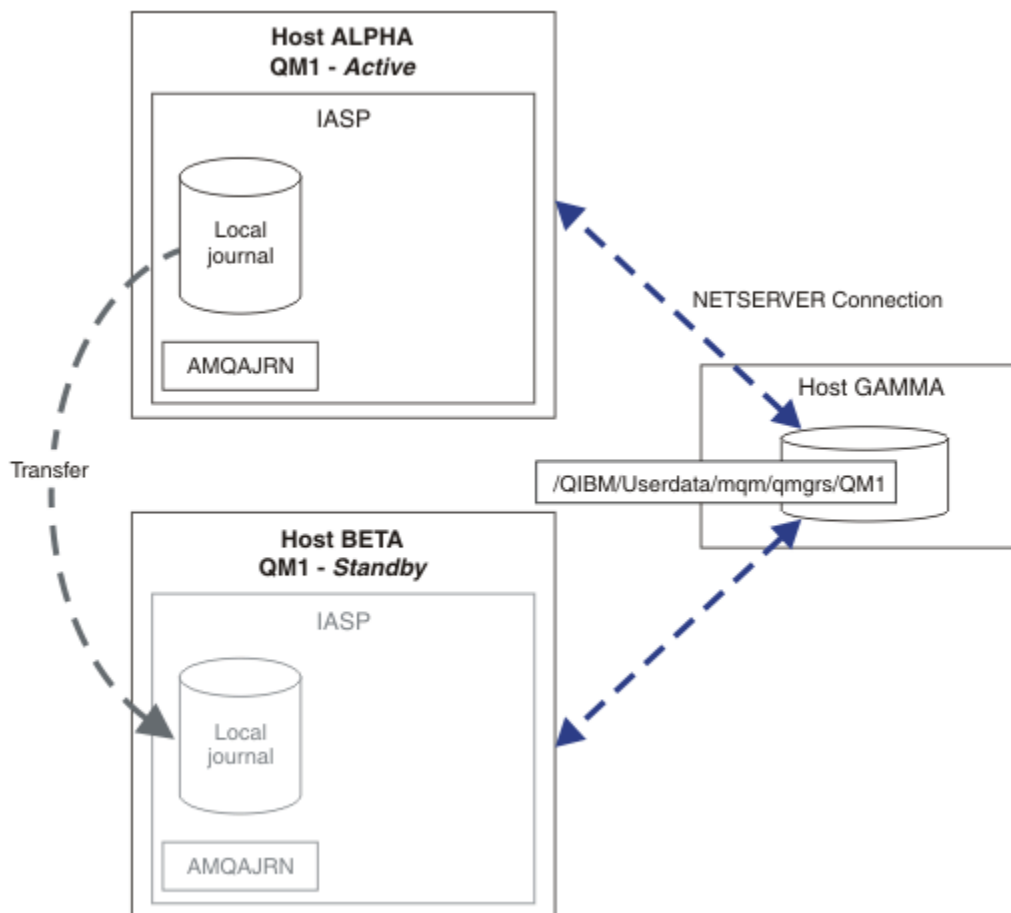


Abbildung 28. WS-Manager-Journal mit einem unabhängigen ASP übertragen

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.

Wählte 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 Replikatjournals 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. Siehe „Multi-Instanz-Warteschlangenmanager unter Verwendung von Journalspiegelung und NetServer unter IBM i erstellen“ auf Seite 449.
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 465 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. Siehe „Multi-Instanz-Warteschlangenmanager unter Verwendung von Journalspiegelung und NetServer unter IBM i erstellen“ auf Seite 449. 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')
```



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 Warteschlangenmanager-Instanzen mit den folgenden Befehlen hinzu und verwalten Sie sie, indem Sie die `mqs.ini`-Datei ändern.

ADDMQMINF

Der Befehl verwendet Informationen, die Sie aus der `mqs.ini`-Datei mit dem Befehl `DSPMQMINF` extrahiert haben, um auf einem anderen IBM i-Server eine neue Warteschlangenmanagerinstanz 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 bei der Konfiguration eines Multi-Instanz-Warteschlangenmanagers.

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](#) in der Dokumentation zu IBM i.

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 469) oder ob alle Warte-

schlangenmanager heruntergefahren werden sollen (siehe „[Alle Warteschlangenmanager für IBM MQ for IBM i herunterfahren](#)“ auf Seite 471).

5. Fahren Sie den mqweb-Server herunter, indem Sie den folgenden Befehl in qshell eingeben:

```
/QIBM/ProdData/mqm/bin/endmqweb
```

ENDMQM, Parameter ENDCCTJOB (*YES)

Die Funktionsweise des ENDMQM-Parameters ENDCCTJOB(*YES) in IBM MQ for IBM i Version 6.0 und höher unterscheidet sich von der in vorherigen Versionen.

In früheren Versionen, wenn Sie ENDCCTJOB (*YES) angeben, beendet MQ Ihre Anwendungen zwangsweise für Sie.

In IBM MQ for IBM i Version 6.0 oder höher hingegen werden die Anwendungen bei Angabe von ENDCCTJOB(*YES) nicht beendet, sondern ihre Verbindung zum Warteschlangenmanager wird 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 den Befehl ENDJOBABN nur dann für IBM MQ-Jobs, die nicht mit ENDJOB oder ENDSBS beendet werden konnten, wenn Sie die Maschine anschließend sofort über IPL starten.

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.

Administering IBM MQ for z/OS

IBM MQ for z/OS can be controlled and managed by MQSC and PCF commands, by a set of utilities and programs provided with the product, and by authorized applications.

For details of how to administer IBM MQ for z/OS and the different administrative tasks you might have to undertake, see the following links.

You can also administer IBM MQ for z/OS using the IBM MQ Explorer running in a Linux shell. For more information, see [“Verwaltung über die IBM MQ Explorer” on page 125](#).

Related concepts

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

Related tasks

[“IBM MQ verwalten” on page 7](#)

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.

[Planning your IBM MQ environment on z/OS](#)

[Configuring queue managers on z/OS](#)

Issuing queue manager commands on z/OS

You can control most of the operational environment of IBM MQ by using control commands. You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or authorized applications.

About this task

You use MQSC commands, in batch or interactive mode, to administer queue managers directly. You use PCF commands to help you create applications that administer queue managers. MQSC commands are in human-readable text form, whereas PCF commands let applications create requests and read the replies without having to parse text strings. Like MQSC commands, applications issue PCF commands by sending them as messages to the command input queue.

The following topics describe how you issue queue manager commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications.

Not all commands can be issued from all sources. See [“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS”](#) on page 473.

Related tasks

[Preparing sample applications for the TSO environment on z/OS](#)

Related information

[Administering IBM MQ using MQSC commands](#)

Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS

You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications. Not all commands can be issued from all these sources.

Which MQSC and PCF commands can control each IBM MQ object

Table 1 in "Command summary for IBM MQ for z/OS" maps which MQSC and PCF commands can be used on IBM MQ for z/OS to alter, define, delete and display each IBM MQ object. See also [MQSC commands reference](#) and [“IBM MQ Programmable Command Formats verwenden”](#) on page 28.

List of sources from which commands can be issued

If you are a suitably authorized user, you can issue IBM MQ commands from the following sources:

- The z/OS console or equivalent (such as SDSF/TSO).

See also [“Using the operations and control panels on z/OS”](#) on page 487.

Note: When using the z/OS console, you need to add /cpf to the start of a command, where cpf is the command prefix for the queue manager subsystem.

- The initialization input data sets CSQINP1, CSQINP2, CSQINPT and CSQINPX.

See [“Initialization commands for IBM MQ for z/OS”](#) on page 484.

- The z/OS master get command routine, MGCRE (SVC 34).
- The IBM MQ batch utility programs such as CSQUTIL, which processes a list of commands in a sequential data set.

See [“Using the IBM MQ for z/OS utilities”](#) on page 495.

- Suitably authorized applications, sending commands as messages to the SYSTEM.COMMAND.INPUT queue.

The application can be any of the following:

- A batch region program
- A CICS application
- An IMS application
- A TSO application
- An application program or utility on another IBM MQ system

See [“Writing programs to administer IBM MQ for z/OS” on page 503](#) and [Preparing sample applications for the TSO environment on z/OS](#).

Not all commands can be issued from all sources

Commands are classified according to where they can be issued from:

1

CSQINP1

2

CSQINP2

C

The z/OS console

R

The command server and command queue, by means of CSQUTIL, CSQINPT, CSQINPX, or authorized applications.

Within the command descriptions in MQSC commands reference, these sources are identified by the use of the characters 1, 2, C, and R in each command description. Table 2 in ["Command summary for IBM MQ for z/OS"](#) summarizes the MQSC commands and the sources from which they can be issued.

Related tasks

[Preparing sample applications for the TSO environment on z/OS](#)

Related information

[Administering IBM MQ using MQSC commands](#)

Command summary for IBM MQ for z/OS

A summary of the main MQSC and PCF commands, and of the sources from which you can run MQSC commands on IBM MQ for z/OS.

Table 25 on page 474 maps which MQSC and PCF commands can be used on IBM MQ for z/OS to alter, define, delete and display each IBM MQ object.

<i>Table 25. Summary of the main MQSC and PCF commands by object type</i>				
MQSC command	ALTER	DEFINE	DISPLAY	DELETE
PCF command	Change	Create/Copy	Inquire	Delete
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

Table 25. Summary of the main MQSC and PCF commands by object type (continued)

MQSC command	ALTER	DEFINE	DISPLAY	DELETE
QCLUSTER			M	
QLOCAL	M	M	M	M
QMGR	X		X	
QMODEL	M	M	M	M
QREMOTE	M	M	M	M
QUEUE	P	P	X	P
QSTATUS			X	
STGCLASS	X	X	X	X

Key to table symbols:

- M = MQSC only
- P = PCF only
- X = both

There are many other MQSC and PCF commands which allow you to manage other IBM MQ resources, and carry out other actions in addition to those summarized in [Table 25 on page 474](#).

[Table 26 on page 475](#) shows every MQSC command, and where each command can be issued from.

- CSQINP1 initialization input data set
- CSQINP2 initialization input data set
- z/OS console (or equivalent)
- SYSTEM.COMMAND.INPUT queue and command server (from applications, CSQUTIL, or the CSQINPX initialization input data set)

Table 26. Sources from which to run MQSC commands

Command	CSQINP1	CSQINP2	z/OS console	Command in-put queue and 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 PROCESS		X	X	X
ALTER PSID			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

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
ALTER SECURITY	X	X	X	X
ALTER SMDS		X	X	X
ALTER STGCLASS		X	X	X
ALTER SUB			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
CLEAR TOPICSTR			X	X
DEFINE AUTHINFO		X	X	X
DEFINE BUFFPOOL	X			
DEFINE CFSTRUCT		X	X	X
DEFINE CHANNEL		X	X	X
DEFINE LOG			X	X
DEFINE MAXSMGS		X	X	X
DEFINE NAMELIST		X	X	X
DEFINE PROCESS		X	X	X
DEFINE PSID	X		X	X
DEFINE QALIAS		X	X	X
DEFINE QLOCAL		X	X	X
DEFINE QMODEL		X	X	X
DEFINE QREMOTE		X	X	X
DEFINE STGCLASS		X	X	X
DEFINE SUB			X	X
DEFINE TOPIC		X	X	X
DELETE AUTHINFO		X	X	X
DELETE BUFFPOOL		X	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

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
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
DELETE TOPIC		X	X	X
DISPLAY ARCHIVE	X	X	X	X
DISPLAY AUTHINFO		X	X	X
DISPLAY CFSTATUS			X	X
DISPLAY CFSTRUCT		X	X	X
DISPLAY CHANNEL		X	X	X
DISPLAY CHINIT			X	X
DISPLAY CHLAUTH		X	X	X
DISPLAY CHSTATUS			X	X
DISPLAY CLUSQMGR			X	X
DISPLAY CMDSERV	X	X	X	X
DISPLAY CONN		X	X	X
DISPLAY GROUP		X	X	X
DISPLAY LOG	X	X	X	X
DISPLAY MAXSMSGS		X	X	X
DISPLAY NAMELIST		X	X	X
DISPLAY PROCESS		X	X	X
DISPLAY PUBSUB		X	X	X
DISPLAY QALIAS		X	X	X
DISPLAY QCLUSTER		X	X	X
DISPLAY QLOCAL		X	X	X
DISPLAY QMGR		X	X	X
DISPLAY QMODEL		X	X	X
DISPLAY QREMOTE		X	X	X
DISPLAY QSTATUS		X	X	X
DISPLAY QUEUE		X	X	X
DISPLAY SBSTATUS			X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
DISPLAY SECURITY			X	X
DISPLAY SMDS		X	X	X
DISPLAY SMDSCONN		X	X	X
DISPLAY STGCLASS		X	X	X
DISPLAY SUB			X	X
DISPLAY SYSTEM	X	X	X	X
DISPLAY TCLUSTER		X	X	X
DISPLAY THREAD		X	X	X
DISPLAY TOPIC		X	X	X
DISPLAY TPSTATUS		X	X	X
DISPLAY TRACE	X	X	X	X
DISPLAY USAGE		X	X	X
MOVE QLOCAL		X	X	X
PING CHANNEL			X	X
RECOVER BSDS			X	X
RECOVER CFSTRUCT			X	X
REFRESH CLUSTER			X	X
REFRESH QMGR		X	X	X
REFRESH SECURITY			X	X
RESET CFSTRUCT			X	X
RESET CHANNEL			X	X
RESET CLUSTER			X	X
RESET QMGR		X	X	X
RESET QSTATS		X	X	X
RESET SMDS			X	X
RESET TPIPE			X	X
RESOLVE CHANNEL			X	X
RESOLVE INDOUBT		X	X	X
RESUME QMGR			X	X
RVERIFY SECURITY		X	X	X
SET ARCHIVE	X	X	X	X
SET CHLAUTH		X	X	X
SET LOG	X	X	X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
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 SMDSCONN		X	X	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 SMDSCONN		X	X	X
STOP TRACE	X	X	X	X
SUSPEND QMGR			X	X

In [MQSC commands](#), each command description identifies the sources from which that command can be run.

Using MQSC to start and stop a queue manager on z/OS

An introduction to using control commands on IBM MQ for z/OS: After you have installed IBM MQ, use MQSC commands to start and stop a queue manager.

Before you begin

After you have installed IBM MQ, it is defined as a formal z/OS subsystem. This message appears during any initial program load (IPL) of z/OS:

```
CSQ3110I +CSQ1 CSQ3UR00 - SUBSYSTEM ssnm INITIALIZATION COMPLETE
```

where *ssnm* is the IBM MQ subsystem name.

From now on, you can start the queue manager for that subsystem *from any z/OS console that has been authorized to issue system control commands*; that is, a z/OS SYS command group. You must issue the START command from the authorized console, you cannot issue it through JES or TSO.

If you are using queue sharing groups, you must start RRS first, and then Db2®, before you start the queue manager.

About this task

When a queue manager stops under normal conditions, its last action is to take a termination checkpoint. This checkpoint, and the logs, give the queue manager the information it needs to restart.

The following steps contain information about the START and STOP commands, and contain a brief overview of start-up after an abnormal termination has occurred.

Procedure

1. Start a queue manager

You start a queue manager by issuing a START QMGR command. However, you cannot successfully use the START command unless you have appropriate authority. See the [Setting up security on z/OS](#) for information about IBM MQ security. The following code shows examples of the START command. Note that you must prefix an MQSC command with a command prefix string (CPF).

```
+CSQ1 START QMGR
+CSQ1 START QMGR PARM(NEWLOG)
```

See [START QMGR](#) for information about the syntax of the START QMGR command.

You cannot run the queue manager as a batch job or start it using a z/OS command START. These methods are likely to start an address space for IBM MQ that then ends abnormally. Nor can you start a queue manager from the CSQUTIL utility program or a similar user application.

You can, however, start a queue manager from an APF-authorized program by passing a START QMGR command to the z/OS MGCRC (SVC 34) service.

If you are using queue sharing groups, the associated Db2 systems and RRS must be active when you start the queue manager.

Start options

When you start a queue manager, a system parameter module is loaded. You can specify the name of the system parameter module in one of two ways:

- With the PARM parameter of the /cpf START QMGR command, for example

```
/cpf START QMGR PARM(CSQ1ZPRM)
```

- With a parameter in the startup procedure, for example, code the JCL EXEC statement as

```
//MQM EXEC PGM=CSQYASCP,PARM='ZPARM(CSQ1ZPRM)'
```

A system parameter module provides information specified when the queue manager was customized.

You can use the **QMGRPROD** option to specify the product against which the queue manager usage is to be recorded, and the **AMSPROD** option to specify the equivalent for AMS if that is used. See the [MQSC START QMGR](#) command for details of the permitted values.

An example JCL EXEC statement follows:

```
//MQM EXEC PGM=CSQYASCP,PARM='QMGRPROD(MQ)'
```

See [z/OS MVS Product Management](#) for more information on measured usage and product registration.

You can also use the ENVPARM option to substitute one or more parameters in the JCL procedure for the queue manager.

For example, you can update your queue manager startup procedure, so that the **DDname** CSQINP2 is a variable. This means that you can change the CSQINP2 **DDname** without changing the startup procedure. This is useful for implementing changes, providing back-outs for operators, and queue manager operations.

Suppose your start-up procedure for queue manager CSQ1 looked like this:

```
//CSQ1MSTR PROC INP2=NORM
//MQMESA EXEC PGM=CSQYASCP
//STEPLIB DD DISP=SHR,DSN=th1qual.SCSQANLE
// DD DISP=SHR,DSN=th1qual.SCSQAUTH
// DD DISP=SHR,DSN=db2qual.SDSNLOAD
//BDS1 DD DISP=SHR,DSN=myqual.BSDS01
//BDS2 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=*
```

If you then start your queue manager with the following command:

```
+CSQ1 START QMGR
```

then the CSQINP2 used is a member called CSQ1NORM.

However, suppose you are putting a new suite of programs into production so that the next time you start queue manager CSQ1, the CSQINP2 definitions are to be taken from member CSQ1NEW. To do this, you would start the queue manager with this command:

```
+CSQ1 START QMGR ENVPARM('INP2=NEW')
```

and CSQ1NEW would be used instead of CSQ1NORM. Note: z/OS limits the **KEYWORD=value** specifications for symbolic parameters (as in **INP2=NEW**) to 255 characters.

Starting after an abnormal termination

IBM MQ automatically detects whether restart follows a normal shutdown or an abnormal termination.

Starting a queue manager after it ends abnormally is different from starting it after the **STOP QMGR** command has been issued. After **STOP QMGR**, the system finishes its work in an orderly way and takes a termination checkpoint before stopping. When you restart the queue manager, it uses information from the system checkpoint and recovery log to determine the system status at shutdown.

However, if the queue manager ends abnormally, it terminates without being able to finish its work or take a termination checkpoint. When you restart a queue manager after an **abend**, it refreshes its knowledge of its status at termination using information in the log, and notifies you of the status of various tasks. Normally, the restart process resolves all inconsistent states. But, in some cases, you must take specific steps to resolve inconsistencies.

User messages on start-up

When you start a queue manager successfully, the queue manager produces a set of startup messages.

2. Stop a queue manager.

Before stopping a queue manager, all IBM MQ-related write-to-operator-with-reply (WTOR) messages must receive replies, for example, getting log requests. Each of the following commands terminates a running queue manager.

```
+CSQ1 STOP QMGR
+CSQ1 STOP QMGR MODE(QUIESCE)
+CSQ1 STOP QMGR MODE(FORCE)
+CSQ1 STOP QMGR MODE(RESTART)
```

The command STOP QMGR defaults to STOP QMGR MODE(QUIESCE).

In QUIESCE mode, IBM MQ does not allow any new connection threads to be created, but allows existing threads to continue; it terminates only when all threads have ended. Applications can request to be notified in the event of the queue manager quiescing. Therefore, use the QUIESCE mode where possible so that applications that have requested notification have the opportunity to disconnect. See [What happens during termination](#) for details.

If the queue manager does not terminate in a reasonable time in response to a STOP QMGR MODE(QUIESCE) command, use the DISPLAY CONN command to determine whether any connection threads exist, and take the necessary steps to terminate the associated applications. If there are no threads, issue a STOP QMGR MODE(FORCE) command.

The STOP QMGR MODE(QUIESCE) and STOP QMGR MODE(FORCE) commands deregister IBM MQ from the MVS Automatic Restart Manager (ARM), preventing ARM from restarting the queue manager automatically. The STOP QMGR MODE(RESTART) command works in the same way as the STOP QMGR MODE(FORCE) command, except that it does not deregister IBM MQ from ARM. This means that the queue manager is eligible for immediate automatic restart.

If the IBM MQ subsystem is not registered with ARM, the STOP QMGR MODE(RESTART) command is rejected and the following message is sent to the z/OS console:

```
CSQY205I ARM element arm-element is not registered
```

If this message is not issued, the queue manager is restarted automatically. For more information about ARM, see [“Using the z/OS Automatic Restart Manager \(ARM\)” on page 562](#).

Only cancel the queue manager address space if STOP QMGR MODE(FORCE) does not terminate the queue manager.

If a queue manager is stopped by either canceling the address space or by using the command STOP QMGR MODE(FORCE), consistency is maintained with connected CICS or IMS systems. Resynchronization of resources is started when a queue manager restarts and is completed when the connection to the CICS or IMS system is established.

Note: When you stop your queue manager, you might find message IEF352I is issued. z/OS issues this message if it detects that failing to mark the address space as unusable would lead to an integrity exposure. You can ignore this message.

Stop messages

After issuing a STOP QMGR command, you get the messages CSQY009I and CSQY002I, for example:

```
CSQY009I +CSQ1 ' STOP QMGR' COMMAND ACCEPTED FROM
USER(userid), STOP MODE(FORCE)
CSQY002I +CSQ1 QUEUE MANAGER STOPPING
```

Where `userid` is the user ID that issued the STOP QMGR command, and the MODE parameter depends on that specified in the command.

When the STOP command has completed successfully, the following messages are displayed on the z/OS console:

```
CSQ9022I +CSQ1 CSQYASCP ' STOP QMGR' NORMAL COMPLETION
CSQ3104I +CSQ1 CSQ3EC0X - TERMINATION COMPLETE
```

If you are using ARM, and you did not specify MODE(RESTART), the following message is also displayed:

```
CSQY204I +CSQ1 ARM DEREGISTER for element arm-element type
arm-element-type successful
```

You cannot restart the queue manager until the following message has been displayed:

```
CSQ3100I +CSQ1 CSQ3EC0X - SUBSYSTEM ssnm READY FOR START COMMAND
```

Issuing commands from a z/OS console or equivalent

You can issue IBM MQ MQSC and PCF commands from a z/OS console or its equivalent. You can also issue IBM MQ commands from anywhere where you can issue z/OS commands, such as SDSF or by a program using the MGCRC macro. When using the z/OS console, you add `/cpf` to the start of a command.

Before you begin

Not all commands can be issued by the z/OS console. Within the command description topics (the children of [MQSC commands reference](#)), each command that can be issued by the console is identified by the character 'C'. [Table 2 in "Command summary for IBM MQ for z/OS"](#) summarizes the MQSC commands and the sources from which they can be issued.

You cannot issue IBM MQ commands using the IMS/SSR command format from an IMS terminal. This function is not supported by the IMS adapter.

The input field provided by SDSF might not be long enough for some commands, particularly those commands for channels.

The maximum amount of data that can be displayed as a result of a command typed in at the console is 32 KB.

About this task

If you are a suitably authorized user, you can issue IBM MQ commands from the z/OS console or equivalent (such as SDSF/TSO).

When using the z/OS console, you need to add `/cpf` to the start of a command, where `cpf` is the command prefix for the queue manager subsystem.

The following steps refer to commands and attributes using their MQSC command names rather than their PCF names.

Procedure

- Use command prefix strings

Each IBM MQ command must be prefixed with a command prefix string (CPF).

Because more than one IBM MQ subsystem can run under z/OS, the CPF is used to indicate which IBM MQ subsystem processes the command.

For example, to start the queue manager for a subsystem called CSQ1, where CPF is ' +CSQ1 ', you issue the following command from the operator console:

```
+CSQ1 START QMGR
```

This CPF must be defined in the subsystem name table (for the subsystem CSQ1), as described in [Defining command prefix strings \(CPFs\)](#). In the examples, the string +CSQ1 is used as the command prefix.

- Use the z/OS console to issue commands

You can type simple commands from the z/OS console, for example:

```
+CSQ1 DISPLAY QUEUE(TRANSMIT.QUEUE.PROD) TYPE(QLLOCAL)
```

However, for complex commands or for sets of commands that you issue frequently, the other methods of issuing commands are better.

- Receive command responses

Direct responses to commands are sent to the console that issued the command. IBM MQ supports the *Extended Console Support (EMCS)* function available in z/OS, and therefore consoles with 4 byte IDs can be used. Additionally, all commands except START QMGR and STOP QMGR support the use of Command and Response Tokens (CARTs) when the command is issued by a program using the MGCRCRE macro.

Related tasks

[“Using the operations and control panels on z/OS” on page 487](#)

You use these panels for defining, displaying, altering, or deleting IBM MQ objects. Use the panels for day-to-day administration and for making small changes to objects.

[Preparing sample applications for the TSO environment on z/OS](#)

Initialization commands for IBM MQ for z/OS

Initialization commands can be used to control the queue manager startup.

Commands in the initialization input data sets are processed when IBM MQ is initialized on queue manager startup. Three types of command can be issued from the initialization input data sets:

- Commands to define IBM MQ entities that cannot be defined elsewhere, for example DEFINE BUFF-POOL.

These commands must reside in the data set identified by the DD name CSQINP1. They are processed before the restart phase of initialization. They cannot be issued through the console, operations and control panels, or an application program. The responses to these commands are written to the sequential data set that you refer to in the CSQOUT1 statement of the started task procedure.

- Commands to define IBM MQ objects that are recoverable after restart. These definitions must be specified in the data set identified by the DD name CSQINP2. They are stored in page set zero. CSQINP2 is processed after the restart phase of initialization. The responses to these commands are written to the sequential data set that you refer to in the CSQOUT2 statement of the started task procedure.
- Commands to manipulate IBM MQ objects. These commands must also be specified in the data set identified by the DD name CSQINP2. For example, the IBM MQ-supplied sample contains an ALTER QMGR command to specify a dead-letter queue for the subsystem. The response to these commands is written to the CSQOUT2 output data set.

Note: If IBM MQ objects are defined in CSQINP2, IBM MQ attempts to redefine them each time the queue manager is started. If the objects already exist, the attempt to define them fails. If you need to define your objects in CSQINP2, you can avoid this problem by using the REPLACE parameter of the DEFINE commands, however, this overrides any changes that were made during the previous run of the queue manager.

Sample initialization data set members are supplied with IBM MQ for z/OS. They are described in [Sample definitions supplied with IBM MQ](#).

Initialization commands for distributed queuing

You can also use the CSQINP2 initialization data set for the START CHINIT command. If you need a series of other commands to define your distributed queuing environment (for example, starting listeners), IBM MQ provides a third initialization input data set, called CSQINPX, that is processed as part of the channel initiator started task procedure.

The MQSC commands contained in the data set are executed at the end of channel initiator initialization, and output is written to the data set specified by the CSQOUTX DD statement. You might use the CSQINPX initialization data set to start listeners for example.

A sample channel initiator initialization data set member is supplied with IBM MQ for z/OS. It is described in [Sample definitions supplied with IBM MQ](#).

Initialization commands for publish/Subscribe

If you need a series of commands to define your publish/subscribe environment (for example, when defining subscriptions), IBM MQ provides a fourth initialization input data set, called CSQINPT.

The MQSC commands contained in the data set are executed at the end of publish/subscribe initialization, and output is written to the data set specified by the CSQOUTT DD statement. You might use the CSQINPT initialization data set to define subscriptions for example.

A sample publish/subscribe initialization data set member is supplied with IBM MQ for z/OS. It is described in [Sample definitions supplied with IBM MQ](#).

Private and global definitions on IBM MQ for z/OS

When you define an object on IBM MQ for z/OS, you can choose whether you want to share that definition with other queue managers (a *global* definition), or whether the object definition is to be used by one queue manager only (a *private* definition). This is called the object *disposition*.

Global definition

If your queue manager belongs to a queue sharing group, you can choose to share any object definitions you make with the other members of the group. This means that you have to define an object once only, reducing the total number of definitions required for the whole system.

Global object definitions are held in a *shared repository* (a Db2 shared database), and are available to all the queue managers in the queue sharing group. These objects have a disposition of GROUP.

Private definition

If you want to create an object definition that is required by one queue manager only, or if your queue manager is not a member of a queue sharing group, you can create object definitions that are not shared with other members of a queue sharing group.

Private object definitions are held on page set zero of the defining queue manager. These objects have a disposition of QMGR.

You can create private definitions for all types of IBM MQ objects except CF structures (that is, channels, namelists, process definitions, queues, queue managers, storage class definitions, and authentication information objects), and global definitions for all types of objects except queue managers.

IBM MQ automatically copies the definition of a group object to page set zero of each queue manager that uses it. You can alter the copy of the definition temporarily if you want, and IBM MQ allows you to refresh the page set copies from the repository copy if required.

IBM MQ always tries to refresh the page set copies from the repository copy at startup (for channel commands, this is done when the channel initiator restarts), or if the group object is changed.

Note: The copy of the definition is refreshed from the definition of the group, only if the definition of the group has changed after you created the copy of the definition.

This ensures that the page set copies reflect the version on the repository, including any changes that were made when the queue manager was inactive. The copies are refreshed by generating DEFINE REPLACE commands, therefore there are circumstances under which the refresh is not performed, for example:

- If a copy of the queue is open, a refresh that changes the usage of the queue fails.
- If a copy of a queue has messages on it, a refresh that deletes that queue fails.
- If a copy of a queue would require ALTER with FORCE to change it.

In these circumstances, the refresh is not performed on that copy, but is performed on the copies on all other queue managers.

If the queue manager is shut down and then restarted stand-alone, any local copies of objects are deleted, unless for example, the queue has associated messages.

There is a third object disposition that applies to local queues only. This allows you to create shared queues. The definition for a shared queue is held on the shared repository and is available to all the queue managers in the queue sharing group. In addition, the messages on a shared queue are also available to all the queue managers in the queue sharing group. This is described in [Shared queues and queue sharing groups](#). Shared queues have an object disposition of SHARED.

The following table summarizes the effect of the object disposition options for queue managers started stand-alone, and as a member of a queue sharing group.

Disposition	Stand-alone queue manager	Member of a queue sharing group
QMGR	Object definition held on page set zero.	Object definition held on page set zero.
GROUP	Not allowed.	Object definition held in the shared repository. Local copy held on page set zero of each queue manager in the group.
SHARED	Not allowed.	Queue definition held in the shared repository. Messages available to any queue manager in the group.

Manipulating global definitions

If you want to change the definition of an object that is held in the shared repository, you need to specify whether you want to change the version on the repository, or the local copy on page set zero. Use the object disposition as part of the command to do this.

Directing commands to different queue managers on z/OS

You can use the *command scope* to control on which queue manager the command runs.

You can choose to execute a command on the queue manager where it is entered, or on a different queue manager in the queue sharing group. You can also choose to issue a particular command in parallel on all the queue managers in a queue sharing group. This is possible for both MQSC commands and PCF commands.

This is determined by the *command scope*. The command scope is used with the object disposition to determine which version of an object you want to work with.

For example, you might want to alter some of the attributes of an object, the definition of which is held in the shared repository.

- You might want to change the version on one queue manager only, and not make any changes to the version on the repository or those in use by other queue managers.
- You might want to change the version in the shared repository for future users, but leave existing copies unchanged.
- You might want to change the version in the shared repository, but also want your changes to be reflected immediately on all the queue managers in the queue sharing group that hold a copy of the object on their page set zero.

Use the command scope to specify whether the command is executed on this queue manager, another queue manager, or all queue managers. Use the object disposition to specify whether the object you are manipulating is in the shared repository (a group object), or is a local copy on page set zero (a queue manager object).

You do not have to specify the command scope and object disposition to work with a shared queue because every queue manager in the queue sharing group handles the shared queue as a single queue.

z/OS

Using the operations and control panels on z/OS

You use these panels for defining, displaying, altering, or deleting IBM MQ objects. Use the panels for day-to-day administration and for making small changes to objects.

Before you begin

The IBM MQ for z/OS operations and controls panels (CSQOREXX) might not support all new function and parameters added from version 7 onwards. For example, there are no panels for the direct manipulation of topic objects or subscriptions. Use one of the following supported mechanisms to administer publish/subscribe definitions and other system controls that are not directly available from other panels:

1. IBM MQ Explorer
2. z/OS console
3. Programmable Command Format (PCF) messages
4. COMMAND function of CSQUTIL
5. IBM MQ Console

Note that the generic **Command** action in the CSQOREXX panels allows you to issue any valid MQSC command, including SMDS related commands. You can use all the commands that the COMMAND function of CSQUTIL issues.

You cannot issue the IBM MQ commands directly from the command line in the panels.

To use the operations and control panels, you must have the correct security authorization; this is described in the [User IDs for command security and command resource security](#).

You cannot provide a user ID and password using CSQUTIL, or the CSQOREXX panels. Instead, if you user ID has UPDATE authority to the BATCH profile in MQCONN, you can bypass the **CHKLOCL**(REQUIRED) setting. See [Using CHKLOCL on locally bound applications](#) for more information.

If you are setting up or changing many objects, use the COMMAND function of the CSQUTIL utility program. See [“Using the CSQUTIL utility for IBM MQ for z/OS” on page 497](#).

About this task

The operations and control panels support the controls for the channel initiator (for example, to start a channel or a TCP/IP listener), for clustering, and for security. They also enable you to display information about threads and page set usage.

The panels work by sending MQSC type IBM MQ commands to a queue manager, through the system command input queue.

Example

This is the panel that displays when you start a panel session:

```
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 . . . . . MQ1C - local queue manager or group
Target queue manager . . . . . MQ1C
                        - connected or remote queue manager for command input
Action queue manager . . . . . MQ1C - command scope in group
Response wait time . . . . . 30 5 - 999 seconds
(C) Copyright IBM Corporation 1993, 2024. All rights reserved.
Command ==>
F1=Help      F2=Split    F3=Exit     F4=Prompt   F9=SwapNext F10=Messages
F12=Cancel
```

From this panel you can perform actions such as these:

- Choose the local queue manager you want and whether you want the commands issued on that queue manager, on a remote queue manager, or on another queue manager in the same queue sharing group as the local queue manager. Over type the queue manager name if you need to change it.
- Select the action you want to perform by typing in the appropriate number in the **Action** field.
- Specify the object type that you want to work with. Press function key F1 for help about the object types if you are not sure what they are.
- Specify the disposition of the object type that you want to work with.
- Display a list of objects of the type specified. Type in an asterisk (*) in the **Name** field and press **Enter** to display a list of objects (of the type specified) that have already been defined on the action queue manager. You can then select one or more objects to work with in sequence. All the actions are available from the list.

Note: You are recommended to make choices that result in a list of objects being displayed, and then work from that list. Use the **Display** action, because that is allowed for all object types.

Invocation and rules for the operations and control panels

You can control IBM MQ and issue control commands through the ISPF panels.

How to access the IBM MQ operations and control panels

If the ISPF/PDF primary options menu has been updated for IBM MQ, you can access the IBM MQ operations and control panels from that menu. For details about updating the menu, see the [Task 20: Set up the operations and control panels](#).

You can access the IBM MQ operations and control panels from the TSO command processor panel (typically option 6 on the ISPF/PDF primary options menu). The name of the exec that you run to do this is CSQOREXX. It has two parameters; `th1qual` is the high-level qualifier for the IBM MQ libraries to be used, and `langletter` is the letter identifying the national language libraries to be used (for example, E for U.S. English). The parameters can be omitted if the IBM MQ libraries are permanently installed in your ISPF setup. Alternatively, you can issue CSQOREXX from the TSO command line.

These panels are designed to be used by operators and administrators with a minimum of formal training. Read these instructions with the panels running and try out the different tasks suggested.

Note: While using the panels, temporary dynamic queues with names of the form SYSTEM.CSQOREXX.* are created.

Rules for the operations and control panels

See [Rules for naming IBM MQ objects](#) about the general rules for IBM MQ character strings and names. However, there are some rules that apply only to the operations and control panels:

- Do not enclose strings, for example descriptions, in single or double quotation marks.
- If you include an apostrophe or quotation mark in a text field, you do not have to repeat it or add an escape character. The characters are saved exactly as you type them; for example:

```
This is Maria's queue
```

The panel processor doubles them for you to pass them to IBM MQ. However, if it has to truncate your data to do this, it does so.

- You can use uppercase or lowercase characters in most fields, and they are folded to uppercase characters when you press Enter. The exceptions are:
 - Storage class names and coupling facility structure names, which must start with uppercase A through Z and be followed by uppercase A through Z or numeric characters.
 - Certain fields that are not translated. These include:
 - Application ID
 - Description
 - Environment data
 - Object names (but if you use a lowercase object name, you might not be able to enter it at a z/OS console)
 - Remote system name
 - Trigger data
 - User data
- In names, leading blanks and leading underscores are ignored. Therefore, you cannot have object names beginning with blanks or underscores.
- Underscores are used to show the extent of blank fields. When you press Enter, trailing underscores are replaced by blanks.
- Many description and text fields are presented in multiple parts, each part being handled by IBM MQ independently. This means that trailing blanks are retained and the text is not contiguous.

Blank fields

When you specify the **Define** action for an IBM MQ object, each field on the define panel contains a value. See the general help (extended help) for the display panels for information about where IBM MQ gets the values. If you type over a field with blanks, and blanks are not allowed, IBM MQ puts the installation default value in the field or prompts you to enter the required value.

When you specify the **Alter** action for an IBM MQ object, each field on the alter panel contains the current value for that field. If you type over a field with blanks, and blanks are not allowed, the value of that field is unchanged.

Objects and actions on z/OS

The operations and control panels offer you many different types of object and a number of actions that you can perform on them.

The actions are listed on the initial panel and enable you to manipulate the objects and display information about them. These objects include all the IBM MQ objects, together with some extra ones. The objects fall into the following categories.

- [Queues, processes, authentication information objects, namelists, storage classes and CF structures](#)
- [Channels](#)
- [Cluster objects](#)
- [Queue manager and security](#)
- [Connections](#)
- [System](#)

Refer to [Actions](#) for a cross-reference table of the actions which can be taken with the IBM MQ objects.

Queues, processes, authentication information objects, namelists, storage classes and CF structures

These are the basic IBM MQ objects. There can be many of each type. They can be listed, listed with filter, defined, and deleted, and have attributes that can be displayed and altered, using the LIST or DISPLAY, LIST with FILTER, DEFINE LIKE, MANAGE, and ALTER actions. (Objects are deleted using the MANAGE action.)

This category consists of the following objects:

QLOCAL	Local queue
QREMOTE	Remote queue
QALIAS	Alias queue for indirect reference to a queue
QMODEL	Model queue for defining queues dynamically
QUEUE	Any type of queue
QSTATUS	Status of a local queue
PROCESS	Information about an application to be started when a trigger event occurs
AUTHINFO	Authentication information: definitions required to perform Certificate Revocation List (CRL) checking using LDAP servers
NAMELIST	List of names, such as queues or clusters
STGCLASS	Storage class
CFSTRUCT	coupling facility (CF) structure
CFSTATUS	Status of a CF structure

Channels

Channels are used for distributed queuing. There can be many of each type, and they can be listed, listed with filter, defined, deleted, displayed, and altered. They also have other functions available using the START, STOP and PERFORM actions. PERFORM provides reset, ping, and resolve channel functions.

This category consists of the following objects:

CHANNEL	Any type of channel
SENDER	Sender channel
SERVER	Server channel
RECEIVER	Receiver channel
REQUESTER	Requester channel
CLUSRCVR	Cluster-receiver channel
CLUSDR	Cluster-sender channel
SVRCONN	Server-connection channel
CLNTCONN	Client-connection channel

CHSTATUS Status of a channel connection

Cluster objects

Cluster objects are created automatically for queues and channels that belong to a cluster. The base queue and channel definitions can be on another queue manager. There can be many of each type, and names can be duplicated. They can be listed, listed with filter, and displayed. PERFORM, START, and STOP are also available through the LIST actions.

This category consists of the following objects:

CLUSQ	Cluster queue, created for a queue that belongs to a cluster
CLUSCHL	Cluster channel, created for a channel that belongs to a cluster
CLUSQMGR	Cluster queue manager, the same as a cluster channel but identified by its queue manager name

Cluster channels and cluster queue managers do have the PERFORM, START and STOP actions, but only indirectly through the DISPLAY action.

Queue manager and security

Queue manager and security objects have a single instance. They can be listed, and have attributes that can be displayed and altered (using the LIST or DISPLAY, and ALTER actions), and have other functions available using the PERFORM action.

This category consists of the following objects:

MANAGER	Queue manager: the PERFORM action provides suspend and resume cluster functions
SECURITY	Security functions: the PERFORM action provides refresh and reverify functions

Connection

Connections can be listed, listed with filter and displayed.

This category consists only of the connection object, CONNECT.

System

A collection of other functions. This category consists of the following objects:

SYSTEM	System functions
CONTROL	Synonym for SYSTEM

The functions available are:

LIST or DISPLAY	Display queue sharing group, distributed queuing, page set, or data set usage information.
PERFORM	Refresh or reset clustering
START	Start the channel initiator or listeners
STOP	Stop the channel initiator or listeners

Actions

The actions that you can perform for each type of object are shown in the following table:

Table 27. Valid operations and control panel actions for IBM MQ objects

Object	Alter	Define like	Manage (1)	List or Display	List with Filter	Perform	Start	Stop
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
CONNECT				X	X			
CONTROL				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			
QUEUE	X	X	X	X	X			
RECEIVER	X	X	X	X	X	X	X	X
REQUESTER	X	X	X	X	X	X	X	X
SECURITY	X			X		X		
SENDER	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
STGCLASS	X	X	X	X	X			
SYSTEM				X		X	X	X

Note:

1. Provides Delete and other functions
2. Using the List or Display action

z/OS Object dispositions on z/OS

You can specify the *disposition* of the object with which you need to work. The disposition signifies where the object **definition** is kept, and how the object behaves.

The disposition is significant only if you are working with any of the following object types:

- queues
- channels
- processes
- namelists
- storage classes
- authentication information objects

If you are working with other object types, the disposition is disregarded.

Permitted values are:

Q

QMGR. The object definitions are on the page set of the queue manager and are accessible only by the queue manager.

C

COPY. The object definitions are on the page set of the queue manager and are accessible only by the queue manager. They are local copies of objects defined as having a disposition of GROUP.

P

PRIVATE. The object definitions are on the page set of the queue manager and are accessible only by the queue manager. The objects have been defined as having a disposition of QMGR or COPY.

G

GROUP. The object definitions are in the shared repository, and are accessible by all queue managers in the queue sharing group.

S

SHARED. This disposition applies only to local queues. The queue definitions are in the shared repository, and are accessible by all queue managers in the queue sharing group.

A

ALL. If the action queue manager is either the target queue manager, or *, objects of **all** dispositions are included; otherwise, objects of QMGR and COPY dispositions only are included. This is the default.

z/OS Selecting a queue manager, defaults, and levels using the ISPF control panel on z/OS

You can use the CSQOREXX exec in ISPF to control your queue managers.

While you are viewing the initial panel, you are not connected to any queue manager. However, as soon as you press Enter, you are connected to the queue manager, or a queue manager in the queue sharing group named in the **Connect name** field. You can leave this field blank; this means that you are using the default queue manager for batch applications. This is defined in CSQBDEFV (see [Task 19: Set up Batch, TSO, and RRS adapters](#) for information about this).

Use the **Target queue manager** field to specify the queue manager where the actions you request are to be performed. If you leave this field blank, it defaults to the queue manager specified in the **Connect name** field. You can specify a target queue manager that is not the one you connect to. In this case, you would normally specify the name of a remote queue manager object that provides a queue manager alias definition (the name is used as the *ObjectQMgrName* when opening the command input queue). To do this, you must have suitable queues and channels set up to access the remote queue manager.

The **Action queue manager** field allows you to specify a queue manager that is in the same queue sharing group as the queue manager specified in the **Target queue manager** field to be the queue manager where the actions you request are to be performed. If you specify * in this field, the actions you request are

performed on all queue managers in the queue sharing group. If you leave this field blank, it defaults to the value specified in the **Target queue manager** field. The **Action queue manager** field corresponds to using the CMDSCOPE command modifier described in [The MQSC commands](#).

Queue manager defaults

If you leave any queue manager fields blank, or choose to connect to a queue sharing group, a secondary window opens when you press **Enter**. This window confirms the names of the queue managers you will be using. Press **Enter** to continue. When you return to the initial panel after having made some requests, you find fields completed with the actual names.

Queue manager levels

If the action queue manager is not at IBM MQ 8.0.0 or later, some fields are not displayed, and some values cannot be entered. A few objects and actions are disallowed. In such cases, a secondary window opens asking for you to confirm that you want to proceed.

Using the function keys and command line with the ISPF control panels on z/OS

To use the panels, you must use the function keys or enter the equivalent commands in the ISPF control panel command area.

- [Function keys](#)
 - [Processing your actions](#)
 - [“Displaying IBM MQ user messages” on page 495](#)
 - [Canceling your actions](#)
 - [Getting help](#)
- [Using the command line](#)

Function keys

The function keys have special settings for IBM MQ. (This means that you cannot use the ISPF default values for the function keys; if you have previously used the `KEYLIST OFF` ISPF command anywhere, you must type `KEYLIST ON` in the command area of any operations and control panel and then press **Enter** to enable the IBM MQ settings.)

These function key settings can be displayed on the panels, as shown in [“Using the operations and control panels on z/OS” on page 487](#). If the settings are not shown, type `PFSHOW` in the command area of any operations and control panel and then press **Enter**. To remove the display of the settings, use the command `PFSHOW OFF`.

The function key settings in the operations and control panels conform to CUA standards. Although you can change the key setting through normal ISPF procedures (such as the **KEYLIST** utility), you are not recommended to do so.

Note: Using the **PFSHOW** and **KEYLIST** commands affects any other logical ISPF screens that you have, and their settings remain when you leave the operations and control panels.

Processing your actions

Press **Enter** to carry out the action requested on a panel. The information from the panel is sent to the queue manager for processing.

Each time you press **Enter** in the panels, IBM MQ generates one or more operator messages. If the operation was successful, you get confirmation message `CSQ9022I`, otherwise you get some error messages.

Displaying IBM MQ user messages

Press function key F10 in any panel to see the IBM MQ user messages.

Canceling your actions

On the initial panel, both F3 and F12 exit the operations and control panels and return you to ISPF. No information is sent to the queue manager.

On any other panel, press function keys F3 or F12 to leave the current panel **ignoring any data you have typed since last pressing Enter**. Again, no information is sent to the queue manager.

- F3 takes you straight back to the initial panel.
- F12 takes you back to the previous panel.

Getting help

Each panel has help panels associated with it. The help panels use the ISPF protocols:

- Press function key F1 on any panel to see general help (extended help) about the task.
- Press function key F1 with the cursor on any field to see specific help about that field.
- Press function key F5 from any field help panel to get the general help.
- Press function key F3 to return to the base panel, that is, the panel from which you pressed function key F1.
- Press function key F6 from any help panel to get help about the function keys.

If the help information carries on into a second or subsequent pages, a **More** indicator is displayed in the upper-right of the panel. Use these function keys to navigate through the help pages:

- F11 to get to the next help page (if there is one).
- F10 to get back to the previous help page (if there is one).

Using the command line

You never need to use the command line to issue the commands used by the operations and control panels because they are available from function keys. The command line is provided to allow you to enter normal ISPF commands (like **PFSHOW**).

The ISPF command `PANELID ON` displays the name of the current CSQOREXX panel.

The command line is initially displayed in the default position at the bottom of the panels, regardless of what ISPF settings you have. You can use the `SETTINGS ISPF` command from any of the operations and control panels to change the position of the command line. The settings are remembered for subsequent sessions with the operations and control panels.

Using the IBM MQ for z/OS utilities

IBM MQ for z/OS provides a set of utility programs that you can use to help with system administration.

IBM MQ for z/OS supplies a set of utility programs to help you perform various administrative tasks, including the following:

- Manage message security policies.
- Perform backup, restoration, and reorganization tasks.
- Issue commands and process object definitions.
- Generate data-conversion exits.
- Modify the bootstrap data set.
- List information about the logs.
- Print the logs.

- Set up Db2 tables and other Db2 utilities.
- Process messages on the dead-letter queue.

The message security policy utility

The message security policy utility (CSQ0UTIL) runs as a stand-alone utility to manage message security policies. See [The message security policy utility \(CSQ0UTIL\)](#) for more information.

The CSQUTIL utility

This is a utility program provided to help you with backup, restore and reorganize tasks. See [“Using the CSQUTIL utility for IBM MQ for z/OS” on page 497.](#)

The data conversion exit utility

The IBM MQ for z/OS data conversion exit utility (**CSQUCVX**) runs as a stand-alone utility to create data conversion exit routines.

The change log inventory utility

The IBM MQ for z/OS change log inventory utility program (**CSQJU003**) runs as a stand-alone utility to change the bootstrap data set (BSDS). You can use the utility to perform the following functions:

- Add or delete active or archive log data sets.
- Supply passwords for archive logs.

The print log map utility

The IBM MQ for z/OS print log map utility program (**CSQJU004**) runs as a stand-alone utility to list the following information:

- Log data set name and log RBA association for both copies of all active and archive log data sets. If dual logging is not active, there is only one copy of the data sets.
- Active log data sets available for new log data.
- Contents of the queue of checkpoint records in the bootstrap data set (BSDS).
- Contents of the archive log command history record.
- System and utility time stamps.

The log print utility

The log print utility program (**CSQ1LOGP**) is run as a stand-alone utility. You can run the utility specifying:

- A bootstrap data set (BSDS)
- Active logs (with no BSDS)
- Archive logs (with no BSDS)

The queue sharing group utility

The queue sharing group utility program (**CSQ5PQSG**) runs as a stand-alone utility to set up Db2 tables and perform other Db2 tasks required for queue sharing groups.

The active log preformat utility

The active log preformat utility (**CSQJUFMT**) formats active log data sets before they are used by a queue manager. If the active log data sets are preformatted by the utility, log write performance is improved on the queue manager's first pass through the active logs.

The dead-letter queue handler utility

The dead-letter queue handler utility program (**CSQUDLQH**) runs as a stand-alone utility. It checks messages that are on the dead-letter queue and processes them according to a set of rules that you supply to the utility.

The queue load and unload utility

The queue load and unload utility copies or moves the contents of a queue, or its messages, to a file. The utility was originally shipped as the **QLOAD** utility in IBM MQ Supportpac MO03. From IBM MQ 8.0 it is integrated into the product as executable module **CSQUDMSG** in the SCSQLOAD library, with an alias of QLOAD for compatibility. Sample JCL is provided as member CSQ4QLOD in SCSQPROC.

The equivalent utility for Multiplatforms is called **dmpmqmsg**. For details of the available options, including the differences for z/OS, see [dmpmqmsg \(queue load and unload\)](#).

You can also reload messages as described in [Restoring messages from a data set to a queue \(LOAD\) on z/OS](#) and [Restoring messages from a data set to a queue \(SLOAD\) on z/OS](#).

Using the CSQUTIL utility for IBM MQ for z/OS

The CSQUTIL utility program is provided with IBM MQ for z/OS to help you perform backup, restoration, and reorganization tasks, and to issue commands and process object definitions.

About this task

Use this utility program to invoke the following functions. For example, you can issue commands from a sequential data set using the COMMAND function of the CSQUTIL utility. This function transfers the commands, as messages, to the *system-command input queue* and waits for the response, which is printed together with the original commands in SYSPRINT.

For more information about the CSQUTIL utility program, see [IBM MQ utility program \(CSQUTIL\)](#).

Procedure

- [COMMAND](#)
Use this function to issue MQSC commands, to record object definitions, and to make client-channel definition files.
- [COPY](#)
Use this function to read the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set, and put them into a sequential file and retain the original queue.
- [COPYPAGE](#)
Use this function to copy whole page sets to larger page sets.
- [EMPTY](#)
Use this function to delete the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set, retaining the definitions of the queues.
- [FORMAT](#)
Use this function to format IBM MQ for z/OS page sets.
- [Restoring messages from a data set to a queue \(LOAD\) on z/OS](#)

Use this function to restore the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set from a sequential file created by the COPY function.

- PAGEINFO

Use this function to extract page set information from one or more page sets.

- RESETPAGE

Use this function to copy whole page sets to other page set data sets and reset the log information in the copy.

- SCOPY

Use this function to copy the contents of a queue to a data set while the queue manager is offline.

- SDEFS

Use this function to produce a set of define commands for objects while the queue manager is offline.

- SLOAD

Use this function to restore messages from the destination data set of an earlier COPY or SCOPY operation. SLOAD processes a single queue.

- SWITCH

Use this function to switch or query the transmission queue associated with cluster-sender channels.

z/OS

Using the Command Facility on z/OS

Use the editor to enter or amend MQSC commands to be passed to the queue manager.

From the primary panel, CSQOPRIA, select option **8 Command**, to start the Command Facility.

You are presented with an edit session of a sequential file, *prefix*.CSQUTIL.COMMANDS, used as input to the CSQUTIL COMMAND function; see [Issuing commands to IBM MQ](#).

You do not need to prefix commands with the command prefix string (CPF).

You can continue MQSC commands on subsequent lines by terminating the current line with the continuation characters + or -. Alternatively, use line edit mode to provide long MQSC commands or the values of long attribute values within the command.

line edit

To use line edit, move the cursor to the appropriate line in the edit panel and use **F4** to display a single line in a scrollable panel. A single line can be up to 32 760 bytes of data.

To leave line edit:

- **F3 exit** saves changes made to the line and exits
- **F12 cancel** returns to the edit panel discarding changes made to the line.

To discard changes made in the edit session, use **F12 cancel** to terminate the edit session leaving the contents of the file unchanged. Commands are not executed.

Executing commands

When you have finished entering MQSC commands, terminate the edit session with **F3 exit** to save the contents of the file and invoke CSQUTIL to pass the commands to the queue manager. The output from command processing is held in file *prefix*.CSQUTIL.OUTPUT. An edit session opens automatically on this file so that you can view the responses. Press **F3 exit** to exit this session and return to the main menu.

z/OS

Working with IBM MQ objects on z/OS

Many of the tasks described in this documentation involve manipulating IBM MQ objects. The object types are queue managers, queues, process definitions, namelists, channels, client connection channels, listeners, services, and authentication information objects.

- [Defining simple queue objects](#)
- [Defining other types of objects](#)
- [Working with object definitions](#)
- [Working with namelists](#)

Defining simple queue objects

To define a new object, use an existing definition as the basis for it. You can do this in one of three ways:

- By selecting an object that is a member of a list displayed as a result of options selected on the initial panel. You then enter action type 2 (**Define like**) in the action field next to the selected object. Your new object has the attributes of the selected object, except the disposition. You can then change any attributes in your new object as you require.
- On the initial panel, select the **Define like** action type, enter the type of object that you are defining in the **Object type** field, and enter the name of a specific existing object in the **Name** field. Your new object has the same attributes as the object you named in the **Name** field, except the disposition. You can then change any attributes in your new object definition as you require.
- By selecting the **Define like** action type, specifying an object type and then leaving the **Name** field blank. You can then define your new object and it has the default attributes defined for your installation. You can then change any attributes in your new object definition as you require.

Note: You do not enter the name of the object you are defining on the initial panel, but on the **Define** panel you are presented with.

The following example demonstrates how to define a local queue using an existing queue as a template.

Defining a local queue

To define a local queue object from the operations and control panels, use an existing queue definition as the basis for your new definition. There are several panels to complete. When you have completed all the panels and you are satisfied that the attributes are correct, press Enter to send your definition to the queue manager, which then creates the actual queue.

Use the **Define like** action either on the initial panel or against an object entry in a list displayed as a result of options selected on the initial panel.

For example, starting from the initial panel, complete these fields:

Action	2 (Define like)
Object type	QLOCAL
Name	QUEUE.YOU.LIKE. This is the name of the queue that provides the attributes for your new queue.

Press Enter to display the **Define a Local Queue** panel. The queue name field is blank so that you can supply the name for the new queue. The description is that of the queue upon which you are basing this new definition. Over type this field with your own description for the new queue.

The values in the other fields are those of the queue upon which you are basing this new queue, except the disposition. You can over type these fields as you require. For example, type Y in the **Put enabled** field (if it is not already Y) if suitably authorized applications can put messages on this queue.

You get field help by moving the cursor into a field and pressing function key F1. Field help provides information about the values that can be used for each attribute.

When you have completed the first panel, press function key F8 to display the second panel.

Hints:

1. Do not press Enter at this stage, otherwise the queue will be created before you have a chance to complete the remaining fields. (If you do press Enter prematurely, do not worry; you can always alter your definition later on.)
2. Do not press function keys F3 or F12, or the data you typed will be lost.

Press function key F8 repeatedly to see and complete the remaining panels, including the trigger definition, event control, and backout reporting panels.

When your local queue definition is complete

When your definition is complete, press Enter to send the information to the queue manager for processing. The queue manager creates the queue according to the definition you have supplied. If you do not want the queue to be created, press function key F3 to exit and cancel the definition.

Defining other types of objects

To define other types of object, use an existing definition as the base for your new definition as explained in [Defining a local queue](#).

Use the **Define like** action either on the initial panel or against an object entry in a list displayed as a result of options selected on the initial panel.

For example, starting from the initial panel, complete these fields:

Action	2 (Define like)
Object type	QALIAS, NAMELIST, PROCESS, CHANNEL, and other resource objects.
Name	Leave blank or enter the name of an existing object of the same type.

Press Enter to display the corresponding DEFINE panels. Complete the fields as required and then press Enter again to send the information to the queue manager.

Like defining a local queue, defining another type of object generally requires several panels to be completed. Defining a namelist requires some additional work, as described in [“Working with namelists”](#) on page 501.

Working with object definitions

When an object has been defined, you can specify an action in the **Action** field, to alter, display, or manage it.

In each case, you can either:

- Select the object you want to work with from a list displayed as a result of options selected on the initial panel. For example, having entered 1 in the **Action** field to display objects, Queue in the **Object type** field, and * in the **Name** field, you are presented with a list of all queues defined in the system. You can then select from this list the queue with which you need to work.
- Start from the initial panel, where you specify the object you are working with by completing the **Object type** and **Name** fields.

Altering an object definition

To alter an object definition, specify action 3 and press Enter to see the ALTER panels. These panels are very similar to the DEFINE panels. You can alter the values you want. When your changes are complete, press Enter to send the information to the queue manager.

Displaying an object definition

If you want to see the details of an object without being able to change them, specify action 1 and press Enter to see the DISPLAY panels. Again, these panels are similar to the DEFINE panels except that you cannot change any of the fields. Change the object name to display details of another object.

Deleting an object

To delete an object, specify action 4 (Manage) and the **Delete** action is one of the actions presented on the resulting menu. Select the **Delete** action.

You are asked to confirm your request. If you press function key F3 or F12, the request is canceled. If you press Enter, the request is confirmed and passed to the queue manager. The object you specified is then deleted.

Note: You cannot delete most types of channel object unless the channel initiator is started.

Working with namelists

When working with namelists, proceed as you would for other objects.

For the actions DEFINE LIKE or ALTER, press function key F11 to add names to the list or to change the names in the list. This involves working with the ISPF editor and all the normal ISPF edit commands are available. Enter each name in the namelist on a separate line.

When you use the ISPF editor in this way, the function key settings are the normal ISPF settings, and **not** those used by the other operations and control panels.

If you need to specify lowercase names in the list, specify CAPS(OFF) on the editor panel command line. When you do this, all the namelists that you edit in the future are in lowercase until you specify CAPS(ON).

When you have finished editing the namelist, press function key F3 to end the ISPF edit session. Then press Enter to send the changes to the queue manager.

Attention: If you do not press Enter at this stage but press function key F3 instead, you lose any updates that you have typed in.

Implementing the system using multiple cluster transmission queues

It makes no difference if the channel is used in a single cluster, or an overlapping cluster. When the channel is selected and started, the channel selects the transmission queue depending on the definitions.

Procedure

- If you are using the DEFCLXQ option, see [“Using the automatic definition of queues and switching” on page 501.](#)
- If you are using a staged approach, see [“Changing your cluster-sender channels using a phased approach” on page 502.](#)

Using the automatic definition of queues and switching

Use this option if you are planning on using the DEFCLXQ option. There will be a queue created for every channel, and every new channel.

Procedure

1. Review the definition of the SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE and change the attributes if required.

This queue is defined in member SCSQPROC(csq4insx).

2. Create the SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE model queue.
3. Apply security policies for this model queue, and the SYSTEM.CLUSTER.TRANSMIT.** queues.

For z/OS the channel initiator started task user ID needs:

- Control access to CLASS(MQADMIN) for

```
ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channelname
```

- Update access to CLASS(MQQUEUE) for

```
ssid.SYSTEM.CLUSTER.TRANSMIT.channelname
```

Changing your cluster-sender channels using a phased approach

Use this option if you are planning on using a staged approach. This process allows you to move to the new cluster-sender channels at various times to suit the needs of your enterprise.

Before you begin

- Identify your business applications, and which channels are used.
- For the queues you use, display the clusters they are in.
- Display the channels to show the connection names, the names of the remote queue managers, and which clusters the channel supports.

About this task

- Create a transmission queue. On z/OS you might want to consider which page set you use for the queue.
- Set up security policy for the queue.
- Change any queue monitoring to include this queue name.
- Decide which channels are to use this transmission queue. The channels should have a similar name, so generic characters ' * ' in the CLCHNAME identify the channel.
- When you are ready to use the new function, alter the transmission queue to specify the name of the channels to use this transmission queue. For example CLUSTER1.TOPARIS, or CLUSTER1.* or *.TOPARIS
- Start the channels

Procedure

1. Use the DIS CLUSQMGR(xxxx) XMITQ command to display the cluster sender channels defined in the cluster, where xxxx is the name of the remote queue manager.
2. Set up the security profile for the transmission queue and give the queue access to the channel initiator.
3. Define the transmission queue to be used, and specify USAGE(XMITQ) INDXTYPE(CORRELID) SHARE and CLCHNAME(*value*)

The channel initiator started task user ID needs the following access:

```
alter class(MQADMIN) ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channel1
update class(MQQUEUE ssid.SYSTEM.CLUSTER.TRANSMIT.channel1
```

and the user ID using the SWITCH command needs the following access:

```
alter cl(MQADMIN) ssid.QUEUE.queueuname
```

4. Stop and restart the channels.

The channel change occurs when the channel starts using an MQSC command, or you use CSQUTIL. You can identify which channels need to be restarted using the SWITCH CHANNEL (*) STATUS of CSQUTIL

If you have problems when the channel is started, stop the channel, resolve the problems, and restart the channel.

Note that you can change the CLCHNAME attribute as often as you need to.

The value of CLCHNAME used is the one when the channel is started, so you can change the CLCHNAME definition while the channel continues to use the definitions from the time that it started. The channel uses the new definition when it is restarted.

Undoing a change to a transmission queue on z/OS

You need to have a process to backout a change if the results are not as you expect.

What can go wrong?

If the new transmission queue is not what you expect:

1. Check the CLCHNAME is as you expect
2. Review the job log to check if the switch process has finished. If not, wait and check the new transmission queue of the channel later.

If you are using multiple cluster transmission queues, it is important that you design the transmission queues definitions explicitly and avoid complicated overlapping configuration. In this way, you can make sure that if there are problems, you can go back to the original queues and configuration.

If you encounter problems during the move to using a different transmission queue, you must resolve any problems before you can proceed with the change.

An existing change request must complete before a new change request can be made. For example, you:

1. Define a new transmission queue with a maximum depth of one and there are 10 messages waiting to be sent.
2. Change the transmission queue to specify the channel name in the CLCHNAME parameter.
3. Stop and restart the channel. The attempt to move the messages fails and reports the problems.
4. Change the CLCHNAME parameter on the transmission queue to be blank.
5. Stop and restart the channel. The channel continues to try and complete the original request, so the channel continues to use the new transmission queue.
6. Need to resolve the problems and restart the channel so the moving of messages completes successfully.

Next time the channel is restarted it picks up any changes, so if you had set CLCHNAME to blanks, the channel will not use the specified transmission queue.

In this example, changing the CLCHNAME on the transmission queue to blanks does not necessarily mean that the channel uses the SYSTEM.CLUSTER.TRANSMIT queue, as there might be other transmission queues whose CLCHNAME parameter match the channel name. For example, a generic name, or the queue manager attribute DEFCLXQ might be set to channel, so the channel uses a dynamic queue instead of the SYSTEM.CLUSTER.TRANSMIT queue.

Writing programs to administer IBM MQ for z/OS

You can write your own application programs to administer a queue manager. Use this topic to understand the requirements for writing your own administration programs.

Start of General-use programming interface information

This set of topics contains hints and guidance to enable you to issue IBM MQ commands from an IBM MQ application program.

Note: In this topic, the MQI calls are described using C-language notation. For typical invocations of the calls in the COBOL, PL/I, and assembler languages, see [Function calls](#).

Understanding how it all works

In outline, the procedure for issuing commands from an application program is as follows:

1. Build an IBM MQ command into a type of IBM MQ message called a *request message*. The command can be in MQSC or PCF format.
2. Send (use MQPUT) this message to a special queue called the system-command input queue. The IBM MQ command processor runs the command.
3. Retrieve (use MQGET) the results of the command as *reply messages* on the reply-to queue. These messages contain the user messages that you need to determine whether your command was successful and, if it was, what the results were.

Then it is up to your application program to process the results.

This set of topics contains:

Preparing queues for administration programs

Administration programs require a number of predefined queues for system command input and receiving responses.

This section applies to commands in the MQSC format. For the equivalent in PCF, see [“IBM MQ Programmable Command Formats verwenden”](#) on page 28.

Before you can issue any MQPUT or MQGET calls, you must first define, and then open, the queues you are going to use.

Defining the system-command input queue

The system-command input queue is a local queue called SYSTEM.COMMAND.INPUT. The supplied CSQINP2 initialization data set, thlqual.SCSQPROC(CSQ4INSG), contains a default definition for the system-command input queue. For compatibility with IBM MQ on other platforms, an alias of this queue, called SYSTEM.ADMIN.COMMAND.QUEUE is also supplied. See [Sample definitions supplied with IBM MQ](#) for more information.

Defining a reply-to queue

You must define a reply-to queue to receive reply messages from the IBM MQ command processor. It can be any queue with attributes that allow reply messages to be put on it. However, for normal operation, specify these attributes:

- USAGE(NORMAL)
- NOTRIGGER (unless your application uses triggering)

Avoid using persistent messages for commands, but if you choose to do so, the reply-to queue must not be a temporary dynamic queue.

The supplied CSQINP2 initialization data set, thlqual.SCSQPROC(CSQ4INSG), contains a definition for a model queue called SYSTEM.COMMAND.REPLY.MODEL. You can use this model to create a dynamic reply-to queue.

Note: Replies generated by the command processor can be up to 15 000 bytes in length.

If you use a permanent dynamic queue as a reply-to queue, your application should allow time for all PUT and GET operations to complete before attempting to delete the queue, otherwise MQRC2055 (MQRC_Q_NOT_EMPTY) can be returned. If this occurs, try the queue deletion again after a few seconds.

Opening the system-command input queue

Before you can open the system-command input queue, your application program must be connected to your queue manager. Use the MQI call MQCONN or MQCONNX to do this.

Then use the MQI call MQOPEN to open the system-command input queue. To use this call:

1. Set the **Options** parameter to MQOO_OUTPUT
2. Set the MQOD object descriptor fields as follows:

ObjectType

MQOT_Q (the object is a queue)

ObjectName

SYSTEM.COMMAND.INPUT

ObjectQMgrName

If you want to send your request messages to your local queue manager, leave this field blank. This means that your commands are processed locally.

If you want your IBM MQ commands to be processed on a remote queue manager, put its name here. You must also have the correct queues and links set up, as described in [Distributed queuing and clusters](#).

Opening a reply-to queue

To retrieve the replies from an IBM MQ command, you must open a reply-to queue. One way of doing this is to specify the model queue, SYSTEM.COMMAND.REPLY.MODEL in an MQOPEN call, to create a permanent dynamic queue as the reply-to queue. To use this call:

1. Set the **Options** parameter to MQOO_INPUT_SHARED
2. Set the MQOD object descriptor fields as follows:

ObjectType

MQOT_Q (the object is a queue)

ObjectName

The name of the reply-to queue. If the queue name you specify is the name of a model queue object, the queue manager creates a dynamic queue.

ObjectQMgrName

To receive replies on your local queue manager, leave this field blank.

DynamicQName

Specify the name of the dynamic queue to be created.

Using the command server

The command server is an IBM MQ component that works with the command processor component. You can send formatted messages to the command server which interprets the messages, runs the administration requests, and sends responses back to your administration application.

The command server reads request messages from the system-command input queue, verifies them, and passes the valid ones as commands to the command processor. The command processor processes the commands and puts any replies as reply messages on to the reply-to queue that you specify. The first reply message contains the user message CSQN205I. See [“Interpreting the reply messages from the command server”](#) on page 509 for more information. The command server also processes channel initiator and queue sharing group commands, wherever they are issued from.

Identifying the queue manager that processes your commands

The queue manager that processes the commands you issue from an administration program is the queue manager that owns the system-command input queue that the message is put onto.

Starting the command server

Normally, the command server is started automatically when the queue manager is started. It becomes available as soon as the message CSQ9022I 'START QMGR' NORMAL COMPLETION is returned

from the START QMGR command. The command server is stopped when all the connected tasks have been disconnected during the system termination phase.

You can control the command server yourself using the START CMDSERV and STOP CMDSERV commands. To prevent the command server starting automatically when IBM MQ is restarted, you can add a STOP CMDSERV command to your CSQINP1 or CSQINP2 initialization data sets. However, this is not recommended as it prevents any channel initiator or queue sharing group commands being processed.

The STOP CMDSERV command stops the command server as soon as it has finished processing the current message, or immediately if no messages are being processed.

If the command server has been stopped by a STOP CMDSERV command in the program, no other commands from the program can be processed. To restart the command server, you must issue a START CMDSERV command from the z/OS console.

If you stop and restart the command server while the queue manager is running, all the messages that are on the system-command input queue when the command server stops are processed when the command server is restarted. However, if you stop and restart the queue manager after the command server is stopped, only the persistent messages on the system-command input queue are processed when the command server is restarted. All nonpersistent messages on the system-command input queue are lost.

Sending commands to the command server

For each command, you build a message containing the command, then put it onto the system-command input queue.

Building a message that includes IBM MQ commands

You can incorporate IBM MQ commands in an application program by building request messages that include the required commands. For each such command you:

1. Create a buffer containing a character string representing the command.
2. Issue an MQPUT call specifying the buffer name in the **buffer** parameter of the call.

The simplest way to do this in C is to define a buffer using 'char'. For example:

```
char message_buffer[ ] = "ALTER QLOCAL(SALES) PUT(ENABLED)";
```

When you build a command, use a null-terminated character string. Do not specify a command prefix string (CPF) at the start of a command defined in this way. This means that you do not have to alter your command scripts if you want to run them on another queue manager. However, you must take into account that a CPF is included in any response messages that are put onto the reply-to queue.

The command server folds all lowercase characters to uppercase unless they are inside quotation marks.

Commands can be any length up to a maximum 32 762 characters.

Putting messages on the system-command input queue

Use the MQPUT call to put request messages containing commands on the system-command input queue. In this call you specify the name of the reply-to queue that you have already opened.

To use the MQPUT call:

1. Set these MQPUT parameters:

Hconn

The connection handle returned by the MQCONN or MQCONNX call.

Hobj

The object handle returned by the MQOPEN call for the system-command input queue.

BufferLength

The length of the formatted command.

Buffer

The name of the buffer containing the command.

2. Set these MQMD fields:

MsgType

MQMT_REQUEST

Format

MQFMT_STRING or MQFMT_NONE

If you are not using the same code page as the queue manager, set *CodedCharSetId* as appropriate and set MQFMT_STRING, so that the command server can convert the message. Do not set MQFMT_ADMIN, as that causes your command to be interpreted as PCF.

ReplyToQ

Name of your reply-to queue.

ReplyToQMgr

If you want replies sent to your local queue manager, leave this field blank. If you want your IBM MQ commands to be sent to a remote queue manager, put its name here. You must also have the correct queues and links set up, as described in [Distributed queuing and clusters](#).

3. Set any other MQMD fields, as required. You should normally use nonpersistent messages for commands.
4. Set any *PutMsgOpts* options, as required.

If you specify MQPMO_SYNCPOINT (the default), you must follow the MQPUT call with a syncpoint call.

Using MQPUT1 and the system-command input queue

If you want to put just one message on the system-command input queue, you can use the **MQPUT1** call. This call combines the functions of an **MQOPEN**, followed by an **MQPUT** of one message, followed by an **MQCLOSE**, all in one call. If you use this call, modify the parameters accordingly. See [Putting one message on a queue using the MQPUT1 call](#) for details.

Retrieving replies to your commands

The command server sends a response to a reply queue for each request message it receives. Any administration application must receive, and handle the reply messages.

When the command processor processes your commands, any reply messages are put onto the reply-to queue specified in the MQPUT call. The command server sends the reply messages with the same persistence as the command message it received.

Waiting for a reply

Use the MQGET call to retrieve a reply from your request message. One request message can produce several reply messages. For details, see [“Interpreting the reply messages from the command server” on page 509](#).

You can specify a time interval that an MQGET call waits for a reply message to be generated. If you do not get a reply, use the checklist beginning in topic [“If you do not receive a reply” on page 510](#).

To use the MQGET call:

1. Set these parameters:

Hconn

The connection handle returned by the MQCONN or MQCONNX call.

Hobj

The object handle returned by the MQOPEN call for the reply-to queue.

Buffer

The name of the area to receive the reply.

BufferLength

The length of the buffer to receive the reply. This must be a minimum of 80 bytes.

- To ensure that you only get the responses from the command that you issued, you must specify the appropriate *MsgId* and *CorrelId* fields. These depend on the report options, MQMD_REPORT, you specified in the MQPUT call:

MQRO_NONE

Binary zero, '00...00' (24 nulls).

MQRO_NEW_MSG_ID

Binary zero, '00...00' (24 nulls).

This is the default if none of these options has been specified.

MQRO_PASS_MSG_ID

The *MsgId* from the MQPUT.

MQRO_NONE

The *MsgId* from the MQPUT call.

MQRO_COPY_MSG_ID_TO_CORREL_ID

The *MsgId* from the MQPUT call.

This is the default if none of these options has been specified.

MQRO_PASS_CORREL_ID

The *CorrelId* from the MQPUT call.

For more details on report options, see [Report options and message flags](#).

- Set the following *GetMsgOpts* fields:

Options

MQGMO_WAIT

If you are not using the same code page as the queue manager, set MQGMO_CONVERT, and set *CodedCharSetId* as appropriate in the MQMD.

WaitInterval

For replies from the local queue manager, try 5 seconds. Coded in milliseconds, this becomes 5 000. For replies from a remote queue manager, and channel control and status commands, try 30 seconds. Coded in milliseconds, this becomes 30 000.

Discarded messages

If the command server finds that a request message is not valid, it discards this message and writes the message `CSQN205I` to the named reply-to queue. If there is no reply-to queue, the `CSQN205I` message is put onto the dead-letter queue. The return code in this message shows why the original request message was not valid:

- 00D5020F** It is not of type MQMT_REQUEST.
- 00D50210** It has zero length.
- 00D50212** It is longer than 32 762 bytes.
- 00D50211** It contains all blanks.
- 00D5483E** It needed converting, but *Format* was not MQFMT_STRING.
- Other** See [Command server codes](#)

The command server reply message descriptor

For any reply message, the following MQMD message descriptor fields are set:

<i>MsgType</i>	MQMT_REPLY
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>Priority</i>	As for the MQMD in the message you issued.
<i>Persistence</i>	As for the MQMD in the message you issued.
<i>CorrelId</i>	Depends on the MQPUT report options.
<i>ReplyToQ</i>	None.

The command server sets the *Options* field of the MQPMO structure to MQPMO_NO_SYNCPOINT. This means that you can retrieve the replies as they are created, rather than as a group at the next syncpoint.

Interpreting the reply messages from the command server

Each request message correctly processed by IBM MQ produces at least two reply messages. Each reply message contains a single IBM MQ user message.

The length of a reply depends on the command that was issued. The longest reply you can get is from a **DISPLAY NAMELIST** command, and that can be up to 15 000 bytes in length.

The first user message, CSQN205I, always contains:

- A count of the replies (in decimal), which you can use as a counter in a loop to get the rest of the replies. The count includes this first message.
- The return code from the command preprocessor.
- A reason code, which is the reason code from the command processor.

This message does not contain a CPF.

For example:

```
CSQN205I    COUNT=    4, RETURN=0000000C, REASON=00000008
```

The COUNT field is 8 bytes long and is right-justified. It always starts at position 18, that is, immediately after COUNT=. The RETURN field is 8 bytes long in character hexadecimal and is immediately after RETURN= at position 35. The REASON field is 8 bytes long in character hexadecimal and is immediately after REASON= at position 52.

If the RETURN= value is 00000000 and the REASON= value is 00000004, the set of reply messages is incomplete. After retrieving the replies indicated by the CSQN205I message, issue a further MQGET call to wait for a further set of replies. The first message in the next set of replies is again CSQN205I, indicating how many replies there are, and whether there are still more to come.

See the [IBM MQ for z/OS -Nachrichten, -Beendigungs-codes und -Ursachencodes](#) documentation for more details about the individual messages.

If you are using a non-English language feature, the text and layout of the replies are different from those shown here. However, the size and position of the count and return codes in message CSQN205I are the same.

If you do not receive a reply

There are a series of steps you can take if you do not receive a response to request to the command server.

If you do not receive a reply to your request message, work through this checklist:

- Is the command server running?
- Is the *WaitInterval* long enough?
- Are the system-command input and reply-to queues correctly defined?
- Were the MQOPEN calls to these queues successful?
- Are both the system-command input and reply-to queues enabled for MQPUT and MQGET calls?
- Have you considered increasing the MAXDEPTH and MAXMSGL attributes of your queues?
- Are you are using the *CorrelId* and *MsgId* fields correctly?
- Is the queue manager still running?
- Was the command built correctly?
- Are all your remote links defined and operating correctly?
- Were the MQPUT calls correctly defined?
- Has the reply-to queue been defined as a temporary dynamic queue instead of a permanent dynamic queue? (If the request message is persistent, you must use a permanent dynamic queue for the reply.)

When the command server generates replies but cannot write them to the reply-to queue that you specify, it writes them to the dead-letter queue.

Passing commands using MGCRE

With appropriate authorization, an application program can make requests to multiple queue managers using a z/OS service routine.

If you have the correct authorization, you can pass IBM MQ commands from your program to multiple queue managers by the MGCRE (SVC 34) z/OS service. See the [z/OS MVS Programming: Authorized Assembler Services Guide](#) for more information.

The value of the CPF identifies the particular queue manager to which the command is directed. For information about CPFs, see [User IDs for command security and command resource security](#) and [“Issuing queue manager commands on z/OS” on page 473](#).

If you use MGCRE, you can use a Command and Response Token (CART) to get the direct responses to the command.

Examples of commands and their replies

Use this topic as a series of examples of commands to the command server and the responses from the command server.

Here are some examples of commands that could be built into IBM MQ messages, and the user messages that are the replies. Unless otherwise stated, each line of the reply is a separate message.

- [Messages from a DEFINE command](#)
- [Messages from a DELETE command](#)
- [Messages from DISPLAY commands](#)
- [Messages from commands with CMDSCOPE](#)
- [Messages from commands that generate commands with CMDSCOPE](#)

Messages from a DEFINE command

The following command:

```
DEFINE QLOCAL(Q1)
```

produces these messages:

```
CSQN205I    COUNT=    2, RETURN=00000000, REASON=00000000  
CSQ9022I +CSQ1 CSQMMSGP ' DEFINE QLOCAL' NORMAL COMPLETION
```

These reply messages are produced on normal completion.

Messages from a DELETE command

The following command:

```
DELETE QLOCAL(Q2)
```

produces these messages:

```
CSQN205I    COUNT=    4, RETURN=00000000, REASON=00000008  
CSQM125I +CSQ1 CSQMUQLC QLOCAL (Q2) QSGDISP(QMGR) WAS NOT FOUND  
CSQM090E +CSQ1 CSQMUQLC FAILURE REASON CODE X'00D44002'  
CSQ9023E +CSQ1 CSQMUQLC ' DELETE QLOCAL' ABNORMAL COMPLETION
```

These messages indicate that a local queue called Q2 does not exist.

Messages from DISPLAY commands

The following examples show the replies from some DISPLAY commands.

Finding out the name of the dead-letter queue

If you want to find out the name of the dead-letter queue for a queue manager, issue this command from an application program:

```
DISPLAY QMGR DEADQ
```

The following three user messages are returned, from which you can extract the required name:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000  
CSQM409I +CSQ1 QMNAME(CSQ1) DEADQ(SYSTEM.DEAD.QUEUE      )  
CSQ9022I +CSQ1 CSQMDRTS ' DISPLAY QMGR' NORMAL COMPLETION
```

Messages from the DISPLAY QUEUE command

The following examples show how the results from a command depend on the attributes specified in that command.

Example 1

You define a local queue using the command:

```
DEFINE QLOCAL(Q1) DESCR('A sample queue') GET(ENABLED) SHARE
```

If you issue the following command from an application program:

```
DISPLAY QUEUE(Q1) SHARE GET DESCR
```

these three user messages are returned:

```
CSQN205I    COUNT=    3, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(Q1
QLOCAL ) QSGDISP(QMGR )
DESCR(A sample queue
) SHARE GET(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

Note: The second message, CSQM401I, is shown here occupying four lines.

Example 2

Two queues have names beginning with the letter A:

- A1 is a local queue with its PUT attribute set to DISABLED.
- A2 is a remote queue with its PUT attribute set to ENABLED.

If you issue the following command from an application program:

```
DISPLAY QUEUE(A*) PUT
```

these four user messages are returned:

```
CSQN205I    COUNT=    4, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(A1
QLOCAL ) QSGDISP(QMGR )
PUT(DISABLED )
CSQM406I +CSQ1 QUEUE(A2
QREMOTE ) PUT(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION
```

Note: The second and third messages, CSQM401I and CSQM406I, are shown here occupying three and two lines.

Messages from the DISPLAY NAMELIST command

You define a namelist using the command:

```
DEFINE NAMELIST(N1) NAMES(Q1,SAMPLE_QUEUE)
```

If you issue the following command from an application program:


```
DISPLAY NAMELIST(N1) NAMES NAMCOUNT
```

the following three user messages are returned:

```
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
```

Note: The second message, CSQM407I, is shown here occupying three lines.

Messages from commands with CMDSCOPE

The following examples show the replies from commands that have been entered with the CMDSCOPE attribute.

Messages from the ALTER PROCESS command

The following command:

```
ALT PRO(V4) CMDSCOPE(*)
```

produces the following messages:

```
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
```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). The command was successful on MQ25 but the process definition did not exist on MQ26, so the command failed on that queue manager.

Messages from the DISPLAY PROCESS command

The following command:

```
DIS PRO(V*) CMDSCOPE(*)
```

produces the following messages:

```

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

```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Information is displayed about all the processes on each queue manager with names starting with the letter V.

Messages from the DISPLAY CHSTATUS command

The following command:

```
DIS CHS(VT) CMDSCOPE(*)
```

produces the following messages:

```

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

```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Information is displayed about channel status on each queue manager.

Messages from the STOP CHANNEL command

The following command:

```
STOP CHL(VT) CMDSCOPE(*)
```

produces these messages:

```

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

```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Channel VT was stopped on each queue manager.

Messages from commands that generate commands with CMDSCOPE

The following command:

```
DEF PRO(V2) QSGDISP(GROUP)
```

produces these messages:

```

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

```

These messages tell you that the command was entered on queue manager MQ25. When the object was created on the shared repository, another command was generated and sent to all the active queue managers in the queue sharing group (MQ25 and MQ26).

z/OS

Managing IBM MQ resources on z/OS

Use the links in this topic to find out how to manage the resources used by IBM MQ for z/OS, for example, managing log files, data sets, page sets, buffer pools, and coupling facility structures.

Use the following links for details of the different administrative tasks you might have to complete while using IBM MQ for z/OS:

- [“Managing the logs” on page 516](#)
- [“Managing the bootstrap data set \(BSDS\)” on page 525](#)
- [“Managing page sets” on page 532](#)
- [“How to back up and recover page sets” on page 539](#)
- [“How to back up and restore queues using CSQUTIL” on page 542](#)
- [“Managing buffer pools” on page 542](#)

- [“Managing queue sharing groups and shared queues on z/OS” on page 544](#)

Related concepts

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

[“Administering IBM MQ for z/OS” on page 472](#)

IBM MQ for z/OS can be controlled and managed by MQSC and PCF commands, by a set of utilities and programs provided with the product, and by authorized applications.

[“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS” on page 473](#)

You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications. Not all commands can be issued from all these sources.

[“Recovery and restart on z/OS” on page 553](#)

Use this topic to understand the recovery and restart mechanisms used by IBM MQ.

Related tasks

[Planning your IBM MQ environment on z/OS](#)

[Configuring queue managers on z/OS](#)

[IBM MQ utilities on z/OS reference](#)

Related reference

[“Using the IBM MQ for z/OS utilities” on page 495](#)

IBM MQ for z/OS provides a set of utility programs that you can use to help with system administration.

[Programmable command formats reference](#)

Managing the logs

Use this topic to understand how to manage your IBM MQ log files, including the log archiving process, using log record compression, log record recovery, and printing log records.

This topic describes the tasks involved in managing the IBM MQ logs. It contains these sections:

Archiving logs with the ARCHIVE LOG command

An authorized operator can archive the current IBM MQ active log data sets whenever required using the **ARCHIVE LOG** command.

When you issue the ARCHIVE LOG command, IBM MQ truncates the current active log data sets, then runs an asynchronous offload process, and updates the BSDS with a record of the offload process.

The **ARCHIVE LOG** command has a **MODE(QUIESCE)** option. With this option, IBM MQ jobs and users are quiesced after a commit point, and the resulting point of consistency is captured in the current active log before it is offloaded.

Consider using the **MODE(QUIESCE)** option when planning a backup strategy for off site recovery. It creates a system-wide point of consistency, which minimizes the number of data inconsistencies when the archive log is used with the most current backup page set copy during recovery. For example:

```
ARCHIVE LOG MODE(QUIESCE)
```

If you issue the **ARCHIVE LOG** command without specifying a **TIME** parameter, the quiesce time period defaults to the value of the **QUIESCE** parameter of the CSQ6ARVP macro. If the time required for the ARCHIVE LOG MODE(QUIESCE) to complete is less than the time specified, the command completes successfully; otherwise, the command fails when the time period expires. You can specify the time period explicitly by using the **TIME** option, for example:

```
ARCHIVE LOG MODE(QUIESCE) TIME(60)
```

This command specifies a quiesce period of up to 60 seconds before **ARCHIVE LOG** processing occurs.

Attention: Using the **TIME** option when time is critical can significantly disrupt IBM MQ availability for all jobs and users that use IBM MQ resources.

By default, the command is processed asynchronously from the time you submit the command. (To process the command synchronously with other IBM MQ commands use the **WAIT (YES)** option with **QUIESCE**, but be aware that the z/OS console is locked from IBM MQ command input for the entire **QUIESCE** period.)

During the quiesce period:

- Jobs and users on the queue manager are allowed to go through commit processing, but are suspended if they try to update any IBM MQ resource after the commit.
- Jobs and users that only read data can be affected, since they might be waiting for locks held by jobs or users that were suspended.
- New tasks can start, but they cannot update data.

The output from the **DISPLAY LOG** command uses the message CSQV400I to indicate that a quiesce is in effect.

For example:

```
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 zHyperWrite Encrypted DSName
  1   68 NO          NO          VICY.CSQ1.LOGCOPY1.DS01
  2   68 NO          NO          VICY.CSQ1.LOGCOPY2.DS01
Restarted at 2019-08-15 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
```

When all updates are quiesced, the quiesce history record in the BSDS is updated with the date and time that the active log data sets were truncated, and with the last-written RBA in the current active log data sets. IBM MQ truncates the current active log data sets, switches to the next available active log data sets, and issues message CSQJ311I stating that the offload process started.

If updates cannot be quiesced before the quiesce period expires, IBM MQ issues message CSQJ317I, and **ARCHIVE LOG** processing terminates. The current active log data sets are not truncated, nor switched to the next available log data sets, and the offload process is not started.

Whether the quiesce was successful or not, all suspended users and jobs are then resumed, and IBM MQ issues message CSQJ312I, stating that the quiesce is ended and update activity is resumed.

If **ARCHIVE LOG** is issued when the current active log is the last available active log data set, the command is not processed, and IBM MQ issues the following message:

```
CSQJ319I - csect-name CURRENT ACTIVE LOG DATA SET IS THE LAST
AVAILABLE ACTIVE LOG DATA SET. ARCHIVE LOG PROCESSING
WILL BE TERMINATED
```

If **ARCHIVE LOG** is issued when another **ARCHIVE LOG** command is already in progress, the new command is not processed, and IBM MQ issues the following message:

CSQJ318I - ARCHIVE LOG COMMAND ALREADY IN PROGRESS

For information about the messages issued during archiving, see [Messages for IBM MQ for z/OS](#).

Restarting the log archive process after a failure

If there is a problem during the log archive process (for example, a problem with allocation or tape mounts), the archiving of the active log might be suspended. You can cancel the archive process and restart it by using the following command:

```
ARCHIVE LOG CANCEL OFFLOAD
```

This command cancels any offload processing currently in progress, and restarts the archive process. It starts with the oldest log data set that has not been archived, and proceeds through all active log data sets that need offloading. Any log archive operations that have been suspended are restarted.

Use this command only if you are sure that the current log archive task is no longer functioning, or if you want to restart a previous attempt that failed. This is because the command might cause an abnormal termination of the offload task, which might result in a dump.

Controlling archiving and logging

You can control compression, printing, archiving, recovery and logging with using the CSQ6LOGP, CSQ6ARVP, and CSQ6SYSP macros. Note, that changes to private objects only are logged in IBM MQlogs. Changes to GROUP objects (like shared inbound channels) are also logged, because the definitions are propagated around the group and held locally.

Many aspects of archiving and logging are controlled by parameters set using the CSQ6LOGP, CSQ6ARVP and CSQ6SYSP macros of the system parameter module when the queue manager is customized. See [Tailor your system parameter module](#) for details of these macros.

Some of these parameters can be changed while a queue manager is running using the IBM MQ MQSC SET LOG, SET SYSTEM and SET ARCHIVE commands. They are shown in [Table 28 on page 518](#):

SET command	Parameters
LOG	WRTHRSH, MAXARCH, DEALLCT, MAXRTU, COMPLOG
ARCHIVE	All
SYSTEM	LOGLOAD

You can display the settings of all the parameters using the MQSC [DISPLAY LOG](#), [DISPLAY ARCHIVE](#) and [DISPLAY SYSTEM](#) commands. These commands also show status information about archiving and logging.

Controlling log compression

You can enable and disable the compression of log records using either

- The SET and DISPLAY LOG commands in MQSC; see [The MQSC commands](#)
- Invoking PCF interface. See [“Einführung in IBM MQ Programmable Command Formats” on page 27](#)
- Using the CSQ6LOGP macro in the system parameter module; see [Using CSQ6LOGP](#)

Printing log records

You can extract and print log records using the CSQ1LOGP utility. For instructions, see [The log print utility](#).

Recovering logs

Normally, you do not need to back up and restore the IBM MQ logs, especially if you are using dual logging. However, in rare circumstances, such as an I/O error on a log, you might need to recover the logs. Use Access Method Services to delete and redefine the data set, and then copy the corresponding dual log into it.

Discarding archive log data sets

You can discard your archive log data sets and choose to discard the logs automatically or manually.

You must keep enough log data to be able to perform unit of work recovery, page set media recovery if a page set is lost, or CF structure media recovery if a CF structure is lost. Do not discard archive log data sets that might be required for recovery; if you discard these archive log data sets you might not be able to perform required recovery operations.

If you have confirmed that your archive log data sets can be discarded, you can do this in either of the following ways:

- [Automatic archive log data set deletion](#)
- [Manually deleting archive log data sets](#)

Automatic archive log data set deletion

You can use a DASD or tape management system to delete archive log data sets automatically. The retention period for IBM MQ archive log data sets is specified by the retention period field ARCRETN in the CSQ6ARVP installation macro (see the [Using CSQ6ARVP](#) for more information).

The default for the retention period specifies that archive logs are to be kept for 9999 days (the maximum).

Important: You can change the retention period but you must ensure that you can accommodate the number of backup cycles that you have planned for.

.

IBM MQ uses the retention period value as the value for the JCL parameter RETPD when archive log data sets are created.

The retention period set by the MVS™/DFP storage management subsystem (SMS) can be overridden by this IBM MQ parameter. Typically, the retention period is set to the smaller value specified by either IBM MQ or SMS. The storage administrator and IBM MQ administrator must agree on a retention period value that is appropriate for IBM MQ.

Note: IBM MQ does not have an automated method to delete information about archive log data sets from the BSDS, because some tape management systems provide external manual overrides of retention periods. Therefore, information about an archive log data set can still be in the BSDS long after the data set retention period has expired and the data set has been scratched by the tape management system. Conversely, the maximum number of archive log data sets might have been exceeded and the data from the BSDS might have been dropped before the data set has reached its expiration date.

If archive log data sets are deleted automatically, remember that the operation does not update the list of archive logs in the BSDS. You can update the BSDS with the change log inventory utility, as described in “[Changing the BSDS](#)” on page 526. The update is not essential. Recording old archive logs wastes space in the BSDS, but does no other harm.

Manually deleting archive log data sets

You must keep all the log records as far back as the lowest RBA identified in messages CSQI024I and CSQI025I. This RBA is obtained using the DISPLAY USAGE command that you issued when creating a point of recovery using [Method 1: Full backup](#).

Read [Creating a point of recovery for non-shared resources before discarding any logs](#).

Locate and discard archive log data sets

Having established the minimum log RBA required for recovery, you can find archive log data sets that contain only earlier log records by performing the following procedure:

1. Use the print log map utility to print the contents of the BSDS. For an example of the output, see [The print log map utility](#).
2. Find the sections of the output titled ARCHIVE LOG COPY n DATA SETS. If you use dual logging, there are two sections. The columns labeled STARTRBA and ENDRBA show the range of RBAs contained in each volume. Find the volumes with ranges that include the minimum RBA you found with messages CSQI024I and CSQI025I. These are the earliest volumes you need to keep. If you are using dual-logging, there are two such volumes.

If no volumes have an appropriate range, one of the following cases applies:

- The minimum RBA has not yet been archived, and you can discard all archive log volumes.
- The list of archive log volumes in the BSDS wrapped around when the number of volumes exceeded the number allowed by the MAXARCH parameter of the CSQ6LOGP macro. If the BSDS does not register an archive log volume, that volume cannot be used for recovery. Therefore, consider adding information about existing volumes to the BSDS. For instructions, see [“Changes for archive logs” on page 528](#).

Also consider increasing the value of MAXARCH. For information, see the [Using CSQ6LOGP](#).

3. Delete any archive log data set or volume with an ENDRBA value that is less than the STARTRBA value of the earliest volume you want to keep. If you are using dual logging, delete both such copies.

Because BSDS entries wrap around, the first few entries in the BSDS archive log section might be more recent than the entries at the end. Look at the combination of date and time and compare their ages. Do not assume that you can discard all entries before the entry for the archive log containing the minimum LOGRBA.

Delete the data sets. If the archives are on tape, erase the tapes. If they are on DASD, run a z/OS utility to delete each data set. Then, if you want the BSDS to list only existing archive volumes, use the change log inventory utility (CSQJU003) to delete entries for the discarded volumes. See [“Changes for archive logs” on page 528](#) for an example.

The effect of log shunting

Long running transactions can cause unit of work log records which span log data sets. IBM MQ handles this scenario by using log shunting, a technique which moves the log records to optimize the quantity of log data retained, and queue manager restart time.

When a unit of work is considered to be long, a representation of each log record is written further down the log. This is known as *log shunting*. It is described more fully in [Log files](#).

The queue manager uses these shunted log records instead of the originals after a failure, to ensure unit of work integrity. There are two benefits to this:

- the quantity of log data which must be retained for unit of work coordination is reduced
- less log data must be traversed at queue manager restart time, so the queue manager is restarted more quickly

Shunted log records do not contain sufficient information for media recovery operations.

Data held in the log is used for two distinct purposes; media recovery and unit of work coordination. If a media failure occurs which affects either a CF structure or page set, the queue manager can recover the media to the point of failure by restoring a prior copy and updating this using data contained in the log.

Persistent activity performed in a unit of work is recorded on the log so that in the event of a failure, it can either be backed out or locks can be recovered on changed resources. The quantity of log data you need to retain to enable queue manager recovery is affected by these two elements.

For media recovery, you must retain sufficient log data to be able to perform media recovery from at least the most recent media copy and to be able to back out. (Your site may stipulate the ability to recover from older backups.) For unit of work integrity, you must retain the log data for your oldest in flight or indoubt units of work.

To assist you with managing the system, the queue manager detects old units of work at each log archive and reports them in messages CSQJ160 and CSQJ161. An internal task reads unit of work log information for these old units of work and rewrites it in a more succinct form to the current position in the log. Message CSQR026 indicates when this has happened. The MQSC command DISPLAY USAGE TYPE(DATASET) can also help you to manage the retention of log data. The command reports the following three pieces of recovery information:

1. How much of the log must be retained for unit of work recovery.
2. How much of the log must be retained for media recovery of page sets.
3. For a queue manager in a queue sharing group, how much of the log must be retained for media recovery of CF structures.

For each of these pieces of information, an attempt is made to map the oldest log data required into a data set. As new units of work start and stop, (1) would be expected to move to a more recent position in the log. If it is not moving, the long running UOW messages warn you that there is an issue. (2) relates to page set media recovery if the queue manager were to be shut down now and restarted. It does not know about when you last backed up your page sets, or which backup you might have to use if there was a page set failure. It normally moves to a more recent position in the log during checkpoint processing as changes held in the buffer pools are written to the page sets. In (3), the queue manager does know about CF structure backups taken either on this queue manager or on other queue managers in the queue sharing group. However, CF structure recovery requires a merge of log data from all queue managers in the queue sharing group which have interacted with the CF structure since the last backup. This means that the log data is identified by a log record sequence number, (or LRSN), which is timestamp based and so applicable across the entire queue sharing group rather than an RBA which would be different on different queue managers in the queue sharing group. It normally moves to a more recent position in the log as BACKUP CFSTRUCT commands are performed on either this or other queue managers in the queue sharing group.

Resetting the queue manager's log

Use this topic to understand how to reset the queue manager's log.

You must not allow the queue manager log RBA to wrap around from the end of the log RBA range to 0, as this leads to a queue manager outage and all persistent data will become unrecoverable. The end of the log RBA is either a value of FFFFFFFFFF (if 6-byte RBAs as in use), or FFFFFFFFFFFFFFFF (if 8-byte RBAs are in use).

The queue manager issues messages CSQI045I, CSQI046E, CSQI047E, CSQJ031D, and CSQJ032E to indicate that the used log range is significant and that you should plan to take action to avoid an unplanned outage.

The queue manager terminates with reason code 00D10257 when the RBA value reaches FFF800000000 (if 6-byte log RBAs are in use) or FFFFFFFC0000000000 (if 8-byte log RBAs are in use).

If 6-byte log RBAs are in use, consider converting the queue manager to use 8-byte log RBAs rather than resetting the queue manager's log, following the process described in [“Implementing the larger log Relative Byte Address”](#) on page 524. Converting a queue manager to use 8-byte log RBAs requires a shorter outage than resetting the log, and increases the period of time before you have to reset the log.

Message CSQJ034I, issued during queue manager initialization, indicates the end of the log RBA range for the queue manager as configured, and can be used to determine whether 6-byte or 8-byte log RBAs are in use.

The procedure to follow to reset the queue manager's log is as follows:

1. Resolve any unresolved units of work. The number of unresolved units of work is displayed at queue manager startup in message CSQR005I as the INDOUBT count. At each checkpoint, and at queue manager shutdown, the queue manager automatically issues the command **DISPLAY CONN(*) TYPE(CONN) ALL WHERE(UOWSTATE EQ UNRESOLVED)** to provide information about unresolved units of work.

See [How in-doubt units of recovery are resolved](#) for information on resolving units of recovery. The ultimate recourse is to use the **RESOLVE INDOUBT** MQSC command to manually resolve indoubt units of recovery.
2. Shut down the queue manager cleanly.

You can use either **STOP QMGR** or **STOP QMGR MODE(FORCE)** as both these commands flush any changed pages from bufferpools to the page sets.
3. If a queue manager is part of a queue sharing group, take CFSTRUCT backups on other queue managers for all structures in the queue sharing group. This ensures that the most recent backups are not in this queue manager's log, and that this queue manager's log is not required for CFSTRUCT recovery.
4. Define new logs and BSDS using CSQJU003 (see [The change log inventory utility](#) for more information on using the change log inventory utility).
5. Run **CSQUTIL RESETPAGE** against all the page sets for this queue manager (see [Copying a page and resetting the log](#) for more information on using this function). Note that page set RBAs can be reset independently, so multiple concurrent jobs (for example, one per page set) can be submitted to reduce the elapsed time for this step.
6. Restart the queue manager

Warning messages

When IBM MQ detects that the end of the log is approaching, it issues console messages in the following order, which indicate that a log reset should be planned. In this section the messages show 6-byte log RBA values. The same console messages are issued when IBM MQ is running in 8-byte log RBA mode but with different values; see [“Warning thresholds” on page 523](#) for the 8-byte log RBA thresholds.

1. When IBM MQ detects that the end of the log is approaching in the near future, (approximately 94% full) IBM MQ issues console message CSQI045I, as in the following example:

```
CSQI045I -CSQ7 CSQILCUR Log RBA has reached 0000F00000000000.  
Plan a log reset
```

2. IBM MQ issues the following CSQI046E error console message when the end of the log is near (approximately 97% full). This informs the IBM MQ administrator to take action soon.

```
CSQI046E -CSQ7 CSQILCUR Log RBA has reached 0000F80000000000.  
Perform a log reset
```

3. After the CSQI046E message is issued, at the next log switch, IBM MQ issues the following CSQJ032E console message with the word WARNING:

```
CSQJ032E -CSQ7 CSQJW307 WARNING - APPROACHING END OF  
THE LOG RBA RANGE OF 0000FFFFFFFFFFFF. CURRENT LOG RBA IS 0000F80000022000.
```

4. After the CSQI046E and CSQJ032E console messages are issued, IBM MQ issues one more error message, which does not require immediate IBM MQ administrator intervention. IBM MQ issues console message CSQI047E (when the log is approximately 99% full):

```
CSQI047E -CSQ7 CSQILCUR Log RBA has reached 0000FF0000000000.  
Stop queue manager and reset logs
```

5. When the log RBA reaches FF8000000000, IBM MQ increases the urgency of the situation and issues console message CSQJ032E with the word CRITICAL:

```
CSQJ032E -CSQ7 CSQJW009 CRITICAL - APPROACHING END OF THE LOG RBA RANGE OF 0000FFFFFFFFFFFF.
CURRENT LOG RBA IS 0000FFF7FFFFDFFF.
```

6. If the queue manager is started when the log RBA is almost at the maximum, the following CSQJ031D console message is issued. This stage requires the input of the IBM MQ administrator:.

```
CSQJ031D -CSQ7 CSQYSTRT THE LOG RBA RANGE MUST BE RESET.
REPLY 'Y' TO CONTINUE STARTUP OR 'N' TO SHUTDOWN
```

7. IBM MQ startup remains suspended until a reply is given to message CSQJ031D.

The purpose of these messages is to give the IBM MQ administrator time to plan for a system outage to reset the logs. In an ideal configuration, there are at least two queue managers, possibly in a queue sharing group (QSG), sharing the workload. When one is down for maintenance the other can continue to receive work.

The severity of console messages that IBM MQ issues becomes greater as the RBA gets closer to the end. Ideally your IBM MQ administrator should plan to reset the log RBA when the first console message is seen.

If the warning and error console messages are ignored, IBM MQ terminates with reason code 5C6-00D10257 when the log RBA reaches FFF800000000, at which point IBM MQ determines that the available range is too small for the queue manager to continue. When this point is reached, the only option is to take an outage and either reset the log or extend the size of the log RBA.

Note: When the end of the log is reached it is not possible to resolve any in-flight units of work (UOW); these are lost during the log reset process. Enough of the RBA range should be left to start the queue manager and resolve any UOW. Because IBM MQ issues console messages several times to inform that the end of the log is approaching, a log reset should be planned.

The preferred option to avoid losing any in-flight UOW is to extend the log RBA to use 8 bytes. This means that a log RBA reset will not be necessary for a long period.

Warning thresholds

The following table lists the thresholds, based on the length of the log RBA.

Console message	6-byte log RBA	8-byte log RBA
CSQI045I	0000F00000000000	FFFF800000000000
CSQI046E	0000F80000000000	FFFFC00000000000
CSQI047E	0000FF8000000000	FFFFFC0000000000
CSQJ032E	0000FF8000000000 0000FF8000000000	FFFFFC0000000000 FFFFFC0000000000
CSQJ031D	0000FF8000000000	FFFFFC0000000000

Notes:

1. For message CSQJ032E, the first number applies to the WARNING text and the second number applies to the CRITICAL text in the console message.
2. Message CSQJ031D is issued at IBM MQ initialization only.

Related concepts

[“Implementing the larger log Relative Byte Address” on page 524](#)

Before IBM MQ for z/OS 8.0, IBM MQ for z/OS used a 6 byte log RBA to identify the location of data within the log. From IBM MQ for z/OS 8.0, the log RBA can be 8 bytes long, increasing the period of time before you have to reset the log.

Implementing the larger log Relative Byte Address

Before IBM MQ for z/OS 8.0, IBM MQ for z/OS used a 6 byte log RBA to identify the location of data within the log. From IBM MQ for z/OS 8.0, the log RBA can be 8 bytes long, increasing the period of time before you have to reset the log.



Attention: You only have to carry out the following procedure to enable this feature if your queue managers were created before IBM MQ 9.3.0, as queue managers created at IBM MQ 9.3.0 and later already have this feature enabled.

See [Planning to increase the maximum addressable log range](#) for considerations when planning to enable 8 byte log RBA.

Perform these instructions, in the order shown, to enable 8 byte log RBA on a single IBM MQ for z/OS queue manager. For queue managers in a queue sharing group, perform the steps on each queue manager in turn.

1. Allocate new BSDS data sets with similar attributes to the current BSDS. You can tailor sample CSQ4BSDS and delete any irrelevant statement, or you can use your existing JCL, but change the BSDS name to something like ++HLQ++ .NEW .BSDS01.

Notes:

- a. Check the attributes of your new BSDS before submitting the job to allocate the new BSDS. The only attribute that might change is the size of the BSDS.
 - b. The new BSDS contains more data than the current BSDS, therefore, you must ensure that the new data sets are allocated with sufficient available space. The sample JCL in `thlqual.SCSQPROC(CSQ4BSDS)` contains the recommended values when defining a new BSDS.
2. Shut down the queue manager cleanly.
 3. Run the [BSDS conversion utility \(CSQJUCNV\)](#) to convert the existing BSDS to the new BSDS data sets. This usually takes a few seconds to run.

Your existing BSDS will not be changed during this process, and you can use that for the initialization of the queue manager in the case of an unsuccessful conversion.
 4. Rename the current BSDS to become the old BSDS, and the new BSDS to become the current BSDS, so that the new data sets are used when you next restart the queue manager. You can use the DFSMS Access Method Services ALTER command, for example:

```
ALTER '++HLQ++.BSDS01' NEWNAME('++HLQ++.OLD.BSDS01')
ALTER '++HLQ++.NEW.BSDS01' NEWNAME('++HLQ++.BSDS01')
```

Ensure that you also issue commands to rename both the data and index portions of the VSAM cluster.

5. Restart the queue manager. It should start in the same amount of time as it would have done when using 6 byte log RBA.

If the queue manager does not restart successfully due to a failure to access the converted BSDS, attempt to identify the cause of the failure, resolve the problem and retry the operation. If required, contact your IBM support center for assistance.

If necessary, the change can be backed out at this point by:

- a. Renaming the current BSDS to become the new BSDS.
 - b. Renaming the old BSDS to become the current BSDS.
 - c. Restarting the queue manager.
6. Once the queue manager has been successfully restarted with the converted BSDS, do not attempt to start the queue manager using the old BSDS.
 7. Message `CSQJ034I` is issued during queue manager initialization to indicate the end of the log RBA for the queue manager as configured. Confirm that the end of the log RBA range displayed is `FFFFFFFFFFFFFFFF`. This indicates that 8 byte log RBA is in use.

Related tasks

[Planning to increase the maximum addressable log range](#)

Related reference

[Larger log Relative Byte Address](#)

[The BSDS conversion utility \(CSQJUCNV\)](#)

Managing the bootstrap data set (BSDS)

The bootstrap data set (BSDS) is used to reference log data sets, and log records. Use this topic to understand how you can examine, change, and recover the BSDS.

For more information, see [The bootstrap data set](#).

This topic describes the tasks involved in managing the bootstrap data set. It contains these sections:

- [“Finding out what the BSDS contains” on page 525](#)
- [“Changing the BSDS” on page 526](#)
- [“Recovering the BSDS” on page 530](#)

Finding out what the BSDS contains

You can use the print log map utility (CSQJU004) to examine the contents of the BSDS.

The print log map utility (CSQJU004) is a batch utility that lists the information stored in the BSDS. For instructions on running it, see [The print log map utility](#).

The BSDS contains:

- [Time stamps](#)
- [Active log data set status](#)

Time stamps in the BSDS

The output of the print log map utility shows the time stamps, which are used to record the date and time of various system events, that are stored in the BSDS.

The following time stamps are included in the header section of the report:

SYSTEM TIMESTAMP

Reflects the date and time the BSDS was last updated. The BSDS time stamp can be updated when:

- The queue manager starts.
- The write threshold is reached during log write activities. Depending on the number of output buffers you have specified and the system activity rate, the BSDS might be updated several times a second, or might not be updated for several seconds, minutes, or even hours. For details of the write threshold, see the WRTHRS parameter of the CSQ6LOGP macro in [Using CSQ6LOGP](#).
- IBM MQ drops into a single BSDS mode from its normal dual BSDS mode due to an error. This can occur when a request to get, insert, point to, update, or delete a BSDS record is unsuccessful. When this error occurs, IBM MQ updates the time stamp in the remaining BSDS to force a time stamp mismatch with the disabled BSDS.

UTILITY TIMESTAMP

The date and time the contents of the BSDS were altered by the change log inventory utility (CSQJU003).

The following time stamps are included in the active and archive log data sets portion of the report:

Active log date

The date the active log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done.

Active log time

The time the active log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done.

Archive log date

The date the archive log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done or the archive itself was done.

Archive log time

The time the archive log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done or the archive itself was done.

Active log data set status

The BSDS records the status of an active log data set as one of the following:

NEW

The data set has been defined but never used by IBM MQ, or the log was truncated to a point before the data set was first used. In either case, the data set starting and ending RBA values are reset to zero.

REUSABLE

Either the data set has been defined but never used by IBM MQ, or the data set has been offloaded. In the print log map output, the start RBA value for the last REUSABLE data set is equal to the start RBA value of the last archive log data set.

NOT REUSABLE

The data set contains records that have not been offloaded.

STOPPED

The offload processor encountered an error while reading a record, and that record could not be obtained from the other copy of the active log.

TRUNCATED

Either:

- An I/O error occurred, and IBM MQ has stopped writing to this data set. The active log data set is offloaded, beginning with the starting RBA and continuing up to the last valid record segment in the truncated active log data set. The RBA of the last valid record segment is lower than the ending RBA of the active log data set. Logging is switched to the next available active log data set, and continues uninterrupted.

or

- An ARCHIVE LOG function has been called, which has truncated the active log.

The status appears in the output from the print log map utility.

 **Changing the BSDS**

You do not have to take special steps to keep the BSDS updated with records of logging events because IBM MQ does that automatically.

However, you might want to change the BSDS if you do any of the following:

- Add more active log data sets.
- Copy active log data sets to newly allocated data sets, for example, when providing larger active log allocations.
- Move log data sets to other devices.
- Recover a damaged BSDS.
- Discard outdated archive log data sets.

You can change the BSDS by running the change log inventory utility (CSQJU003). Only run this utility when the queue manager is inactive, or you might get inconsistent results. The action of the utility is controlled by statements in the SYSIN data set. This section shows several examples. For complete instructions, see [The change log inventory utility](#).

You can copy an active log data set only when the queue manager is inactive because IBM MQ allocates the active log data sets as exclusive (DISP=OLD) at queue manager startup.

Changes for active logs

Use this topic to understand how you can change the active logs using the BSDS.

You can add to, delete from, and record entries in the BSDS for active logs using the change log utility. Examples only are shown here; replace the data set names shown with the ones you want to use. For more details of the utility, see [The change log inventory utility](#).

See these sections for more information:

- [Adding record entries to the BSDS](#)
- [Deleting information about the active log data set from the BSDS](#)
- [Recording information about the log data set in the BSDS](#)
- [Increasing the size of the active log](#)
- [The use of CSQJUFMT](#)

Adding record entries to the BSDS

If an active log has been flagged as "stopped", it is not reused for logging; however, it continues to be used for reading. Use the access method services to define new active log data sets, then use the change log inventory utility to register the new data sets in the BSDS. For example, use:

```
NEWLOG DSN=MQM111.LOGCOPY1.DS10,COPY1
NEWLOG DSN=MQM111.LOGCOPY2.DS10,COPY2
```

If you are copying the contents of an old active log data set to the new one, you can also give the RBA range and the starting and ending time stamps on the NEWLOG function.

Deleting information about the active log data set from the BSDS

To delete information about an active log data set from the BSDS, you could use:

```
DELETE DSN=MQM111.LOGCOPY1.DS99
DELETE DSN=MQM111.LOGCOPY2.DS99
```

Recording information about the log data set in the BSDS

To record information about an existing active log data set in the BSDS, use:

```
NEWLOG DSN=MQM111.LOGCOPY1.DS10,COPY2,STARTIME=19930212205198,
ENDTIME=19930412205200,STARTRBA=6400,ENDRBA=94FF
```

You might need to insert a record containing this type of information in the BSDS because:

- The entry for the data set has been deleted, but is needed again.
- You are copying the contents of one active log data set to another data set.
- You are recovering the BSDS from a backup copy.

Increasing the size of the active log

There are two methods of achieving this process.

1. When the queue manager is active:
 - a. Define new larger log data sets using JCL.
 - b. Add the new log data sets to the active queue manager using the MQSC DEFINE LOG command.
 - c. Use the MQSC ARCHIVE LOG command to move the current active log, to be a new larger log.
 - d. Wait for the archive of the smaller active log data set to complete.
 - e. Shut down the queue manager, using the CSQJU003 utility to remove the old small active logs.
 - f. Restart the queue manager.
2. When the queue manager is inactive:
 - a. Stop the queue manager. This step is required because IBM MQ allocates all active log data sets for its exclusive use when it is active.
 - b. Use Access Method Services ALTER with the NEWNAME option to rename your active log data sets.
 - c. Use Access Method Services DEFINE to define larger active log data sets.

By reusing the old data set names, you do not have to run the change log inventory utility to establish new names in the BSDSs. The old data set names and the correct RBA ranges are already in the BSDSs.
 - d. Use Access Method Services REPRO to copy the old (renamed) data sets into their appropriate new data sets.

Note: This step can take a long time, so your enterprise could be out of action for this period.
 - e. Start the queue manager.

If all your log data sets are the same size, your system will be operationally more consistent and efficient. If the log data sets are not the same size, it is more difficult to track your system's logs, and so space can be wasted.

The use of CSQJUFMT

Do not run a CSQJUFMT format when increasing the size of an active log.

If you run CSQJUFMT (in order to provide a performance advantage the first time the queue manager writes to the new active log) you receive messages:

```
IEC070I 203-204,XS95GTLX,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
```

In addition, if you use the Access Method Services REPRO, ensure that you define a new empty log.

If you use REPRO to copy the old (renamed) data set into its respective new data set, the default is NOREPLACE.

This means that REPRO does not replace a record that is already on the designated data set. When formatting is done on the data set, the RBA value is reset. The net result is a data set that is not empty after formatting.

Changes for archive logs

Use this topic to understand how to change the archive logs.

You can add to, delete from, and change the password of, entries in the BSDS for archive logs. Examples only are shown here; replace the data set names shown with the ones you want to use. For more details of the utility, see [The change log inventory utility](#).

- [Adding an archive log](#)
- [Deleting an archive log](#)
- [Changing the password of an archive log](#)

Adding an archive log

When the recovery of an object depends on reading an existing archive log data set, the BSDS must contain information about that data set so that IBM MQ can find it. To register information about an existing archive log data set in the BSDS, use:

```
NEWLOG DSNAME=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04,  
UNIT=TAPE,STARTRBA=3A190000,ENDRBA=3A1F0FFF,CATALOG=NO
```

Deleting an archive log

To delete an entire archive log data set on one or more volumes, use:

```
DELETE DSNAME=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04
```

Changing the password of an archive log

If you change the password of an existing archive log data set, you must also change the information in the BSDS.

1. List the BSDS, using the print log map utility.
2. Delete the entry for the archive log data set with the changed password, using the DELETE function of the CSQJU003 utility (see topic [The change log inventory utility](#)).
3. Name the data set as for a new archive log data set. Use the NEWLOG function of the CSQJU003 utility (see topic [The change log inventory utility](#)), and give the new password, the starting and ending RBAs, and the volume serial numbers (which can be found in the print log map utility output, see [The print log map utility](#)).

To change the password for new archive log data sets, use:

```
ARCHIVE PASSWORD= password
```

To stop placing passwords on new archive log data sets, use:

```
ARCHIVE NOPASSWD
```

Note: Only use the ARCHIVE utility function if you do not have an external security manager.

 **Changing the high-level qualifier (HLQ) for the logs and BSDS**

Use this topic to understand the procedure required to change the high-level qualifier (HLQ).

Before you begin

You must end the queue manager normally before copying any of the logs or data sets to the new data sets. This is to ensure that the data is consistent and no recovery is needed during restart.

About this task

This task provides information about how to change the HLQ for the logs and BSDS. To do this, follow these steps:

Procedure

1. Run the log print utility CSQJU004 to record the log data set information. This information is needed later.
2. You can either:
 - a) run DSS backup and restore with rename on the log and BSDS data sets to be renamed, or
 - b) use AMS DEFINE and REPRO to create the HLQ data sets and copy the data from the old data sets.
3. Modify the MSTR and CHIN procedures to point to the new data sets.
4. Delete the old log information in the new copy of the BSDS using CSQJU003.
5. Define the new log data sets to the new BSDS using the NEWLOG function of CSQJU003.
Keep all information about each log the same, apart from the HLQ.
6. The new BSDS should reflect the same information that was recorded for the old logs in the old BSDS.
The HLQ should be the only thing that has changed.

What to do next

Compare the CSQJU004 output for the old and new BSDS to ensure that they look EXACTLY the same (except for the HLQs) before starting the queue manager.

Note: Care must be taken when performing these operations. Incorrect actions might lead to unrecoverable situations. Check the PRINT LOG MAP UTILITY output and make sure that all the information needed for recovery or restart has been included.

Recovering the BSDS

If IBM MQ is operating in dual BSDS mode and one BSDS becomes damaged, forcing IBM MQ into single BSDS mode, IBM MQ continues to operate without a problem (until the next restart).

To return the environment to dual BSDS mode:

1. Use Access Method Services to rename or delete the damaged BSDS and to define a new BSDS with the same name as the damaged BSDS. Example control statements can be found in job CSQ4BREC in thlqual.SCSQPROC.
2. Issue the IBM MQ command RECOVER BSDS to make a copy of the valid BSDS in the newly allocated data set and to reinstate dual BSDS mode.

If IBM MQ is operating in single BSDS mode and the BSDS is damaged, or if IBM MQ is operating in dual BSDS mode and both BSDSs are damaged, the queue manager stops and does not restart until the BSDS data sets are repaired. In this case:

1. Locate the BSDS associated with the most recent archive log data set. The data set name of the most recent archive log appears on the job log in the last occurrence of message CSQJ003I, which indicates that offload processing has been completed successfully. In preparation for the rest of this procedure, it is a good practice to keep a log of all successful archives noted by that message:
 - If archive logs are on DASD, the BSDS is allocated on any available DASD. The BSDS name is like the corresponding archive log data set name; change only the first letter of the last qualifier, from A to B, as in this example:

Archive log nameCSQ.ARCHLOG1. **A** 0000001**BSDS copy name**CSQ.ARCHLOG1. **B** 0000001

- If archive logs are on tape, the BSDS is the first data set of the first archive log volume. The BSDS is not repeated on later volumes.
- 2. If the most recent archive log data set has no copy of the BSDS (for example, because an error occurred when offloading it), locate an earlier copy of the BSDS from earlier offload processing.
- 3. Rename *damaged* BSDSs using the Access Method Services ALTER command with the NEWNAME option. If you want to delete a damaged BSDS, use the Access Method Services DELETE command. For each damaged BSDS, use Access Method Services to define a new BSDS as a replacement data set. Job CSQ4BREC in thlqual.SCSQPROC contains Access Method Services control statements to define a new BSDS.
- 4. Use the Access Method Services REPRO command to copy the BSDS from the archive log to one of the replacement BSDSs you defined in step “3” on page 531. Do not copy any data to the second replacement BSDS, you do that in step “5” on page 532.

- a. Print the contents of the replacement BSDS.

Use the print log map utility (CSQJU004) to print the contents of the replacement BSDS. This enables you to review the contents of the replacement BSDS before continuing your recovery work.

- b. Update the archive log data set inventory in the replacement BSDS.

Examine the output from the print log map utility and check that the replacement BSDS does not contain a record of the archive log from which the BSDS was copied. If the replacement BSDS is an old copy, its inventory might not contain all archive log data sets that were created more recently. The BSDS inventory of the archive log data sets must be updated to reflect the current subsystem inventory.

Use the change log inventory utility (CSQJU003) NEWLOG statement to update the replacement BSDS, adding a record of the archive log from which the BSDS was copied. If the archive log data set is password-protected, use the PASSWORD option of the NEWLOG function. Also, if the archive log data set is cataloged, ensure that the CATALOG option of the NEWLOG function is properly set to CATALOG=YES. Use the NEWLOG statement to add any additional archive log data sets that were created later than the BSDS copy.

- c. Update passwords in the replacement BSDS.

The BSDS contains passwords for the archive log data sets and for the active log data sets. To ensure that the passwords in the replacement BSDS reflect the current passwords used by your installation, use the change log inventory ARCHIVE utility function with the PASSWORD option.

- d. Update the active log data set inventory in the replacement BSDS.

In unusual circumstances, your installation might have added, deleted, or renamed active log data sets since the BSDS was copied. In this case, the replacement BSDS does not reflect the actual number or names of the active log data sets your installation currently has in use.

If you need to delete an active log data set from the replacement BSDS log inventory, use the change log inventory utility DELETE function.

If you need to add an active log data set to the replacement BSDS log inventory, use the change log inventory utility NEWLOG function. Ensure that the RBA range is specified correctly on the NEWLOG function. If the active log data set is password-protected, use the PASSWORD option.

If you need to rename an active log data set in the replacement BSDS log inventory, use the change log inventory utility DELETE function, followed by the NEWLOG function. Ensure that the RBA range is specified correctly on the NEWLOG function. If the active log data set is password-protected, use the PASSWORD option.

- e. Update the active log RBA ranges in the replacement BSDS.

Later, when the queue manager restarts, it compares the RBAs of the active log data sets listed in the BSDS with the RBAs found in the actual active log data sets. If the RBAs do not agree, the queue manager does not restart. The problem is magnified when an old copy of the BSDS is used. To solve this problem, use the change log inventory utility (CSQJU003) to adjust the RBAs found in the BSDS using the RBAs in the actual active log data sets. You do this by:

- Using the print log records utility (CSQ1LOGP) to print a summary report of the active log data set. This shows the starting and ending RBAs.
- Comparing the actual RBA ranges with the RBA ranges you have just printed, when the RBAs of all active log data sets are known.

If the RBA ranges are equal for all active log data sets, you can proceed to the next recovery step without any additional work.

If the RBA ranges are not equal, adjust the values in the BSDS to reflect the actual values. For each active log data set that needs to have the RBA range adjusted, use the change log inventory utility DELETE function to delete the active log data set from the inventory in the replacement BSDS. Then use the NEWLOG function to redefine the active log data set to the BSDS. If the active log data sets are password-protected, use the PASSWORD option of the NEWLOG function.

- f. If only two active log data sets are specified for each copy of the active log, IBM MQ can have difficulty during queue manager restart. The problem can arise when one of the active log data sets is full and has not been offloaded, while the second active log data set is close to filling. In this case, add a new active log data set for each copy of the active log and define each new active log data set in the replacement BSDS log inventory.

Use the Access Method Services DEFINE command to define a new active log data set for each copy of the active log and use the change log inventory utility NEWLOG function to define the new active log data sets in the replacement BSDS. You do not need to specify the RBA ranges on the NEWLOG statement. However, if the active log data sets are password-protected, use the PASSWORD option of the NEWLOG function. Example control statements to accomplish this task can be found in job CSQ4LREC in thlqual.SCSQPROC.

5. Copy the updated BSDS to the second new BSDS data set. The BSDSs are now identical.

Use the print log map utility (CSQJU004) to print the contents of the second replacement BSDS at this point.

6. See [Active log problems](#) for information about what to do if you have lost your current active log data set.
7. Restart the queue manager using the newly constructed BSDS. IBM MQ determines the current RBA and what active logs need to be archived.

Managing page sets

Use this topic to understand how to manage the page sets associated with a queue manager.

This topic describes how to add, copy, and generally manage the page sets associated with a queue manager. It contains these sections:

- [“How to change the high-level qualifier \(HLQ\) for the page sets” on page 533](#)
- [“How to add a page set to a queue manager” on page 533](#)
- [“What to do when one of your page sets becomes full” on page 533](#)
- [“How to balance loads on page sets” on page 534](#)
- [How to increase the size of a page set](#)
- [“How to reduce a page set” on page 537](#)
- [“How to reintroduce a page set” on page 538](#)

- [“How to back up and recover page sets” on page 539](#)
- [“How to delete page sets” on page 542](#)
- [“How to back up and restore queues using CSQUTIL” on page 542](#)

See [Page sets](#) for a description of page sets, storage classes, buffers, and buffer pools, and some of the performance considerations that apply.

How to change the high-level qualifier (HLQ) for the page sets

This task gives information on how to change the HLQ for the page sets. To perform this task, do the following:

1. Define the new HLQ page sets.
2. If the size allocation is the same as the old page sets, copy the existing page set using REPRO to the empty new HLQ page sets.
3. If you are increasing the size of the page sets, use the FORMAT function of CSQUTIL to format the destination pages, and then the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set.

For more information, see [Formatting page sets \(FORMAT\)](#), and [Expanding a page set \(COPYPAGE\)](#).

4. Change the CSQP00xx DD statement in the queue manager procedure to point to the new HLQ page sets.

Restart the queue manager and verify the changes to the page sets.

How to add a page set to a queue manager

This description assumes that you have a queue manager that is already running. You might need to add a page set if, for example, your queue manager has to cope with new applications using new queues.

To add a new page set, use the following procedure:

1. Define and format the new page set. You can use the sample JCL in thlqual.SCSQPROC(CSQ4PAGE) as a basis. For more information, see [Formatting page sets \(FORMAT\)](#).

Take care not to format any page sets that are in use, unless this is what you intend. If so, use the FORCE option of the FORMAT utility function.

2. Use the DEFINE PSID command with the DSN option to associate the page set with a buffer pool.
3. Add the appropriate storage class definitions for your page set by issuing DEFINE STGCLASS commands.
4. Optionally, to document how your queue manager is configured:
 - a. Add the new page set to the started task procedure for your queue manager.
 - b. Add a definition for the new page set to your CSQINP1 initialization data set.
 - c. Add a definition for the new storage class to your CSQ4INYP initialization data set member.

For details of the DEFINE PSID and DEFINE STGCLASS commands, see [DEFINE PSID](#) and [DEFINE STGCLASS](#).

What to do when one of your page sets becomes full

You can find out about the utilization of page sets by using the IBM MQ command DISPLAY USAGE. For example, the command:

```
DISPLAY USAGE PSID(03)
```

displays the current state of the page set 03. This tells you how many free pages this page set has.

If you have defined secondary extents for your page sets, they are dynamically expanded each time they fill up. Eventually, all secondary extents are used, or no further disk space is available. If this happens, an application receives the return code MQRC_STORAGE_MEDIUM_FULL.

If an application receives a return code of MQRC_STORAGE_MEDIUM_FULL from an MQI call, this is a clear indication that there is not enough space remaining on the page set. If the problem persists or is likely to recur, you must do something to solve it.

You can approach this problem in a number of ways:

- Balance the load between page sets by moving queues from one page set to another.
- Expand the page set. See [“How to increase the size of a page set”](#) on page 536 for instructions.
- Redefine the page set so that it can expand beyond 4 GB to a maximum size of 64 GB. See [Defining a page set to be larger than 4 GB](#) for instructions.

How to balance loads on page sets

Load balancing on page sets means moving the messages associated with one or more queues from one page set to another, less used, page set. Use this technique if it is not practical to expand the page set.

To identify which queues are using a page set, use the appropriate IBM MQ commands. For example, to find out which queues are mapped to page set 02, first, find out which storage classes map to page set 02, by using the command:

```
DISPLAY STGCLASS(*) PSID(02)
```

Then use the following command to find out which queues use which storage class:

```
DISPLAY QUEUE(*) TYPE(QLOCAL) STGCLASS
```

Moving a non-shared queue

To move queues and their messages from one page set to another, use the MQSC MOVE QLOCAL command (described in MOVE QLOCAL). When you have identified the queue or queues that you want to move to a new page set, follow this procedure for each of these queues:

1. Ensure that the queue you want to move is not in use by any applications (that is, IPPROCS and OPPOCS values from the DISPLAY QSTATUS command are zero) and that it has no uncommitted messages (the UNCOM value from the DISPLAY QSTATUS command is NO).

Note: The only way to ensure that this state continues is to change the security authorization of the queue temporarily. See [Profiles for queue security](#) for more information.

If you cannot do this, later stages in this procedure might fail if applications start to use the queue despite precautionary steps such as setting PUT(DISABLED). However, messages can never be lost by this procedure.

2. Prevent applications from putting messages on the queue being moved by altering the queue definition to disable MQPUT s. Change the queue definition to PUT(DISABLED).
3. Define a temporary queue with the same attributes as the queue that is being moved, using the command:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
```

Note: If this temporary queue already exists from a previous run, delete it before doing the define.

4. Move the messages to the temporary queue using the following command:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Delete the queue you are moving, using the command:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Define a new storage class that maps to the required page set, for example:

```
DEFINE STGCLASS(NEW) PSID(nn)
```

Add the new storage class definition to the CSQINP2 data sets ready for the next queue manager restart.

7. Redefine the queue that you are moving, by changing the storage class attribute:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
```

When the queue is redefined, it is based on the temporary queue created in step “3” on page 534.

8. Move the messages back to the new queue, using the command:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

9. The queue created in step “3” on page 534 is no longer required. Use the following command to delete it:

```
DELETE QL(TEMP_QUEUE)
```

10. If the queue being moved was defined in the CSQINP2 data sets, change the STGCLASS attribute of the appropriate DEFINE QLOCAL command in the CSQINP2 data sets. Add the REPLACE keyword so that the existing queue definition is replaced.

Figure 29 on page 536 shows an extract from a load balancing job.

```

//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)
/*

```

Figure 29. Extract from a load balancing job for a page set

How to increase the size of a page set

You can initially allocate a page set larger than 4 GB, See [Defining a page set to be larger than 4 GB](#)

A page set can be defined to be automatically expanded as it becomes full by specifying EXPAND(SYSTEM) or EXPAND(USER). If your page set was defined with EXPAND(NONE), you can expand it in either of two ways:

- Alter its definition to allow automatic expansion. See [Altering a page set to allow automatic expansion](#)
- Create a new, larger page set and copy the messages from the old page set to the new one. See [Moving messages to a new, larger page set](#)

Defining a page set to be larger than 4 GB

IBM MQ can use a page set up to 64 GB in size, provided the data set is defined with 'extended addressability' to VSAM. Extended addressability is an attribute which is conferred by an SMS data class.

Note: Page sets and active log data sets are eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV) and, from z/OS V1.12, an archive log dataset can also reside in the EAS.

In the example shown in the following sample JCL, the management class 'EXTENDED' is defined to SMS with 'Extended addressability'. If your existing page set is not currently defined as having extended addressability, use the following method to migrate to an extended addressability format data set.

1. Stop the queue manager.
2. Use Access Method Services to rename the existing page set.
3. Define a destination page set, the same size as the existing page set, but with DATACLAS(EXTENDED).

Note: Extended-format data sets must be SMS managed. These are the mechanisms for requesting extended format for VSAM data sets:

- Using a data class that has a DSNTYPE value of EXT and the subparameter R or P to indicate required or preferred.
- Coding DSNTYPE=EXTREQ (extended format is required) or DSNTYPE=EXTPREF (extended format is preferred) on the DD statement.

- Coding the LIKE= parameter on the DD statement to refer to an existing extended format data set.

For more information, see [Restrictions on Defining Extended-Format Data Sets](#).

4. Use the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set. See [Expanding a page set \(COPYPAGE\)](#) for more details.
5. Restart the queue manager.
6. Alter the page set to use system expansion, to allow it to continue growing beyond its current allocation.

The following JCL shows example Access Method Services commands:

```
//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))
/*
```

Altering a page set to allow automatic expansion

Use the ALTER PSID command with the EXPAND(USER) or EXPAND(SYSTEM) options. See [ALTER PSID](#) and [Expanding a page set \(COPYPAGE\)](#) for general information on expanding page sets.

Moving messages to a new, larger page set

This technique involves stopping and restarting the queue manager. This deletes any nonpersistent messages that are not on shared queues at restart time. If you have nonpersistent messages that you do not want to be deleted, use load balancing instead. For more details, see [“How to balance loads on page sets” on page 534](#). In this description, the page set that you want to expand is referred to as the *source* page set; the new, larger page set is referred to as the *destination* page set.

Follow these steps:

1. Stop the queue manager.
2. Define the destination page set, ensuring that it is larger than the source page set, with a larger secondary extent value.
3. Use the FORMAT function of CSQUTIL to format the destination page set. See [Formatting page sets \(FORMAT\)](#) for more details.
4. Use the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set. See [Expanding a page set \(COPYPAGE\)](#) for more details.
5. Restart the queue manager using the destination page set by doing one of the following:
 - Change the queue manager started task procedure to reference the destination page set.
 - Use Access Method Services to delete the source page set and then rename the destination page set, giving it the same name as that of the source page set.

Attention:

Before you delete any IBM MQ page set, be sure that you have made the required backup copies.

How to reduce a page set

Prevent all users, other than the IBM MQ administrator, from using the queue manager. For example; by changing the access security settings.

If you have a large page set that is mostly empty (as shown by the DISPLAY USAGE command), you might want to reduce its size. The procedure to do this involves using the COPY, FORMAT, and LOAD functions of CSQUTIL (see [IBM MQ utility program](#)). This procedure does not work for page set zero (0), as it is not practical to reduce the size of this page set; the only way to do so is by reinitializing your queue manager (see [“Reinitializing a queue manager” on page 560](#)). The prerequisite of this procedure is to try and remove all users from the system so that all UOWs are complete and the page sets are consistent.

1. Use the STOP QMGR command with the QUIESCE or FORCE attribute to stop the queue manager.
2. Run the SCOPY function of CSQUTIL with the PSID option, to copy all message data from the large page set and save them in a sequential data set.
3. Define a new smaller page set data set to replace the large page set.
4. Run the FORMAT TYPE(NEW) function of CSQUTIL against the page set that you created in step [“3” on page 538](#).
5. Restart the queue manager using the page set created in step [“3” on page 538](#).
6. Run the LOAD function of CSQUTIL to load back all the messages saved during step [“2” on page 538](#).
7. Allow all users access to the queue manager.
8. Delete the old large page set.

How to reintroduce a page set

In certain scenarios it is useful to be able to bring an old page set online again to the queue manager. Unless specific action is taken, when the old page set is brought online the queue manager will recognize that the page set recovery RBA stored in the page set itself and in the checkpoint records is old, and will therefore automatically start media recovery of the page set to bring it up to date.

Such media recovery can only be performed at queue manager restart, and is likely to take a considerable length of time, especially if archive logs held on tape must be read. However, normally in this circumstance, the page set has been offline for the intervening period and so the log contains no information pertinent to the page set recovery.

The following three choices are available:

Allow full media recovery to be performed.

1. Stop the queue manager.
2. Ensure definitions are available for the page set in both the started task procedure for the queue manager and in the CSQINP1 initialization data set.
3. Restart the queue manager.

Allow any messages on the page set to be destroyed.

This choice is useful where a page set has been offline for a long time (some months, for example) and it has now been decided to reuse it for a different purpose.

1. Format the page set using the FORMAT function of CSQUTIL with the TYPE(NEW) option.
2. Add definitions for the page set to both the started task procedure for the queue manager and the CSQINP1 initialization data set.
3. Restart the queue manager.

Using the TYPE(NEW) option for formatting clears the current contents of the page set and tells the queue manager to ignore any historical information in the checkpoint about the page set.

Bring the page set online avoiding the media recovery process.

Use this technique only if you are sure that the page set has been offline since a clean shutdown of the queue manager. This choice is most appropriate where the page set has been offline for a short period, typically due to operational issues such as a backup running while the queue manager is being started.

1. Format the page set using the FORMAT function of CSQUTIL with the TYPE(REPLACE) option.

2. Either add the page set back into the queue manager dynamically using the DEFINE PSID command with the DSN option or allow it to be added at a queue manager restart.

Using the TYPE(REPLACE) option for formatting checks that the page set was cleanly closed by the queue manager, and marks it so that media recovery will not be performed. No other changes are made to the contents of the page set.

How to back up and recover page sets

There are different mechanisms available for back up and recovery. Use this topic to understand these mechanisms.

This section describes the following topics:

- [“Creating a point of recovery for non-shared resources” on page 539](#)
- [“Backing up page sets” on page 540](#)
- [“Recovering page sets” on page 541](#)
- [How to delete page sets](#)

For information about how to create a point of recovery for shared resources, see [“Recovering shared queues” on page 547](#).

Creating a point of recovery for non-shared resources

IBM MQ can recover objects and non-shared persistent messages to their current state if both:

1. Copies of page sets from an earlier point exist.
2. All the IBM MQ logs are available to perform recovery from that point.

These represent a point of recovery for non-shared resources.

Both objects and messages are held on page sets. Multiple objects and messages from different queues can exist on the same page set. For recovery purposes, objects and messages cannot be backed up in isolation, so a page set must be backed up as a whole to ensure the correct recovery of the data.

The IBM MQ recovery log contains a record of all persistent messages and changes made to objects. If IBM MQ fails (for example, due to an I/O error on a page set), you can recover the page set by restoring the backup copy and restarting the queue manager. IBM MQ applies the log changes to the page set from the point of the backup copy.

There are two ways of creating a point of recovery:

Full backup

Stop the queue manager, which forces all updates on to the page sets.

This allows you to restart from the point of recovery, using only the backed up page set data sets and the logs from that point on.

Fuzzy backup

Take *fuzzy* backup copies of the page sets without stopping the queue manager.

If you use this method, and your associated logs later become damaged or lost, you cannot use the fuzzy page set backup copies to recover. This is because the fuzzy page set backup copies contain an inconsistent view of the state of the queue manager and are dependent on the logs being available. If the logs are not available, you need to return to the last set of backup page set copies taken while the subsystem was inactive ([Method 1](#)) and accept the loss of data from that time.

Method 1: Full backup

This method involves shutting the queue manager down. This forces all updates on to the page sets so that the page sets are in a consistent state.

1. Stop all the IBM MQ applications that are using the queue manager (allowing them to complete first). This can be done by changing the access security or queue settings, for example.
2. When all activity has completed, display and resolve any in-doubt units of recovery. (Use the commands `DISPLAY CONN` and `RESOLVE INDOUBT`, as described in [DISPLAY CONN](#) and [RESOLVE INDOUBT](#).)
This brings the page sets to a consistent state; if you do not do this, your page sets might not be consistent, and you are effectively doing a fuzzy backup.
3. Issue the `ARCHIVE LOG` command to ensure that the latest log data is written out to the log data sets.
4. Issue the `STOP QMGR MODE(QUIESCE)` command. Record the lowest RBA value in the `CSQI024I` or `CSQI025I` messages (see [CSQI024I](#) and [CSQI025I](#) for more information). You should keep the log data sets starting from the one indicated by the RBA value up to the current log data set.
5. Take backup copies of all the queue manager page sets (see [“Backing up page sets” on page 540](#)).

Method 2: Fuzzy backup

This method does not involve shutting the queue manager down. Therefore, updates might be in virtual storage buffers during the backup process. This means that the page sets are not in a consistent state, and can only be used for recovery with the logs.

1. Issue the `DISPLAY USAGE TYPE(ALL)` command, and record the RBA value in the `CSQI024I` or `CSQI025I` messages (see [CSQI024I](#) and [CSQI025I](#) for more information).
2. Take backup copies of the page sets (see [“Backing up page sets” on page 540](#)).
3. Issue the `ARCHIVE LOG` command, to ensure that the latest log data is written out to the log data sets. To restart from the point of recovery, you must keep the log data sets starting from the log data set indicated by the RBA value up to the current log data set.

Backing up page sets

To recover a page set, IBM MQ needs to know how far back in the log to go. IBM MQ maintains a log RBA number in page zero of each page set, called the *recovery log sequence number* (LSN). This number is the starting RBA in the log from which IBM MQ can recover the page set. When you back up a page set, this number is also copied.

If the copy is later used to recover the page set, IBM MQ must have access to all the log records from this RBA value to the current RBA. That means you must keep enough of the log records to enable IBM MQ to recover from the oldest backup copy of a page set you intend to keep.

Use `ADRDSSU COPY` function to copy the page sets.

For more information, see the [COPY DATASET Command Syntax for Logical Data Set](#) documentation .

For example:

```
//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 -
FASTRPLICATION(PREF ) -
CANCELERROR -
TOL(ENQF)
/*
//
```

If you copy the page set while the queue manager is running you must use a copy utility that copies page zero of the page set first. If you do not do this you could corrupt the data in your page set.

If the process of dynamically expanding a page set is interrupted, for example by power to the system being lost, you can still use ADRDSSU to take a backup of a page set.

If you perform an Access Method Services IDCAMS LISTCAT ENT('page set data set name') ALLOC, you will see that the HI-ALLOC-RBA is higher than the HI-USED-RBA.

The next time this page set fills up it is extended again, if possible, and the pages between the high used RBA and the highest allocated RBA are used, along with another new extent.

Backing up your object definitions

You should also back up copies of your object definitions. To do this, use the MAKEDEF feature of the CSQUTIL COMMAND function (described in [Issuing commands to IBM MQ \(COMMAND\)](#)).

Back up your object definitions whenever you take a backup copy of your queue manager, and keep the most current version.

Recovering page sets

If the queue manager has terminated due to a failure, the queue manager can normally be restarted with all recovery being performed during restart. However, such recovery is not possible if any of your page sets or log data sets are not available. The extent to which you can now recover depends on the availability of backup copies of page sets and log data sets.

To restart from a point of recovery you must have:

- A backup copy of the page set that is to be recovered.
- If you used the "fuzzy" backup process described in ["Method 2: Fuzzy backup"](#) on page 540, the log data set that included the recorded RBA value, the log data set that was made by the ARCHIVE LOG command, and all the log data sets between these.
- If you used full backup, but you do not have the log data sets following that made by the ARCHIVE LOG command, you do **not** need to run the FORMAT TYPE(REPLACE) function of the CSQUTIL utility against all your page sets.

To recover a page set to its current state, you must also have all the log data sets and records since the ARCHIVE LOG command.

There are two methods for recovering a page set. To use either method, the queue manager must be stopped.

Simple recovery

This is the simpler method, and is appropriate for most recovery situations.

1. Delete the page set you want to restore from backup.
2. Use the ADRDSSU COPY function to recover your page set from the backup copy..

Alternatively, you can rename your backup copy to the original name, or change the CSQP00xx DD statement in your queue manager procedure to point to your backup page set. However, if you then lose or corrupt the page set, you will no longer have a backup copy to restore from.

3. Restart the queue manager.
4. When the queue manager has restarted successfully, you can restart your applications
5. Reinstate your normal backup procedures for the restored page.

Advanced recovery

This method provides performance advantages if you have a large page set to recover, or if there has been much activity on the page set since the last backup copy was taken. However, it requires more manual intervention than the simple method, which might increase the risk of error and the time taken to perform the recovery.

1. Delete and redefine the page set you want to restore from backup.
2. Use ADRDSSU to copy the backup copy of the page set into the new page set. Define your new page set with a secondary extent value so that it can be expanded dynamically.

Alternatively, you can rename your backup copy to the original name, or change the CSQP00xx DD statement in your queue manager procedure to point to your backup page set. However, if you then lose or corrupt the page set, you will no longer have a backup copy to restore from.

3. Change the CSQINP1 definitions for your queue manager to make the buffer pool associated with the page set being recovered as large as possible. By making the buffer pool large, you might be able to keep all the changed pages resident in the buffer pool and reduce the amount of I/O to the page set.
4. Restart the queue manager.
5. When the queue manager has restarted successfully, stop it (using quiesce) and then restart it using the normal buffer pool definition for that page set. After this second restart completes successfully, you can restart your applications
6. Reinstate your normal backup procedures for the restored page.

What happens when the queue manager is restarted

When the queue manager is restarted, it applies all changes made to the page set that are registered in the log, beginning at the restart point for the page set. IBM MQ can recover multiple page sets in this way. The page set is dynamically expanded, if required, during media recovery.

During restart, IBM MQ determines the log RBA to start from by taking the lowest value from the following:

- Recovery LSN from the checkpoint log record for each page set.
- Recovery LSN from page zero in each page set.
- The RBA of the oldest incomplete unit of recovery in the system at the time the backup was taken.

All object definitions are stored on page set zero. Messages can be stored on any available page set.

Note: The queue manager cannot restart if page set zero is not available.

How to delete page sets

You delete a page set by using the DELETE PSID command; see [DELETE PSID](#) for details of this command.

You cannot delete a page set that is still referenced by any storage class. Use DISPLAY STGCLASS to find out which storage classes reference a page set.

The data set is deallocated from IBM MQ but is not deleted. It remains available for future use, or can be deleted using z/OS facilities.

Remove the page set from the started task procedure for your queue manager.

Remove the definition of the page set from your CSQINP1 initialization data set.

How to back up and restore queues using CSQUTIL

Use this topic as a reference for further information about back up and restore using CSQUTIL.

You can use the CSQUTIL utility functions for backing up and restoring queues. To back up a queue, use the COPY or SCOPY function to copy the messages from a queue onto a data set. To restore the queue, use the complementary function LOAD or SLOAD. For more information, see [IBM MQ utility program](#).

Managing buffer pools

Use this topic if you want to change or delete your buffer pools.

This topic describes how to alter and delete buffer pools. It contains these sections:

- [“How to change the number of buffers in a buffer pool” on page 543](#)
- [“How to delete a buffer pool” on page 543](#)

Buffer pools are defined during queue manager initialization, using [DEFINE BUFFPOOL](#) commands issued from the initialization input data set CSQINP1. Their attributes can be altered in response to business requirements while the queue manager is running, using the processes detailed in this topic. The queue manager records the current buffer pool attributes in checkpoint log records. These are automatically restored on subsequent queue manager restart, unless the buffer pool definition in CSQINP1 includes the REPLACE attribute.

Use the [DISPLAY USAGE](#) command to display the current buffer attributes.

You can also define buffer pools dynamically using the [DEFINE PSID](#) command with the DSN option.

If you change buffer pools dynamically, you should also update their definitions in the initialization data set CSQINP1.

See [Planung für z/OS](#) for a description of page sets, storage classes, buffers, and buffer pools, and some of the performance considerations that apply.

Note: Buffer pools use significant storage. When you increase the size of a buffer pool or define a new buffer pool ensure that sufficient storage is available. For more information, see [Address space storage](#).

How to change the number of buffers in a buffer pool

If a buffer pool is too small, the condition can result in message [CSQP020E](#) on the console, you can allocate more buffers to it using the ALTER BUFFPOOL command as follows:

1. Determine how much space is available for new buffers by looking at the [CSQY220I](#) messages in the log. The available space is reported in MB. As a buffer has a size of 4 KB, each MB of available space allows you to allocate 256 buffers. Do not allocate all the free space to buffers, as some is required for other tasks.

If the buffer pool uses fixed 4 KB pages, that is, its PAGECLAS attribute is FIXED4KB, ensure that there is sufficient real storage available on the LPAR.

2. If the reported free space is inadequate, release some buffers from another buffer pool using the command

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

where *buf-pool-id* is the buffer pool from which you want to reclaim space and *integer* is the new number of buffers to be allocated to this buffer pool, which must be smaller than the original number of buffers allocated to it.

3. Add buffers to the buffer pool you want to expand using the command

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

where *buf-pool-id* is the buffer pool to be expanded and *integer* is the new number of buffers to be allocated to this buffer pool, which must be larger than the original number of buffers allocated to it.

How to delete a buffer pool

When a buffer pool is no longer used by any page sets, delete it to release the virtual storage allocated to it.

You delete a buffer pool using the [DELETE BUFFPOOL](#) command. The command fails if any page sets are using this buffer pool.

See [“How to delete page sets” on page 542](#) for information about how to delete page sets.

Managing queue sharing groups and shared queues on z/OS

IBM MQ can use different types of shared resources, for example queue sharing groups, shared queues, and the coupling facility. Use this topic to review the procedures needed to manage these shared resources.

This section contains information about the following topics:

- [“Managing queue sharing groups” on page 544](#)
- [“Managing shared queues” on page 547](#)
- [“Managing group objects” on page 551](#)
- [“Managing the coupling facility” on page 552](#)

Managing queue sharing groups

You can add or remove a queue manager to a queue sharing group (QSG), and manage the associated Db2 tables.

This topic has sections about the following tasks:

- [“Setting up a queue sharing group” on page 544](#)
- [“Adding a queue manager to a queue sharing group” on page 545](#)
- [“Removing a queue manager from a queue sharing group” on page 546](#)
- [“Removing a queue sharing group from the Db2 tables” on page 546](#)
- [“Validating the consistency of Db2 definitions” on page 547](#)

Setting up a queue sharing group

Each queue sharing group has a name of up to four characters. The name must be unique in your network, and must be different from any queue manager names.

Follow these steps to set up a queue sharing group:

1. If this is the first queue sharing group to use the Db2 data-sharing group, [set up the Db2 environment](#).
2. [Set up the coupling facility](#).
3. Add the queue sharing group to the Db2 tables. Use the ADD QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45AQS).
4. Add a queue manager to the queue sharing group by following the steps in [“Adding a queue manager to a queue sharing group” on page 545](#)
5. Define application structures to IBM MQ by following the steps in [“Adding a coupling facility structure” on page 552](#).
6. If required, [migrate non-shared queues to shared queues](#).
7. For availability, create shared channels into and out of the queue sharing group.
 - For connections into the queue sharing group:
 - Set up a VIPA socket or hardware router to distribute workload between the available queue managers in the QSG.
 - Define a receiver channel with QSGDISP(GROUP), to ensure the channel definition is available on all queue managers in the QSG.
 - Start a listener with INDISP(GROUP), on each queue manager, for MCA channel connections into the QSG. Client connections into the QSG should still connect to a listener started with INDISP(QMGR).
 - Change applications to connect using the QSG name, rather than a specific queue manager name.

- Ensure that the channel authentication rules on all queue managers in the QSG are the same, to allow applications to connect to any queue manager in the QSG.
- For connections out of the queue sharing group:
 - Define a shared transmission queue.
 - Define the outbound channel with QSGDISP(GROUP) and DEFCDISP(SHARED).

If you convert an existing channel to a shared channel, you might need to issue the `RESET CHANNEL` command before starting the channel as the synchronization queue used by the channel will have changed.

Adding a queue manager to a queue sharing group

A queue manager can be added to an existing queue sharing group.

Note that:

- The queue sharing group must exist before you can add queue managers to it.
- A queue manager can be a member of only one queue sharing group.

Follow these steps to add a queue manager to a queue sharing group:

1. Perform the tasks in [implement ESM security controls for the queue sharing group](#) to grant the appropriate access to the queue manager and channel initiator user IDs.
2. If the queue sharing group has CF structures configured to offload data to SMDS, perform the tasks in [set up the SMDS environment](#).
3. Stop the queue manager.
4. Use the ADD QMGR function of the queue sharing group utility (CSQ5PQSG). This program is described in [the queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45AQM).
5. [Change your system parameter module](#) to add queue sharing group data:
 - a. Modify CSQ6SYSP to specify the QSGDATA parameter. See [using CSQ6SYSP](#) for more information.
 - b. Assemble and link the system parameter module. You might want to use a different name for the load module.
 - c. Change your startup process to use the new module.
6. Copy and tailor sample member thlqual.SCSQPROC(CSQ4INSS), which defines required CF structures and SYSTEM queues. Add the customized member to the CSQINP2 DD in the queue manager startup JCL.
7. Restart your queue manager using the queue sharing group system parameter module.
8. Optionally, migrate to security profiles prefixed by the queue sharing group name, instead of the queue manager name.
9. If shared channels are used for connections into the QSG, create channel authentication rules that mirror those on the other queue managers in the QSG, to allow applications to connect to any queue manager in the QSG.
10. 10. Optionally, do either of the following to allow applications connected to the queue manager in the QSG to put messages to queues hosted by other queue managers in the QSG:
 - Turn on [intra-group queuing](#) by issuing the command `ALTER QMGR IGQ(ENABLED)`.
 - Define transmission queues and channels to the other queue managers in the QSG. Defining transmission queues with the same name as the target queue managers avoids the need to define remote queues and queue manager aliases.

Note: To add a queue manager to an existing queue sharing group containing queue managers running earlier versions of IBM MQ, you must first apply the coexistence PTF for the highest version of IBM MQ in the group to every earlier version queue manager in the group.

Removing a queue manager from a queue sharing group

You can only remove a queue manager from a queue sharing group if the queue manager's logs are not needed by another process, and all SMDS owned by the queue manager are empty.

See [Deleting shared message data sets](#) and [DELETE CFSTRUCT](#) for more information.

The logs are needed if they contain:

- The latest backup of one of the coupling facility (CF) application structures used by the queue sharing group
- Data needed by a future restore process, that is, the queue manager has used a recoverable structure since the time described by the last backup exclusion interval value.

If either or both of these points apply, or an SMDS owned by the queue manager contains messages, the queue manager cannot be removed. To determine which queue managers' logs are needed for a future restore process, use the MQSC DISPLAY CFSTATUS command with the TYPE(BACKUP) option (for details of this command, see [DISPLAY CFSTATUS](#)).

Use the following steps to remove a queue manager from a queue sharing group:

1. Stop any applications connected to the queue manager that put messages to shared queues.
2. Resolve any indoubt units of work involving this queue manager.
3. Determine if there are any messages in any SMDS owned by the queue manager by issuing the command DISPLAY USAGE TYPE(SMDS).
4. If there are offloaded messages for any application structure, wait until those messages have been retrieved from the queue. The number of offloaded messages reported by DISPLAY USAGE TYPE(SMDS) should be zero before proceeding.
5. Shut the queue manager down cleanly using STOP QMGR MODE(QUIESCE).
6. Wait for an interval at least equivalent to the value of the EXCLINT parameter you will specify in the BACKUP CFSTRUCT command in the next step.
7. On another queue manager, run a CF structure backup for each recoverable CF structure by using the MQSC BACKUP CFSTRUCT command and specifying an EXCLINT value as required in the previous step.
8. Confirm that the queue manager's logs are not needed to restore any CF structures, by inspecting the output from the command DISPLAY CFSTATUS(*) TYPE(BACKUP).
9. Use the REMOVE QMGR function of the CSQ5PQSG utility to remove the queue manager from the queue sharing group. This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45RQM).
10. Before restarting the queue manager, reset the QSGDATA system parameter to its default value, and recreate the system parameter module. See [Using CSQ6SYSP](#) for information about how to tailor your system parameters.

Note, that when removing the last queue manager in a queue sharing group, you must use the FORCE option, rather than REMOVE. This removes the queue manager from the queue sharing group, while not performing the consistency checks of the queue manager logs being required for recovery. You should only perform this operation if you are deleting the queue sharing group.

Removing a queue sharing group from the Db2 tables

To remove a queue sharing group from the Db2 tables, use the REMOVE QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45RQS).

You can only remove a queue sharing group from the common Db2 data-sharing group tables after you have removed all the queue managers from the queue sharing group (as described in [“Removing a queue manager from a queue sharing group”](#) on page 546).

When the queue sharing group record is deleted from the queue sharing group administration table, all objects and administrative information relating to that queue sharing group are deleted from other IBM MQ Db2 tables. This includes shared queue and group object information.

Validating the consistency of Db2 definitions

Problems for shared queues within a queue sharing group can occur if the Db2 object definitions have, for any reason, become inconsistent.

To validate the consistency of the Db2 object definitions for queue managers, CF structures, and shared queues, use the VERIFY QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#).

Managing shared queues

Use this topic to understand how to recover, move, and migrate shared queues.

This section describes the following tasks:

- [“Recovering shared queues” on page 547](#)
- [“Moving shared queues” on page 548](#)
- [“Migrating non-shared queues to shared queues” on page 550](#)
- [Suspending a Db2 connection](#)

Recovering shared queues

IBM MQ can recover persistent messages on shared queues if all:

- Backups of the CF structures containing the messages have been performed.
- All the logs for all queue managers in the queue sharing group are available, to perform recovery from the point the backups are taken.
- Db2 is available and the structure backup table is more recent than the most recent CF structure backup.

The messages on a shared queue are stored in a coupling facility (CF) structure. Persistent messages can be put onto shared queues, and like persistent messages on non-shared queues, they are copied to the queue manager log. The MQSC [BACKUP CFSTRUCT](#) and [RECOVER CFSTRUCT](#) commands are provided to allow the recovery of a CF structure in the unlikely event of a coupling facility failure. In such circumstances, any nonpersistent messages stored in the affected structure are lost, but persistent messages can be recovered. Any further application activity using the structure is prevented until the structure has been recovered.

To enable recovery, you must back up your coupling facility list structures frequently using the MQSC [BACKUP CFSTRUCT](#) command. The messages in the CF structure are written onto the active log data set of the queue manager making the backup. It writes a record of the backup to Db2: the name of the CF structure being backed up, the name of the queue manager doing the backup, the RBA range for this backup on that queue manager log, and the backup time. Back up CF list structures even if you are not actively using shared queues, for example, if you have set up a queue sharing group intending to use it in the future.

You can recover a CF structure by issuing an MQSC [RECOVER CFSTRUCT](#) command to the queue manager that can perform the recovery; you can use any queue manager in the queue sharing group. You can specify a single CF structure to be recovered, or you can recover several CF structures simultaneously.

As noted previously, it is important that you back up your CF list structures frequently, otherwise recovering a CF structure can take a long time. Moreover, the recovery process cannot be canceled.

The definition of a shared queue is kept in a Db2 database and can therefore be recovered if necessary using standard Db2 database procedures. See [Shared queues and queue sharing groups](#) for more information.

Moving shared queues

This section describes how to perform load balancing by moving a shared queue from one coupling facility structure to another. It also describes how to move a non-shared queue to a shared queue, and how to move a shared queue to a non-shared queue.

When you move a queue, you need to define a temporary queue as part of the procedure. This is because every queue must have a unique name, so you cannot have two queues of the same name, even if the queues have different queue dispositions. IBM MQ tolerates having two queues with the same name (as in step “2” on page 548), but you cannot use the queues.

- Moving a queue from one coupling facility structure to another
- Moving a non-shared queue to a shared queue
- Moving a shared queue to a non-shared queue

Moving a queue from one coupling facility structure to another

To move queues and their messages from one CF structure to another, use the MQSC [MOVE QLOCAL](#) command. When you have identified the queue or queues that you want to move to a new CF structure, use the following procedure to move each queue:

1. Ensure that the queue you want to move is not in use by any applications, that is, the queue attributes IPPROCS and OPPROCS are zero on all queue managers in the queue sharing group.
2. Prevent applications from putting messages on the queue being moved by altering the queue definition to disable MQPUT s. Change the queue definition to PUT(DISABLED).
3. Define a temporary queue with the same attributes as the queue that is being moved using the following command:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
```

Note: If this temporary queue exists from a previous run, delete it before doing the define.

4. Move the messages to the temporary queue using the following command:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Delete the queue you are moving, using the command:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Redefine the queue that is being moved, changing the CFSTRUCT attribute, using the following command:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
```

When the queue is redefined, it is based on the temporary queue created in step “3” on page 548.

7. Move the messages back to the new queue using the command:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

8. The queue created in step “3” on page 548 is no longer required. Use the following command to delete it:

```
DELETE QL(TEMP_QUEUE)
```

9. If the queue being moved was defined in the CSQINP2 data sets, change the CFSTRUCT attribute of the appropriate DEFINE QLOCAL command in the CSQINP2 data sets. Add the REPLACE keyword so that the existing queue definition is replaced.

Figure 30 on page 549 shows a sample job for moving a queue from one CF structure to another.

```
//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)
/*
```

Figure 30. Sample job for moving a queue from one CF structure to another

Moving a non-shared queue to a shared queue

The procedure for moving a non-shared queue to a shared queue is like the procedure for moving a queue from one CF structure to another (see “Moving a queue from one coupling facility structure to another” on page 548). Figure 31 on page 549 gives a sample job to do this.

Note: Remember that messages on shared queues are subject to certain restrictions on the maximum message size, message persistence, and queue index type, so you might not be able to move some non-shared queues to a shared queue.

```
//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)
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)
/*
```

Figure 31. Sample job for moving a non-shared queue to a shared queue

Moving a shared queue to a non-shared queue

The procedure for moving a shared queue to a non-shared queue is like the procedure for moving a queue from one CF structure to another (see [“Moving a queue from one coupling facility structure to another”](#) on page 548).

Figure 32 on page 550 gives a sample job to do this.

```
//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) 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)
/*
```

Figure 32. Sample job for moving a shared queue to a non-shared queue

Migrating non-shared queues to shared queues

There are two stages to migrating non-shared queues to shared queues:

- Migrating the first (or only) queue manager in the queue sharing group
- Migrating any other queue managers in the queue sharing group

Migrating the first (or only) queue manager in the queue sharing group

Figure 31 on page 549 shows an example job for moving a non-shared queue to a shared queue. Do this for each queue that needs migrating.

Note:

1. Messages on shared queues are subject to certain restrictions on the maximum message size, message persistence, and queue index type, so you might not be able to move some non-shared queues to a shared queue.
2. You must use the correct index type for shared queues. If you migrate a transmission queue to be a shared queue, the index type must be MSGID.

If the queue is empty, or you do not need to keep the messages that are on it, migrating the queue is simpler. [Figure 33 on page 551](#) shows an example job to use in these circumstances.

```

//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 *
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)
/*

```

Figure 33. Sample job for moving a non-shared queue without messages to a shared queue

Migrating any other queue managers in the queue sharing group

1. For each queue that does not have the same name as an existing shared queue, move the queue as described in [Figure 31 on page 549](#) or [Figure 33 on page 551](#).
2. For queues that have the same name as an existing shared queue, move the messages to the shared queue using the commands shown in [Figure 34 on page 551](#).

```

MOVE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR) TOQLOCAL(Queue_TO_MOVE)
DELETE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR)

```

Figure 34. Moving messages from a non-shared queue to an existing shared queue

Suspending a connection to Db2

If you want to apply maintenance or service to the Db2 tables or package related to shared queues without stopping your queue manager, you must temporarily disconnect queue managers in the data sharing group (DSG) from Db2.

To do this:

1. Use the MQSC command [SUSPEND QMGR FACILITY](#)(Db2).
2. Do the binds.
3. Reconnect to Db2 using the MQSC command [RESUME QMGR FACILITY](#)(Db2)

Note that there are restrictions on the use of these commands.



Attention: While the Db2 connection is suspended, the following operations will not be available. Therefore, you need to do this work during a time when your enterprise is at its least busy.

- Access to Shared queue objects for administration (define, delete,alter)
- Starting shared channels
- Storing messages in Db2
- Backup or recover CFSTRUCT

Managing group objects

Use this topic to understand how to work with group objects.

IBM MQ automatically copies the definition of a group object to page set zero of each queue manager that uses it. You can alter the copy of the definition temporarily, and IBM MQ allows you to refresh the page set copies from the repository copy. IBM MQ always tries to refresh the page set copies from the repository

copy on start-up (for channel objects, this is done when the channel initiator restarts). This ensures that the page set copies reflect the version on the repository, including any changes that were made when the queue manager was inactive.

There are circumstances under which the refresh is not performed, for example:

- If a copy of the queue is open, a refresh that would change the usage of the queue fails.
- If a copy of a queue has messages on it, a refresh that would delete that queue fails.

In these circumstances, the refresh is not performed on that copy, but is performed on the copies on all other queue managers. Check for and correct any problems with copy objects after adding, changing, or deleting a group object, and at queue manager or channel initiator restart.

Managing the coupling facility

Use this topic to understand how to add or remove coupling facility (CF) structures.

This section describes the following tasks:

- [“Adding a coupling facility structure” on page 552](#)
- [“Removing a coupling facility structure” on page 552](#)

Adding a coupling facility structure

To add a coupling facility structure, use the following procedure:

1. Define the CF structure in the CFRM policy data set. The information about setting up the coupling facility in [Set up the coupling facility](#) describes the rules for naming coupling facility structures, and how to define structures in the CFRM policy data set.
2. If you want to configure the structure to offload message data to SMDS, allocate and preformat data sets. See [creating a shared message data set](#) for details.
3. Define the structure to IBM MQ using the [DEFINE CFSTRUCT](#) command.

Removing a coupling facility structure

To remove a coupling facility structure, use the following procedure:

1. Use the following command to get a list of all the queues using the coupling facility structure that you want to delete:

```
DISPLAY QUEUE(*) QSGDISP(SHARED) CFSTRUCT(structure-name)
```

2. Delete all the queues that use the structure.
3. Delete the CF structure from IBM MQ using the [DELETE CFSTRUCT](#) command.
4. If the structure was configured to offload message data to SMDS, delete the SMDS.
5. Remove the structure definition from your CFRM policy data set and run the IXCMIAPU utility. (This is the reverse of the customization task set up the coupling facility, described in [Set up the coupling facility](#).)

Tuning coupling facility list monitoring

Use this topic to understand coupling facility list monitoring

Coupling facility (CF) list monitoring is used to monitor the state of list structures containing IBM MQ shared queues. When a message is added to a shared queue, and the queue's depth transitions from zero to non-zero, the CF notifies all queue managers in the queue sharing group. When notified the

queue managers might perform a number of actions, including notifying trigger monitors that are using TRIGGER(FIRST), or applications which are performing a get-wait.

By default, the CF notifies all queue managers in the queue sharing group at the same time. In certain configurations this can cause problems, such as:

- Skewed workload distribution, where a large percentage of messages go to a particular queue manager in the queue sharing group, often the queue manager running on the fastest LPAR, or which is closest to the CF, or
- A large number of failed gets, resulting in wasted CPU time.

z/OS V2R3 introduces a new coupling facility resource manager (CFRM) attribute called **KEYRNOTIFYDELAY**, which can be used with list structures containing shared queues (that is, application structures, and not the admin structure), and which can, for certain workloads, minimize the effects of workload skewing and empty MQGET calls, or empty MQGET calls.

KEYRNOTIFYDELAY can only be set on structures in a CF, running at CFLEVEL 22 or higher.

Its value must be one to seven decimal digits, in a range from 0 to 1,000,000 microseconds. If set to a non-zero value and the depth of a queue transitions from zero to non-zero, the CF selects a single queue manager from the queue sharing group, and notifies that queue manager before all the other queue managers in the group.

The queue manager is selected in a round-robin manner. If the selected queue manager does not process the message inside the time interval described by **KEYRNOTIFYDELAY** all the other queue managers in the queue sharing group will also be notified.

More information on **KEYRNOTIFYDELAY** is available here: [Understanding Keyrange Monitoring Notification Delay](#).

Note that there are two similar CFRM attributes called **LISTNOTIFYDELAY** and **SUBNOTIFYDELAY**. Neither of these has any measurable effect on IBM MQ workload.

Recovery and restart on z/OS

Use this topic to understand the recovery and restart mechanisms used by IBM MQ.

Restarting IBM MQ

After a queue manager terminates there are different restart procedures needed depending on how the queue manager terminated. Use this topic to understand the different restart procedures that you can use.

This topic contains information about how to restart your queue manager in the following circumstances:

- [“Restarting after a normal shutdown” on page 553](#)
- [“Restarting after an abnormal termination” on page 554](#)
- [“Restarting if you have lost your page sets” on page 554](#)
- [“Restarting if you have lost your log data sets” on page 554](#)
- [Restarting if you have lost your CF structures](#)

Restarting after a normal shutdown

If the queue manager was stopped with the STOP QMGR command, the system finishes its work in an orderly way and takes a termination checkpoint before stopping. When you restart the queue manager, it uses information from the system checkpoint and recovery log to determine the system status at shutdown.

To restart the queue manager, issue the START QMGR command as described in [“Using MQSC to start and stop a queue manager on z/OS” on page 479](#).

Restarting after an abnormal termination

IBM MQ automatically detects whether restart follows a normal shutdown or an abnormal termination.

Starting the queue manager after it has terminated abnormally is different from starting it after the STOP QMGR command has been issued. If the queue manager terminates abnormally, it terminates without being able to finish its work or take a termination checkpoint.

To restart the queue manager, issue the START QMGR command as described in [“Using MQSC to start and stop a queue manager on z/OS” on page 479](#). When you restart a queue manager after an abnormal termination, it refreshes its knowledge of its status at termination using information in the log, and notifies you of the status of various tasks.

Normally, the restart process resolves all inconsistent states. But, in some cases, you must take specific steps to resolve inconsistencies. This is described in [“Recovering units of work manually” on page 566](#).

Restarting if you have lost your page sets

If you have lost your page sets, you need to restore them from your backup copies before you can restart the queue manager. This is described in [“How to back up and recover page sets” on page 539](#).

The queue manager might take a long time to restart under these circumstances because of the length of time needed for media recovery.

Restarting if you have lost your log data sets

If, after stopping a queue manager (using the STOP QMGR command), both copies of the log are lost or damaged, you can restart the queue manager providing you have a consistent set of page sets (produced using [Method 1: Full backup](#)).

Follow this procedure:

1. Define new page sets to correspond to each existing page set in your queue manager. See [Task 15: Define your page sets](#) for information about page set definition.
Ensure that each new page set is larger than the corresponding source page set.
2. Use the FORMAT function of CSQUTIL to format the destination page set. See [Formatting page sets](#) for more details.
3. Use the RESETPAGE function of CSQUTIL to copy the existing page sets or reset them in place, and reset the log RBA in each page. See [Copying a page set and resetting the log](#) for more information about this function.
4. Redefine your queue manager log data sets and BSDS using CSQJU003 (see [The change log inventory utility](#)).
5. Restart the queue manager using the new page sets. To do this, you do one of the following:
 - Change the queue manager started task procedure to reference the new page sets. See [Task 6: Create procedures for the IBM MQ queue manager](#) for more information.
 - Use Access Method Services to delete the old page sets and then rename the new page sets, giving them the same names as the old page sets.

Attention: Before you delete any IBM MQ page set, ensure that you have made the required backup copies.

If the queue manager is a member of a queue sharing group, GROUP and SHARED object definitions are not normally affected by lost or damaged logs. However, if any shared-queue messages are involved in a unit of work that was covered by the lost or damaged logs, the effect on such uncommitted messages is unpredictable.

Note: If logs are damaged and the queue manager is a member of a queue sharing group, the ability to recover shared persistent messages might be lost. Issue a BACKUP CFSTRUCT command immediately on another active queue manager in the queue sharing group for all CF structures with the RECOVER(YES) attribute.

Restarting if you have lost your CF structures

You do not need to restart if you lose your CF structures, because the queue manager does not terminate.

Alternative site recovery on z/OS

You can recover a single queue manager or a queue sharing group, or consider disk mirroring.

See the following sections for more details:

- [Recovering a single queue manager at an alternative site](#)
- [Recovering a queue sharing group.](#)
 - [CF structure media recovery](#)
 - [Backing up the queue sharing group at the prime site](#)
 - [Recovering a queue sharing group at the alternative site](#)
- [Using disk mirroring](#)

Recovering a single queue manager at an alternative site

If a total loss of an IBM MQ computing center occurs, you can recover on another queue manager or queue sharing group at a recovery site. (See “[Recovering a queue sharing group at the alternative site](#)” on page 558 for the alternative site recovery procedure for a queue sharing group.)

To recover on another queue manager at a recovery site, you must regularly back up the page sets and the logs. As with all data recovery operations, the objectives of disaster recovery are to lose as little data, workload processing (updates), and time, as possible.

At the recovery site:

- The recovery queue managers **must** have the same names as the lost queue managers.
- The system parameter module (for example, CSQZPARM) used on each recovery queue manager must contain the same parameters as the corresponding lost queue manager.

When you have done this, reestablish all your queue managers as described in the following procedure. This can be used to perform disaster recovery at the recovery site for a single queue manager. It assumes that all that is available are:

- Copies of the archive logs and BSDSs created by normal running at the primary site (the active logs will have been lost along with the queue manager at the primary site).
- Copies of the page sets from the queue manager at the primary site that are the same age or older than the most recent archive log copies available.

You can use dual logging for the active and archive logs, in which case you need to apply the BSDS updates to both copies:

1. Define new page set data sets and load them with the data in the copies of the page sets from the primary site.
2. Define new active log data sets.
3. Define a new BSDS data set and use Access Method Services REPRO to copy the most recent archived BSDS into it.
4. Use the print log map utility CSQJU004 to print information from this most recent BSDS. At the time this BSDS was archived, the most recent archived log you have would have just been truncated as an active log, and does not appear as an archived log. Record the STARTRBA and ENDRBA of this log.

5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the STARTRBA and ENDRBA recorded in Step “4” on page 555.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. Use the NEWLOG option of CSQJU003 to add active logs to the BSDS, do not specify STARTRBA or ENDRBA.
8. Use CSQJU003 to add a restart control record to the BSDS. Specify CRESTART CREATE, ENDRBA=highrba, where highrba is the high RBA of the most recent archive log available (found in Step “4” on page 555), plus 1.

The BSDS now describes all active logs as being empty, all the archived logs you have available, and no checkpoints beyond the end of your logs.

9. Restart the queue manager with the START QMGR command. During initialization, an operator reply message such as the following is issued:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.
REPLY Y TO CONTINUE, N TO CANCEL
```

Type Y to start the queue manager. The queue manager starts, and recovers data up to ENDRBA specified in the CRESTART statement.

See [IBM MQ utilities on z/OS reference](#) for information about using CSQJU003 and CSQJU004.

The following example shows sample input statements for CSQJU003 for steps 6, 7, and 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
```

The things you need to consider for restarting the channel initiator at the recovery site are like those faced when using ARM to restart the channel initiator on a different z/OS image. See [“Using ARM in an IBM MQ network” on page 564](#) for more information. Your recovery strategy should also cover recovery of the IBM MQ product libraries and the application programming environments that use IBM MQ (CICS , for example).

Other functions of the change log inventory utility (CSQJU003) can also be used in disaster recovery scenarios. The HIGHRBA function allows the update of the highest RBA written and highest RBA offloaded values within the bootstrap data set. The CHECKPT function allows the addition of new checkpoint queue records or the deletion of existing checkpoint queue records in the BSDS.

Attention: These functions might affect the integrity of your IBM MQ data. Only use them in disaster recovery scenarios under the guidance of IBM service personnel.

Fast copy techniques

If copies of all the page sets and logs are made while the queue manager is frozen, the copies will be a consistent set that can be used to restart the queue manager at an alternative site. They typically enable a much faster restart of the queue manager, as there is little media recovery to be performed.

Use the SUSPEND QMGR LOG command to freeze the queue manager. This command flushes buffer pools to the page sets, takes a checkpoint, and stops any further log write activity. Once log write activity has been suspended, the queue manager is effectively frozen until you issue a RESUME QMGR LOG command. While the queue manager is frozen, the page sets and logs can be copied.

By using copying tools such as FLASHCOPY or SNAPSHOT to rapidly copy the page sets and logs, the time during which the queue manager is frozen can be reduced to a minimum.

Within a queue sharing group, however, the SUSPEND QMGR LOG command might not be such a good solution. To be effective, the copies of the logs must all contain the same point in time for recovery, which means that the SUSPEND QMGR LOG command must be issued on all queue managers within the queue sharing group simultaneously, and therefore the entire queue sharing group will be frozen for some time.

Recovering a queue sharing group

In the event of a prime site disaster, you can restart a queue sharing group at a remote site using backup data sets from the prime site. To recover a queue sharing group you need to coordinate the recovery across all the queue managers in the queue sharing group, and coordinate with other resources, primarily Db2. This section describes these tasks in detail.

- [CF structure media recovery](#)
- [Backing up the queue sharing group at the prime site](#)
- [Recovering a queue sharing group at the alternative site](#)

CF structure media recovery

Media recovery of a CF structure used to hold persistent messages on a shared queue, relies on having a backup of the media that can be forward recovered by the application of logged updates. Take backups of your CF structures periodically using the MQSC BACKUP CFSTRUCT command. All updates to shared queues (MQGETs and MQPUTs) are written on the log of the queue manager where the update is performed. To perform media recovery of a CF structure you must apply logged updates to that backup from the logs of all the queue managers that have used that CF structure. When you use the MQSC RECOVER CFSTRUCT command, IBM MQ automatically merges the logs from relevant queue managers, and applies the updates to the most recent backup.

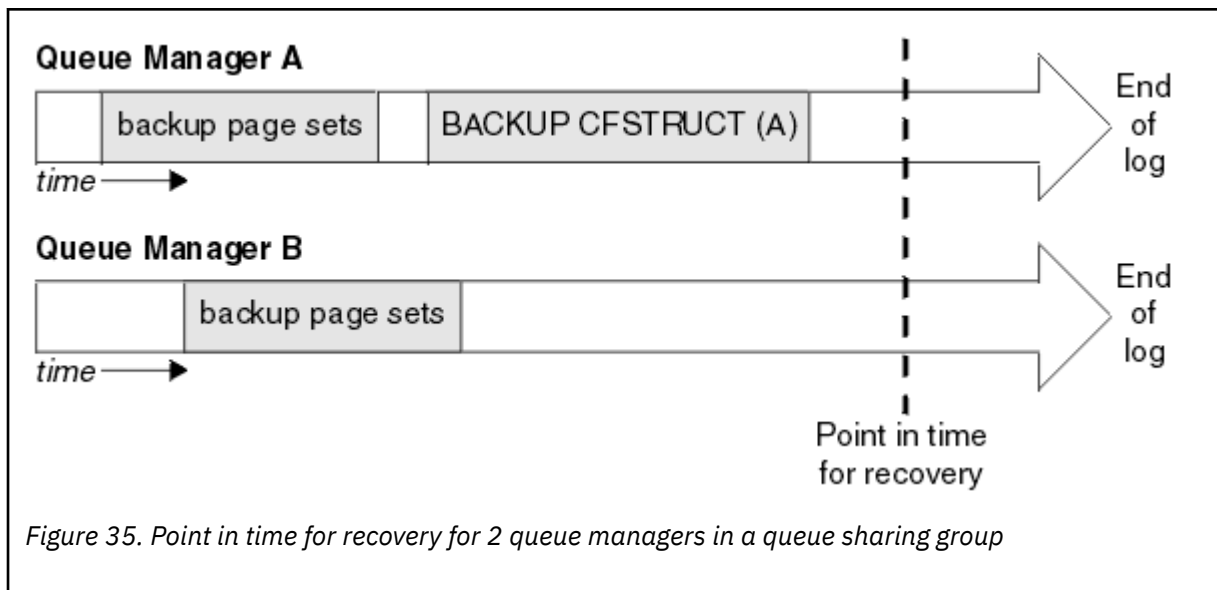
The CF structure backup is written to the log of the queue manager that processed the BACKUP CFSTRUCT command, so there are no additional data sets to be collected and transported to the alternative site.

Backing up the queue sharing group at the prime site

At the prime site you need to establish a consistent set of backups on a regular basis, which can be used in the event of a disaster to rebuild the queue sharing group at an alternative site. For a single queue manager, recovery can be to an arbitrary point in time, typically the end of the logs available at the remote site. However, where persistent messages have been stored on a shared queue, the logs of all the queue managers in the queue sharing group must be merged to recover shared queues, as any queue manager in the queue sharing group might have performed updates (MQPUTs or MQGETs) on the queue.

For recovery of a queue sharing group, you need to establish a point in time that is within the log range of the log data of all queue managers. However, as you can only **forward** recover media from the log, this point in time must be after the BACKUP CFSTRUCT command has been issued and after any page set backups have been performed. (Typically, the point in time for recovery might correspond to the end of a business day or week.)

The following diagram shows time lines for two queue managers in a queue sharing group. For each queue manager, fuzzy backups of page sets are taken (see [Method 2: Fuzzy backup](#)). On queue manager A, a BACKUP CFSTRUCT command is issued. Subsequently, an ARCHIVE LOG command is issued on each queue manager to truncate the active log, and copy it to media offline from the queue manager, which can be transported to the alternative site. End of log identifies the time at which the ARCHIVE LOG command was issued, and therefore marks the extent of log data typically available at the alternative site. The point in time for recovery must lie between the end of any page set or CF structure backups, and the earliest end of log available at the alternative site.



IBM MQ records information associated with the CF structure backups in a table in Db2. Depending on your requirements, you might want to coordinate the point in time for recovery of IBM MQ with that for Db2, or it might be sufficient to take a copy of the IBM MQ CSQ.ADMIN_B_STRBACKUP table after the BACKUP CFSTRUCT commands have finished.

To prepare for a recovery:

1. Create page set backups for each queue manager in the queue sharing group.
2. Issue a BACKUP CFSTRUCT command for each CF structure with the RECOVER(YES) attribute. You can issue these commands from a single queue manager, or from different queue managers within the queue sharing group to balance the workload.
3. Once all the backups have completed, issue an ARCHIVE LOG command to switch the active log and create copies of the logs and BSDS of each queue manager in the queue sharing group.
4. Transport the page set backups, the archived logs, the archived BSDS of all the queue managers in the queue sharing group, and your chosen Db2 backup information, off-site.

Recovering a queue sharing group at the alternative site

Before you can recover the queue sharing group, you need to prepare the environment:

1. If you have old information in your coupling facility from practice startups when you installed the queue sharing group, you need to clean this out first:

Note: If you do not have old information in the coupling facility, you can omit this step.

- a. Enter the following z/OS command to display the CF structures for this queue sharing group:

```
D XCF,STRUCTURE,STRNAME= qsgname
```

- b. For all structures that start with the queue sharing group name, use the z/OS command SETXCF FORCE CONNECTION to force the connection off those structures:

```
SETXCF FORCE,CONNECTION,STRNAME= strname,CONNAME=ALL
```

- c. Delete all the CF structures using the following command for each structure:

```
SETXCF FORCE,STRUCTURE,STRNAME= strname
```

2. Restore Db2 systems and data-sharing groups.
3. Recover the CSQ.ADMIN_B_STRBACKUP table so that it contains information about the most recent structure backups taken at the prime site.

Note: It is important that the STRBACKUP table contains the most recent structure backup information. Older structure backup information might require data sets that you have discarded as a result of the information given by a recent DISPLAY USAGE TYPE(DATASET) command, which would mean that your recovered CF structure would not contain accurate information.

4. Run the ADD QMGR command of the CSQ5PQSG utility for every queue manager in the queue sharing group. This will restore the XCF group entry for each queue manager.

When you run the utility in this scenario, the following messages are 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
```

To recover the queue managers in the queue sharing group:

1. Define new page set data sets and load them with the data in the copies of the page sets from the primary site.
2. Define new active log data sets.
3. Define a new BSDS data set and use Access Method Services REPRO to copy the most recent archived BSDS into it.
4. Use the print log map utility CSQJU004 to print information from this most recent BSDS. At the time this BSDS was archived, the most recent archived log you have would have just been truncated as an active log, and does not appear as an archived log. Record the STARTRBA, STARTLRSN, ENDRBA, and ENDLRSN values of this log.
5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the values recorded in Step “4” on page 559.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. Use the NEWLOG option of CSQJU003 to add active logs to the BSDS, do not specify STARTRBA or ENDRBA.
8. Calculate the *recoverylrsn* for the queue sharing group. The *recoverylrsn* is the lowest of the ENDLRSNs across all queue managers in the queue sharing group (as recorded in Step “4” on page 559), minus 1. For example, if there are two queue managers in the queue sharing group, and the ENDLRSN for one of them is B713 3C72 22C5, and for the other is B713 3D45 2123, the *recoverylrsn* is B713 3C72 22C4.
9. Use CSQJU003 to add a restart control record to the BSDS. Specify:

```
CRESTART CREATE,ENDLRSN= recoverylrsn
```

where *recoverylrsn* is the value you recorded in Step “8” on page 559.

The BSDS now describes all active logs as being empty, all the archived logs you have available, and no checkpoints beyond the end of your logs.

You must add the CRESTART record to the BSDS for each queue manager within the queue sharing group.

10. Restart each queue manager in the queue sharing group with the START QMGR command. During initialization, an operator reply message such as the following is issued:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.  
REPLY Y TO CONTINUE, N TO CANCEL
```

Reply Y to start the queue manager. The queue manager starts, and recovers data up to ENDRBA specified in the CRESTART statement.

The first IBM MQ queue manager started can rebuild the admin structure partitions for other members of the queue sharing group as well as its own, and it is no longer necessary to restart each queue manager in the queue sharing group at this stage.

11. When the admin structure data for all queue managers has been rebuilt, issue a RECOVER CFSTRUCT command for each CF application structure.

If you issue the RECOVER CFSTRUCT command for all structures on a single queue manager, the log merge process is only performed once, so is quicker than issuing the command on a different queue manager for each CF structure, where each queue manager has to perform the log merge step.

When conditional restart processing is used in a queue sharing group, IBM MQ queue managers, performing peer admin rebuild, check that peers BSDS contain the same CRESTART LRSN as their own. This is to ensure the integrity of the rebuilt admin structure. It is therefore important to restart other peers in the QSG, so they can process their own CRESTART information, before the next unconditional restart of any member of the group.

Using disk mirroring

Many installations now use disk mirroring technologies such as IBM Metro Mirror (formerly PPRC) to make synchronous copies of data sets at an alternative site. In such situations, many of the steps detailed become unnecessary as the IBM MQ page sets and logs at the alternative site are effectively identical to those at the prime site. Where such technologies are used, the steps to restart a queue sharing group at an alternative site may be summarized as:

- Clear IBM MQ CF structures at the alternative site. (These often contain residual information from any previous disaster recovery exercise).
- Restore Db2 systems and all tables in the database used by the IBM MQ queue sharing group.
- Restart queue managers. Before IBM WebSphere MQ 7.0.1, it is necessary to restart each queue manager defined in the queue sharing group as each queue manager recovers its own partition of the admin structure during queue manager restart. After each queue manager has been restarted, those not on their home LPAR can be shut down again. The first IBM MQ queue manager started rebuilds the admin structure partitions for other members of the queue sharing group as well as its own, and it is no longer necessary to restart each queue manager in the queue sharing group.
- After the admin structure has been rebuilt, recover the application structures.

IBM MQ for z/OS supports use of zHyperWrite when writing to active logs mirrored using Metro Mirror. zHyperWrite can help reduce the performance impact of using Metro Mirror; see [Using Metro Mirror with IBM MQ](#) for more information.

Reinitializing a queue manager

If the queue manager has terminated abnormally you might not be able to restart it. This could be because your page sets or logs have been lost, truncated, or corrupted. If this has happened, you might have to reinitialize the queue manager (perform a cold start).

Attention

Only perform a cold start if you cannot restart the queue manager any other way. Performing a cold start enables you to recover your queue manager and your object definitions; you will **not** be able to recover your message data. Check that none of the other restart scenarios described in this topic work for you before you do this.

When you have restarted, all your IBM MQ objects are defined and available for use, but there is no message data.

Note: Do not reinitialize a queue manager while it is part of a cluster. You must first remove the queue manager from the cluster (using RESET CLUSTER commands on the other queue managers in the cluster), then reinitialize it, and finally reintroduce it to the cluster as a new queue manager.

This is because during reinitialization, the queue manager identifier (QMID) is changed, so any cluster object with the old queue manager identifier must be removed from the cluster.

For further information see the following sections:

- [Reinitializing a queue manager that is not in a queue sharing group](#)
- [Reinitializing queue managers in a queue sharing group](#)

Reinitializing a queue manager that is not in a queue sharing group

To reinitialize a queue manager, follow this procedure:

1. Prepare the object definition statements that to be used when you restart the queue manager. To do this, either:
 - If page set zero is available, use the CSQUTIL SDEFS function (see [Producing a list of IBM MQ define commands](#)). You must get definitions for all object types (authentication information objects, CF structures, channels, namelists, processes, queues, and storage classes).
 - If page set zero is not available, use the definitions from the last time you backed up your object definitions.
2. Redefine your queue manager data sets (do not do this until you have completed step “1” on page [561](#)).
See [creating the bootstrap and log data sets](#) and [defining your page sets](#) for more information.
3. Restart the queue manager using the newly defined and initialized log data sets, BSDS, and page sets. Use the object definition input statements that you created in step “1” on page [561](#) as input in the CSQINP2 initialization input data set.

Reinitializing queue managers in a queue sharing group

In a queue sharing group, reinitializing a queue manager is more complex. It might be necessary to reinitialize one or more queue managers because of page set or log problems, but there might also be problems with Db2 or the coupling facility to deal with. Because of this, there are a number of alternatives:

Cold start

Reinitializing the entire queue sharing group involves forcing all the coupling facilities structures, clearing all object definitions for the queue sharing group from Db2, deleting or redefining the logs and BSDS, and formatting page sets for all the queue managers in the queue sharing group.

Shared definitions retained

Delete or redefine the logs and BSDS, format page sets for all queue managers in the queue sharing group, and force all the coupling facilities structures. On restart, all messages will have been deleted. The queue managers re-create COPY objects that correspond to GROUP objects that still exist in the Db2 database. Any shared queues still exist and can be used.

Single queue manager reinitialized

Delete or redefine the logs and BSDS, and format page sets for the single queue manager (this deletes all its private objects and messages). On restart, the queue manager re-creates COPY objects that correspond to GROUP objects that still exist in the Db2 database. Any shared queues still exist, as do the messages on them, and can be used.

Point in time recovery of a queue sharing group

This is the alternative site disaster recovery scenario.

Shared objects are recovered to the point in time achieved by Db2 recovery (described in [A Db2 system fails](#)). Each queue manager can be recovered to a point in time achievable from the backup copies available at the alternative site.

Persistent messages can be used in queue sharing groups, and can be recovered using the MQSC RECOVER CFSTRUCT command. Note that this command recovers to the time of failure. However, there is no recovery of nonpersistent shared queue messages; they are lost unless you have made backup copies independently using the COPY function of the CSQUTIL utility program.

It is not necessary to try to restore each queue manager to the same point in time because there are no interdependencies between the local objects on different queue managers (which are what is actually being recovered), and the queue manager resynchronization with Db2 on restart creates or deletes COPY objects as necessary on a queue manager by queue manager basis.

Using the z/OS Automatic Restart Manager (ARM)

Use this topic to understand how you can use ARM to automatically restart your queue managers.

This section contains information about the following topics:

- [“What is the ARM?” on page 562](#)
- [“ARM policies” on page 563](#)
- [“Using ARM in an IBM MQ network” on page 564](#)

What is the ARM?

The z/OS Automatic Restart Manager (ARM) is a z/OS recovery function that can improve the availability of your queue managers. When a job or task fails, or the system on which it is running fails, ARM can restart the job or task without operator intervention.

If a queue manager or a channel initiator has failed, ARM restarts it on the same z/OS image. If z/OS, and hence a whole group of related subsystems and applications have failed, ARM can restart all the failed systems automatically, in a predefined order, on another z/OS image within the sysplex. This is called a *cross-system restart*.

Restart the channel initiator by ARM only in exceptional circumstances. If the queue manager is restarted by ARM, restart the channel initiator from the CSQINP2 initialization data set (see [“Using ARM in an IBM MQ network” on page 564](#)).

You can use ARM to restart a queue manager on a different z/OS image within the sysplex in the event of z/OS failure. The network implications of IBM MQ ARM restart on a different z/OS image are described in [“Using ARM in an IBM MQ network” on page 564](#).

To enable automatic restart:

- Set up an ARM couple data set.
- Define the automatic restart actions that you want z/OS to perform in an *ARM policy*.
- Start the ARM policy.

Also, IBM MQ must register with ARM at startup (this happens automatically).

Note: If you want to restart queue managers in different z/OS images automatically, you must define every queue manager as a subsystem in each z/OS image on which that queue manager might be restarted, with a sysplex wide unique four character subsystem name.

ARM couple data sets

Ensure that you define the couple data sets required for ARM, and that they are online and active before you start any queue manager for which you want ARM support. IBM MQ automatic ARM registration fails if the couple data sets are not available at queue manager startup. In this situation, IBM MQ assumes that the absence of the couple data set means that you do not want ARM support, and initialization continues.

See *z/OS MVS Setting up a Sysplex* for information about ARM couple data sets.

ARM policies

The Automatic Restart Manager policies are user-defined rules that control ARM functions that can control any restarts of a queue manager.

ARM functions are controlled by a user-defined *ARM policy*. Each z/OS image running a queue manager instance that is to be restarted by ARM must be connected to an ARM couple data set with an active ARM policy.

IBM provides a default ARM policy. You can define new policies, or override the policy defaults by using the *administrative data utility* (IXCMIAPU) provided with z/OS. *z/OS MVS Setting up a Sysplex* describes this utility, and includes full details of how to define an ARM policy.

Figure 36 on page 563 shows an example of an ARM policy. This sample policy restarts any queue manager within a sysplex, if either the queue manager failed, or a whole system failed.

```
//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(SYSMQGRMQ*) /* These jobs to be restarted by ARM */
/*
```

Figure 36. Sample ARM policy

For more information see:

- [Defining an ARM policy](#)
- [Activating an ARM policy](#)
- [Registering with ARM](#)

Defining an ARM policy

Set up your ARM policy as follows:

- Define RESTART_GROUPS for each queue manager instance that also contain any CICS or IMS subsystems that connect to that queue manager instance. If you use a subsystem naming convention, you might be able to use the '?' and '*' wild-card characters in your element names to define RESTART_GROUPS with minimum definition effort.
- Specify TERMTYPE(ELEMTERM) for your channel initiators to indicate that they will be restarted only if the channel initiator has failed and the z/OS image has not failed.

- Specify `TERMTYPE(ALLTERM)` for your queue managers to indicate that they will be restarted if either the queue manager has failed or the z/OS image has failed.
- Specify `RESTART_METHOD(BOTH, PERSIST)` for both queue managers and channel initiators. This tells ARM to restart using the JCL it saved (after resolution of system symbols) during the last startup. It tells ARM to do this irrespective of whether the individual element failed, or the z/OS image failed.
- Accept the default values for all the other ARM policy options.

Activating an ARM policy

To start your automatic restart management policy, issue the following z/OS command:

```
SETXCF START,POLICY,TYPE=ARM,POLNAME= mypol
```

When the policy is started, all systems connected to the ARM couple data set use the same active policy. Use the `SETXCF STOP` command to disable automatic restarts.

Registering with ARM

IBM MQ registers automatically as an *ARM element* during queue manager startup (subject to ARM availability). It deregisters during its shutdown phase, unless requested not to.

At startup, the queue manager determines whether ARM is available. If it is, IBM MQ registers using the name `SYSMQMGR ssid`, where *ssid* is the four character queue manager name, and `SYSMQMGR` is the element type.

The `STOP QMGR MODE(QUIESCE)` and `STOP QMGR MODE(FORCE)` commands deregister the queue manager from ARM (if it was registered with ARM at startup). This prevents ARM restarting this queue manager. The `STOP QMGR MODE(RESTART)` command does not deregister the queue manager from ARM, so it is eligible for immediate automatic restart.

Each channel initiator address space determines whether ARM is available, and if so registers with the element name `SYSMQCH ssid`, where *ssid* is the queue manager name, and `SYSMQCH` is the element type.

The channel initiator is always deregistered from ARM when it stops normally, and remains registered only if it ends abnormally. The channel initiator is always deregistered if the queue manager fails.

Using ARM in an IBM MQ network

You can set up your queue manager so that the channel initiators and associated listeners are started automatically when the queue manager is restarted.

To ensure fully automatic queue manager restart on the same z/OS image for both LU 6.2 and TCP/IP communication protocols:

- Start your listeners automatically by adding the appropriate `START LISTENER` command to the `CSQINPX` data set.
- Start your channel initiator automatically by adding the appropriate `START CHINIT` command to the `CSQINP2` data set.

For restarting a queue manager with TCP/IP or LU6.2, see

- [“Restarting on a different z/OS image with TCP/IP” on page 565](#)
- [“Restarting on a different z/OS image with LU 6.2” on page 566](#)

See [Task 13: Customize the initialization input data sets](#) for information about the `CSQINP2` and `CSQINPX` data sets.

Restarting on a different z/OS image with TCP/IP

If you are using TCP/IP as your communication protocol, and you are using virtual IP addresses, you can configure these to recover on other z/OS images, allowing channels connecting to that queue manager to reconnect without any changes. Otherwise, you can reallocate a TCP/IP address after moving a queue manager to a different z/OS image only if you are using clusters or if you are connecting to a queue sharing group using a WLM dynamic Domain Name System (DNS) logical group name.

- [When using clustering](#)
- [When connecting to a queue sharing group](#)

When using clustering

z/OS ARM responds to a system failure by restarting the queue manager on a different z/OS image in the same sysplex; this system has a different TCP/IP address to the original z/OS image. The following explains how you can use IBM MQ clusters to reassign a queue manager's TCP/IP address after it has been moved by ARM restart to a different z/OS image.

When a client queue manager detects the queue manager failure (as a channel failure), it responds by reallocating suitable messages on its cluster transmission queue to a different server queue manager that hosts a different instance of the target cluster queue. However, it cannot reallocate messages that are bound to the original server by affinity constraints, or messages that are in doubt because the server queue manager failed during end-of-batch processing. To process these messages, do the following:

1. Allocate a different cluster-receiver channel name and a different TCP/IP port to each z/OS queue manager. Each queue manager needs a different port so that two systems can share a single TCP/IP stack on a z/OS image. One of these is the queue manager originally running on that z/OS image, and the other is the queue manager that ARM will restart on that z/OS image following a system failure. Configure each port on each z/OS image, so that ARM can restart any queue manager on any z/OS image.
2. Create a different channel initiator command input file (CSQINPX) for each queue manager and z/OS image combination, to be referenced during channel initiator startup.

Each CSQINPX file must include a START LISTENER PORT(port) command specific to that queue manager, and an ALTER CHANNEL command for a cluster-receiver channel specific to that queue manager and z/OS image combination. The ALTER CHANNEL command needs to set the connection name to the TCP/IP name of the z/OS image on which it is restarted. It must include the port number specific to the restarted queue manager as part of the connection name.

The start-up JCL of each queue manager can have a fixed data set name for this CSQINPX file, and each z/OS image must have a different version of each CSQINPX file on a non-shared DASD volume.

If an ARM restart occurs, IBM MQ advertises the changed channel definition to the cluster repository, which in turn publishes it to all the client queue managers that have expressed an interest in the server queue manager.

The client queue manager treats the server queue manager failure as a channel failure, and tries to restart the failed channel. When the client queue manager learns the new server connection-name, the channel restart reconnects the client queue manager to the restarted server queue manager. The client queue manager can then resynchronize its messages, resolve any in-doubt messages on the client queue manager's transmission queue, and normal processing can continue.

When connecting to a queue sharing group

When connecting to a queue sharing group through a TCP/IP dynamic Domain Name System (DNS) logical group name, the connection name in your channel definition specifies the logical group name of your queue sharing group, not the host name or IP address of a physical machine. When this channel starts, it connects to the dynamic DNS and is then connected to one of the queue managers in the queue sharing group. This process is explained in [Setting up communication for IBM MQ for z/OS using queue sharing groups](#).

In the unlikely event of an image failure, one of the following occurs:

- The queue managers on the failing image de-register from the dynamic DNS running on your sysplex. The channel responds to the connection failure by entering RETRYING state and then connects to the dynamic DNS running on the sysplex. The dynamic DNS allocates the inbound request to one of the remaining members of the queue sharing group that is still running on the remaining images.
- If no other queue manager in the queue sharing group is active and ARM restarts the queue manager and channel initiator on a different image, the group listener registers with dynamic DNS from this new image. This means that the logical group name (from the connection name field of the channel) connects to the dynamic DNS and is then connected to the same queue manager, now running on a different image. No change was required to the channel definition.

For this type of recovery to occur, the following points must be noted:

- On z/OS, the dynamic DNS runs on one of the z/OS images in the sysplex. If this image were to fail, the dynamic DNS needs to be configured so that there is a secondary name server active in the sysplex, acting as an alternative to the primary name server. Information about primary and secondary dynamic DNS servers can be found in the [OS/390® SecureWay CS IP Configuration](#) manual
- The TCP/IP group listener might have been started on a particular IP address that might not be available on this z/OS image. If so, the listener might need to be started on a different IP address on the new image. If you are using virtual IP addresses, you can configure these to recover on other z/OS images so that no change to the START LISTENER command is required.

Restarting on a different z/OS image with LU 6.2

If you use only LU 6.2 communication protocols, carry out the following procedure to enable network reconnect after automatic restart of a queue manager on a different z/OS image within the sysplex:

- Define each queue manager within the sysplex with a unique subsystem name.
- Define each channel initiator within the sysplex with a unique LUNAME. This is specified in both the queue manager attributes and in the START LISTENER command.

Note: The LUNAME names an entry in the APPC side table, which in turn maps this to the actual LUNAME.

- Set up a shared APPC side table, which is referenced by each z/OS image within the sysplex. This should contain an entry for each channel initiator's LUNAME. See [z/OS MVS Planning: APPC/MVS Management](#) for information about this.
- Set up an APPCPM xx member of SYS1.PARMLIB for each channel initiator within the sysplex to contain an LUADD to activate the APPC side table entry for that channel initiator. These members should be shared by each z/OS image. The appropriate SYS1.PARMLIB member is activated by a z/OS command SET APPC= xx, which is issued automatically during ARM restart of the queue manager (and its channel initiator) on a different z/OS image, as described in the following text.
- Use the LU62ARM queue manager attribute to specify the xx suffix of this SYS1.PARMLIB member for each channel initiator. This causes the channel initiator to issue the required z/OS command SET APPC= xx to activate its LUNAME.

Define your ARM policy so that it restarts the channel initiator only if it fails while its z/OS image stays up; the user ID associated with the XCFAS address space must be authorized to issue the IBM MQ command START CHINIT. Do not restart the channel initiator automatically if its z/OS image also fails, instead use commands in the CSQINP2 and CSQINPX data sets to start the channel initiator and listeners.

Recovering units of work manually

You can manually recover units of work CICS, IMS, RRS, or other queue managers in a queue sharing group. You can use queue manager commands to display the status of the units of work associated with each connection to the queue manager.

This topic contains information about the following subjects:

- [“Displaying connections and threads” on page 567](#)
- [“Recovering CICS units of recovery manually” on page 567](#)
- [“Recovering IMS units of recovery manually” on page 571](#)
- [“Recovering RRS units of recovery manually” on page 572](#)
- [“Recovering units of recovery on another queue manager in the queue sharing group” on page 573](#)

Displaying connections and threads

You can use the `DISPLAY CONN` command to get information about connections to queue managers and their associated units of work. You can display active units of work to see what is currently happening, or to see what needs to be terminated to allow the queue manager to shut down, and you can display unresolved units of work to help with recovery.

Active units of work

To display only active units of work, use

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)
```

Unresolved units of work

An unresolved unit of work, also known as an "in-doubt thread", is one that is in the second pass of the two-phase commit operation. Resources are held in IBM MQ on its behalf. To display unresolved units of work, use

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

External intervention is needed to resolve the status of unresolved units of work. This might only involve starting the recovery coordinator (CICS, IMS, or RRS) or might involve more, as described in the following sections.

Recovering CICS units of recovery manually

Use this topic to understand what happens when the CICS adapter restarts, and then explains how to deal with any unresolved units of recovery that arise.

What happens when the CICS adapter restarts

Whenever a connection is broken, the adapter has to go through a *restart phase* during the *reconnect process*. The restart phase resynchronizes resources. Resynchronization between CICS and IBM MQ enables in-doubt units of work to be identified and resolved.

Resynchronization can be caused by:

- An explicit request from the distributed queuing component
- An implicit request when a connection is made to IBM MQ

If the resynchronization is caused by connecting to IBM MQ, the sequence of events is:

1. The connection process retrieves a list of in-doubt units of work (UOW) IDs from IBM MQ.
2. The UOW IDs are displayed on the console in CSQC313I messages.
3. The UOW IDs are passed to CICS.
4. CICS initiates a resynchronization task (CRSY) for each in-doubt UOW ID.

5. The result of the task for each in-doubt UOW is displayed on the console.

You need to check the messages that are displayed during the connect process:

CSQC313I

Shows that a UOW is in doubt.

CSQC400I

Identifies the UOW and is followed by one of these messages:

- CSQC402I or CSQC403I shows that the UOW was resolved successfully (committed or backed out).
- CSQC404E, CSQC405E, CSQC406E, or CSQC407E shows that the UOW was not resolved.

CSQC409I

Shows that all UOWs were resolved successfully.

CSQC408I

Shows that not all UOWs were resolved successfully.

CSQC314I

Warns that UOW IDs highlighted with a * are not resolved automatically. These UOWs must be resolved explicitly by the distributed queuing component when it is restarted.

Figure 37 on page 568 shows an example set of restart messages displayed on the z/OS console.

```
CSQ9022I +CSQ1 CSQYASCP ' START QMGR' NORMAL COMPLETION
+CSQC323I VICIC1 CSQCQCON 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 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F0E2178D25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F055B2AC25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F07AB56D22
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6EFFD60D425
+CSQC409I VICIC1 CSQCTRUE Resynchronization completed successfully
```

Figure 37. Example restart messages

The total number of CSQC313I messages should equal the total number of CSQC402I plus CSQC403I messages. If the totals are not equal, there are UOWs that the connection process cannot resolve. Those UOWs that cannot be resolved are caused by problems with CICS (for example, a cold start) or with IBM MQ, or by distributing queuing. When these problems have been fixed, you can initiate another resynchronization by disconnecting and then reconnecting.

Alternatively, you can resolve each outstanding UOW yourself using the RESOLVE INDOUBT command and the UOW ID shown in message CSQC400I. You must then initiate a disconnect and a connect to clean

up the *unit of recovery descriptors* in CICS. You need to know the correct outcome of the UOW to resolve UOWs manually.

All messages that are associated with unresolved UOWs are locked by IBM MQ and no Batch, TSO, or CICS task can access them.

If CICS fails and an emergency restart is necessary, *do not* vary the GENERIC APPLID of the CICS system. If you do and then reconnect to IBM MQ, data integrity with IBM MQ cannot be guaranteed. This is because IBM MQ treats the new instance of CICS as a different CICS (because the APPLID is different). In-doubt resolution is then based on the wrong CICS log.

How to resolve CICS units of recovery manually

If the adapter ends abnormally, CICS and IBM MQ build in-doubt lists either dynamically or during restart, depending on which subsystem caused the abend.

Note: If you use the DFH\$INDB sample program to show units of work, you might find that it does not always show IBM MQ UOWs correctly.

When CICS connects to IBM MQ, there might be one or more units of recovery that have not been resolved.

One of the following messages is sent to the console:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E
- CSQC408I

For details of what these messages mean, see the [CICS adapter and Bridge messages](#) messages.

CICS retains details of units of recovery that were not resolved during connection startup. An entry is purged when it no longer appears on the list presented by IBM MQ.

Any units of recovery that CICS cannot resolve must be resolved manually using IBM MQ commands. This manual procedure is rarely used within an installation, because it is required only where operational errors or software problems have prevented automatic resolution. *Any inconsistencies found during in-doubt resolution must be investigated.*

To resolve the units of recovery:

1. Obtain a list of the units of recovery from IBM MQ using the following command:

```
+CSQ1 DISPLAY CONN( * ) WHERE(UOWSTATE EQ UNRESOLVED)
```

You receive the following message:

```

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

```

For CICS connections, the NID consists of the CICS applid and a unique number provided by CICS at the time the syncpoint log entries are written. This unique number is stored in records written to both the CICS system log and the IBM MQ log at syncpoint processing time. This value is referred to in CICS as the *recovery token*.

2. Scan the CICS log for entries related to a particular unit of recovery.

Look for a PREPARE record for the task-related installation where the recovery token field (JCSRMTKN) equals the value obtained from the network ID. The network ID is supplied by IBM MQ in the DISPLAY CONN command output.

The PREPARE record in the CICS log for the units of recovery provides the CICS task number. All other entries on the log for this CICS task can be located using this number.

You can use the CICS journal print utility DFHJUP when scanning the log. For details of using this program, see the *CICS Operations and Utilities Guide*.

3. Scan the IBM MQ log for records with the NID related to a particular unit of recovery. Then use the URID from this record to obtain the rest of the log records for this unit of recovery.

When scanning the IBM MQ log, note that the IBM MQ startup message CSQJ001I provides the start RBA for this session.

The print log records program (CSQ1LOGP) can be used for that purpose.

4. If you need to, do in-doubt resolution in IBM MQ.

IBM MQ can be directed to take the recovery action for a unit of recovery using an IBM MQ [RESOLVE INDOUBT](#) command.

To recover all threads associated with a specific *connection-name*, use the NID(*) option.

The command produces one of the following messages showing whether the thread is committed or backed out:

```

CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED

```

When performing in-doubt resolution, CICS and the adapter are not aware of the commands to IBM MQ to commit or back out units of recovery, because only IBM MQ resources are affected. However, CICS keeps details about the in-doubt threads that could not be resolved by IBM MQ. This information is purged

either when the list presented is empty, or when the list does not include a unit of recovery of which CICS has details.

Recovering IMS units of recovery manually

Use this topic to understand what happens when the IMS adapter restarts, and then explains how to deal with any unresolved units of recovery that arise.

What happens when the IMS adapter restarts

Whenever the connection to IBM MQ is restarted, either following a queue manager restart or an IMS / START SUBSYS command, IMS initiates the following resynchronization process:

1. IMS presents the list of unit of work (UOW) IDs that it believes are in doubt to the IBM MQ IMS adapter one at a time with a resolution parameter of Commit or Backout.
2. The IMS adapter passes the resolution request to IBM MQ and reports the result back to IMS.
3. Having processed all the IMS resolution requests, the IMS adapter gets from IBM MQ a list of all UOWs that IBM MQ still holds in doubt that were initiated by the IMS system. These are reported to the IMS master terminal in message CSQQ008I.

Note: While a UOW is in doubt, any associated IBM MQ message is locked by IBM MQ and is not available to any application.

How to resolve IMS units of recovery manually

When IMS connects to IBM MQ, IBM MQ might have one, or more in-doubt units of recovery that have not been resolved.

If IBM MQ has in-doubt units of recovery that IMS did not resolve, the following message is issued at the IMS master terminal:

```
CSQQ008I nn units of recovery are still in doubt in queue manager qmgr-name
```

If this message is issued, IMS was either cold-started or it was started with an incomplete log tape. This message can also be issued if IBM MQ or IMS terminates abnormally because of a software error or other subsystem failure.

After receiving the CSQQ008I message:

- The connection remains active.
- IMS applications can still access IBM MQ resources.
- Some IBM MQ resources remain locked out.

If the in-doubt thread is not resolved, IMS message queues can start to build up. If the IMS queues fill to capacity, IMS terminates. You must be aware of this potential difficulty, and you must monitor IMS until the in-doubt units of recovery are fully resolved.

Recovery procedure

Use the following procedure to recover the IMS units of work:

1. Force the IMS log closed, using /SWI OLDS, and then archive the IMS log. Use the utility, DFSE-RA10, to print the records from the previous IMS log tape. Type X ' 3730 ' log records indicate a phase-2 commit request and type X ' 38 ' log records indicate an abort request. Record the requested action for the last transaction in each dependent region.
2. Run the DL/I batch job to back out each PSB involved that has not reached a commit point. The process might take some time because transactions are still being processed. It might also lock

up a number of records, which could affect the rest of the processing and the rest of the message queues.

3. Produce a list of the in-doubt units of recovery from IBM MQ using the following command:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

You receive the following message:

```
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)
```

For IMS, the NID consists of the IMS connection name and a unique number provided by IMS. The value is referred to in IMS as the *recovery token*. For more information, see the [IMS documentation](#).

4. Compare the NIDs (IMSID plus OASN in hexadecimal) displayed in the DISPLAY THREAD messages with the OASNs (4 bytes decimal) shown in the DFSERA10 output. Decide whether to commit or back out.
5. Perform in-doubt resolution in IBM MQ with the [RESOLVE INDOUBT](#) command, as follows:

```
RESOLVE INDOUBT( connection-name )
ACTION(COMMIT|BACKOUT)
NID( network-id )
```

To recover all threads associated with *connection-name*, use the NID(*) option. The command results in one of the following messages to indicate whether the thread is committed or backed out:

```
CSQV414I THREAD network-id COMMIT SCHEDULED
CSQV415I THREAD network-id BACKOUT SCHEDULED
```

When performing in-doubt resolution, IMS and the adapter are not aware of the commands to IBM MQ to commit or back out in-doubt units of recovery because only IBM MQ resources are affected.

Recovering RRS units of recovery manually

Use this topic to understand the how to determine if there are in-doubt RRS units of recovery, and how to manually resolve those units of recovery.

When RRS connects to IBM MQ, IBM MQ might have one, or more in-doubt units of recovery that have not been resolved. If IBM MQ has in-doubt units of recovery that RRS did not resolve, one of the following messages is issued at the z/OS console:

- CSQ3011I
- CSQ3013I

- CSQ3014I
- CSQ3016I

Both IBM MQ and RRS provide tools to display information about in-doubt units of recovery, and techniques for manually resolving them.

In IBM MQ, use the DISPLAY CONN command to display information about in-doubt IBM MQ threads. The output from the command includes RRS unit of recovery IDs for those IBM MQ threads that have RRS as a coordinator. This can be used to determine the outcome of the unit of recovery.

Use the RESOLVE INDOUBT command to resolve the IBM MQ in-doubt thread manually. This command can be used to either commit or back out the unit of recovery after you have determined what the correct decision is.

Recovering units of recovery on another queue manager in the queue sharing group

Use this topic to identify, and manually recover units of recovery on other queue managers in a queue sharing group.

If a queue manager that is a member of a queue sharing group fails and cannot be restarted, other queue managers in the group can perform peer recovery, and take over from it. However, the queue manager might have in-doubt units of recovery that cannot be resolved by peer recovery because the final disposition of that unit of recovery is known only to the failed queue manager. These units of recovery are resolved when the queue manager is eventually restarted, but until then, they remain in doubt.

This means that certain resources (for example, messages) might be locked, making them unavailable to other queue managers in the group. In this situation, you can use the DISPLAY THREAD command to display these units of work on the inactive queue manager. If you want to resolve these units of recovery manually to make the messages available to other queue managers in the group, you can use the RESOLVE INDOUBT command.

When you issue the DISPLAY THREAD command to display units of recovery that are in doubt, you can use the QMNAME keyword to specify the name of the inactive queue manager. For example, if you issue the following command:

```
+CSQ1 DISPLAY THREAD(*) TYPE(INDOUBT) QMNAME(QM01)
```

You receive the following messages:

```
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
```

If the queue manager specified is active, IBM MQ does not return information about in-doubt threads, but issues the following message:

```
CSQV435I CANNOT USE QMNAME KEYWORD, QM01 IS ACTIVE
```

Use the IBM MQ command RESOLVE INDOUBT to resolve the in-doubt threads manually. Use the QMNAME keyword to specify the name of the inactive queue manager in the command.

This command can be used to commit or back out the unit of recovery. The command resolves the shared portion of the unit of recovery only; any local messages are unaffected and remain locked until the queue manager restarts, or reconnects to CICS, IMS, or RRS batch.

z/OS IBM MQ and IMS

IBM MQ provides two components to interface with IMS, the IBM MQ - IMS adapter, and the IBM MQ - IMS bridge. These components are commonly called the IMS adapter, and the IMS bridge.

z/OS Operating the IMS adapter

Use this topic to understand how to operate the IMS adapter, which connects IBM MQ to IMS systems.

Note: The IMS adapter does not incorporate any operations and control panels.

This topic contains the following sections:

- [“Controlling IMS connections” on page 574](#)
- [“Connecting from the IMS control region” on page 574](#)
- [“Displaying in-doubt units of recovery” on page 576](#)
- [“Controlling IMS dependent region connections” on page 578](#)
- [“Disconnecting from IMS” on page 580](#)
- [“Controlling the IMS trigger monitor” on page 581](#)

z/OS Controlling IMS connections

Use this topic to understand the IMS operator commands which control and monitor the connection to IBM MQ.

IMS provides the following operator commands to control and monitor the connection to IBM MQ:

/CHANGE SUBSYS

Deletes an in-doubt unit of recovery from IMS.

/DISPLAY OASN SUBSYS

Displays outstanding recovery elements.

/DISPLAY SUBSYS

Displays connection status and thread activity.

/START SUBSYS

Connects the IMS control region to a queue manager.

/STOP SUBSYS

Disconnects IMS from a queue manager.

/TRACE

Controls the IMS trace.

For more information about these commands, see the *IMS/ESA® Operator's Reference* manual for the level of IMS that you are using.

IMS command responses are sent to the terminal from which the command was issued. Authorization to issue IMS commands is based on IMS security.

z/OS Connecting from the IMS control region

Use this topic to understand the mechanisms available to connect from IMS to IBM MQ.

IMS makes one connection from its control region to each queue manager that uses IMS. IMS must be enabled to make the connection in one of these ways:

- Automatically during either:
 - A cold start initialization.
 - A warm start of IMS, if the IBM MQ connection was active when IMS was shut down.
- In response to the IMS command:

```
/START SUBSYS sysid
```

where *sysid* is the queue manager name.

The command can be issued regardless of whether the queue manager is active.

The connection is not made until the first IBM MQ API call to the queue manager is made. Until that time, the IMS command /DIS SUBSYS shows the status as 'NOT CONN'.

The order in which you start IMS and the queue manager is not significant.

IMS cannot re-enable the connection to the queue manager automatically if the queue manager is stopped with a STOP QMGR command, the IMS command /STOP SUBSYS, or an abnormal end. Therefore, you must make the connection by using the IMS command /START SUBSYS.

If an IMS command is seen in the queue manager console log similar to this:

```
MODIFY IMS*,SS*
```

check the IMS master log and ensure that IBM MQ has RACF authority to issue IMS Adapter MODIFY commands.

Initializing the adapter and connecting to the queue manager

The adapter is a set of modules loaded into the IMS control and dependent regions, using the IMS external Subsystem Attach Facility.

This procedure initializes the adapter and connects to the queue manager:

1. Read the subsystem member (SSM) from IMS.PROCLIB. The SSM chosen is an IMS EXEC parameter. There is one entry in the member for each queue manager to which IMS can connect. Each entry contains control information about an IBM MQ adapter.

2. Load the IMS adapter.

Note: IMS loads one copy of the adapter modules for each IBM MQ instance that is defined in the SSM member.

3. Attach the external subsystem task for IBM MQ.
4. Run the adapter with the CTL EXEC parameter (IMSID) as the connection name.

The process is the same whether the connection is part of initialization or a result of the IMS command /START SUBSYS.

If the queue manager is active when IMS tries to make the connection, the following messages are sent:

- to the z/OS console:

```
DFS3613I ESS TCB INITIALIZATION COMPLETE
```

- to the IMS master terminal:

```
CSQQ000I IMS/TM imsid connected to queue manager ssnm
```

When IMS tries to make the connection and *the queue manager is not active*, the following messages are sent to the IMS master terminal each time an application makes an MQI call:

```
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
```

If you get DFS3607I messages when you start the connection to IMS or on system startup, this indicates that the queue manager is not available. To prevent a large number of messages being generated, you must do one of the following:

1. Start the relevant queue manager.
2. Issue the IMS command:

```
/STOP SUBSYS
```

so that IMS does not expect to connect to the queue manager.

If you do neither, a DFS3607I message and the associated CSQQ001I message are issued each time a job is scheduled in the region and each time a connection request to the queue manager is made by an application.

Thread attachment

In an MPP or IFP region, IMS makes a thread connection when the first application program is scheduled into that region, even if that application program does not make an IBM MQ call. In a BMP region, the thread connection is made when the application makes its first IBM MQ call (MQCONN or MQCONNX). This thread is retained for the duration of the region or until the connection is stopped.

For both the message driven and non-message driven regions, the recovery thread cross-reference identifier, *Thread-xref*, associated with the thread is:

```
PSTid + PSBname
```

where:

PSTid

Partition specification table region identifier

PSBname

Program specification block name

You can use connection IDs as unique identifiers in IBM MQ commands, in which case IBM MQ automatically inserts these IDs into any operator message that it generates.

Displaying in-doubt units of recovery

You can display in-doubt units of recovery and attempt to recover them.

The operational steps used to list and recover in-doubt units of recovery in this topic are for relatively simple cases only. If the queue manager ends abnormally while connected to IMS, IMS might commit or back out work without IBM MQ being aware of it. When the queue manager restarts, that work is termed *in doubt*. A decision must be made about the status of the work.

To display a list of in-doubt units of recovery, issue the command:


```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

IBM MQ responds with a message like the following:

```
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
```

For an explanation of the attributes in this message, see the description of the [DISPLAY CONN](#) command.

Recovering in-doubt units of recovery

To recover in-doubt units of recovery, issue this command:

```
+CSQ1 RESOLVE INDOUBT( connection-name ) ACTION(COMMIT|BACKOUT)
NID( net-node.number )
```

where:

connection-name

The IMS system ID.

ACTION

Indicates whether to commit (COMMIT) or back out (BACKOUT) this unit of recovery.

net-node.number

The associated net-node.number.

When you have issued the RESOLVE INDOUBT command, one of the following messages is displayed:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id BACKOUT SCHEDULED
```

Resolving residual recovery entries

At given times, IMS builds a list of residual recovery entries (RREs). RREs are units of recovery about which IBM MQ might be in doubt. They arise in several situations:

- If the queue manager is not active, IMS has RREs that cannot be resolved until the queue manager is active. These RREs are not a problem.
- If the queue manager is active and connected to IMS, and if IMS backs out the work that IBM MQ has committed, the IMS adapter issues message CSQQ010E. If the data in the two systems must be consistent, there is a problem. For information about resolving this problem, see [“Recovering IMS units of recovery manually”](#) on page 571.
- If the queue manager is active and connected to IMS, there might still be RREs even though no messages have informed you of this problem. After the IBM MQ connection to IMS has been established, you can issue the following IMS command to find out if there is a problem:

```
/DISPLAY OASN SUBSYS sysid
```

To purge the RRE, issue one of the following IMS commands:

```
/CHANGE SUBSYS sysid RESET  
/CHANGE SUBSYS sysid RESET OASN nnnn
```

where *nnnn* is the originating application sequence number listed in response to your +CSQ1 DISPLAY command. This is the schedule number of the program instance, giving its place in the sequence of invocations of that program since the last IMS cold start. IMS cannot have two in-doubt units of recovery with the same schedule number.

These commands reset the status of IMS ; they do not result in any communication with IBM MQ.

Controlling IMS dependent region connections

You can control, monitor, and, when necessary, terminate connections between IMS and IBM MQ.

Controlling IMS dependent region connections involves the following activities:

- [Connecting from dependent regions](#)
- [Region error options](#)
- [Monitoring the activity on connections](#)
- [Disconnecting from dependent regions](#)

Connecting from dependent regions

The IMS adapter used in the control region is also loaded into dependent regions. A connection is made from each dependent region to IBM MQ. This connection is used to coordinate the commitment of IBM MQ and IMS work. To initialize and make the connection, IMS does the following:

1. Reads the subsystem member (SSM) from IMS.PROCLIB.

A subsystem member can be specified on the dependent region EXEC parameter. If it is not specified, the control region SSM is used. If the region is never likely to connect to IBM MQ, to avoid loading the adapter, specify a member with no entries.

2. Loads the IBM MQ adapter.

For a batch message program, the load is not done until the application issues its first messaging command. At that time, IMS tries to make the connection.

For a message-processing program region or IMS fast-path region, the attempt is made when the region is initialized.

Region error options

If the queue manager is not active, or if resources are not available when the first messaging command is sent from application programs, the action taken depends on the error option specified on the SSM entry. The options are:

R

The appropriate return code is sent to the application.

Q

The application ends abnormally with abend code U3051. The input message is re-queued.

A

The application ends abnormally with abend code U3047. The input message is discarded.

Monitoring the activity on connections

A thread is established from a dependent region when an application makes its first successful IBM MQ request. You can display information about connections and the applications currently using them by issuing the following command from IBM MQ:

```
+CSQ1 DISPLAY CONN(*) ALL
```

The command produces a message like the following:

```
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)
```

For the control region, *thread-xref* is the special value CONTROL. For dependent regions, it is the PSTid concatenated with the PSBname. *auth-id* is either the user field from the job card, or the ID from the z/OS started procedures table.

For an explanation of the displayed list, see the description of message CSQV402I in the [IBM MQ for z/OS -Nachrichten, -Beendigungs-codes und -Ursachencodes](#) documentation.

IMS provides a display command to monitor the connection to IBM MQ. It shows which program is active on each dependent region connection, the LTERM user name, and the control region connection status. The command is:

```
/DISPLAY SUBSYS name
```

The status of the connection between IMS and IBM MQ is shown as one of:

```
CONNECTED  
NOT CONNECTED  
CONNECT IN PROGRESS  
STOPPED  
STOP IN PROGRESS  
INVALID SUBSYSTEM NAME= name  
SUBSYSTEM name NOT DEFINED BUT RECOVERY OUTSTANDING
```

The thread status from each dependent region is one of the following:

```
CONN  
CONN, ACTIVE (includes LTERM of user)
```

Disconnecting from dependent regions

To change values in the SSM member of IMS.PROCLIB, you disconnect a dependent region. To do this, you must:

1. Issue the IMS command:

```
/STOP REGION
```

2. Update the SSM member.
3. Issue the IMS command:

```
/START REGION
```

Disconnecting from IMS

The connection is ended when either IMS or the queue manager terminates. Alternatively, the IMS master terminal operator can explicitly break the connection.

To terminate the connection between IMS and IBM MQ, use the following IMS command:

```
/STOP SUBSYS sysid
```

The command sends the following message to the terminal that issued it, typically the master terminal operator (MTO):

```
DFS058I STOP COMMAND IN PROGRESS
```

The IMS command:

```
/START SUBSYS sysid
```

is required to reestablish the connection.

Note: The IMS command /STOP SUBSYS is not completed if an IMS trigger monitor is running.

Controlling the IMS trigger monitor

You can use the CSQQTRMN transaction to stop, and start the IMS trigger monitor.

The IMS trigger monitor (the CSQQTRMN transaction) is described in the [Setting up the IMS trigger monitor](#).

To control the IMS trigger monitor see:

- [Starting CSQQTRMN](#)
- [Stopping CSQQTRMN](#)

Starting CSQQTRMN

1. Start a batch-oriented BMP that runs the program CSQQTRMN for each initiation queue you want to monitor.
2. Modify your batch JCL to add a DDname of CSQQUT1 that points to a data set containing the following information:

```
QMGRNAME=q_manager_name      Comment: queue manager name
INITQUEUEUENAME=init_q_name   Comment: initiation queue name
LTERM=lterm                   Comment: LTERM to remove error messages
CONSOLEMESSAGES=YES          Comment: Send error messages to console
```

where:

q_manager_name	The name of the queue manager (if this is blank, the default nominated in CSQQDEFV is assumed)
init_q_name	The name of the initiation queue to be monitored
lterm	The IMS LTERM name for the destination of error messages (if this is blank, the default value is MASTER).
CONSOLEMESSAGES=YES	Requests that messages sent to the nominated IMS LTERM are also sent to the z/OS console. If this parameter is omitted or misspelled, the default is NOT to send messages to the console.

3. Add a DD name of CSQQUT2 if you want a printed report of the processing of CSQQUT1 input.

Note:

1. The data set CSQQUT1 is defined with LRECL=80. Other DCB information is taken from the data set. The DCB for data set CSQQUT2 is RECFM=VBA and LRECL=125.

2. You can put only one keyword on each record. The keyword value is delimited by the first blank following the keyword; this means that you can include comments. An asterisk in column 1 means that the whole input record is a comment.
3. If you misspell either of the QMGRNAME or LTERM keywords, CSQQTRMN uses the default for that keyword.
4. Ensure that the subsystem is started in IMS (by the /START SUBSYS command) before submitting the trigger monitor BMP job. If it is not started, your trigger monitor job terminates with abend code U3042.

Stopping CSQQTRMN

Once started, CSQQTRMN runs until either the connection between IBM MQ and IMS is broken due to one of the following events:

- the queue manager ending
 - IMS ending
- or a z/OS STOP **jobname** command is entered.

Controlling the IMS bridge

Use this topic to understand the IMS commands that you can use to control the IMS bridge.

There are no IBM MQ commands to control the IBM MQ-IMS bridge. However, you can stop messages being delivered to IMS in the following ways:

- For non-shared queues, by using the ALTER QLOCAL(xxx) GET(DISABLED) command for all bridge queues.
- For clustered queues, by using the SUSPEND QMGR CLUSTER(xxx) command. This is effective only when another queue manager is also hosting the clustered bridge queue.
- For clustered queues, by using the SUSPEND QMGR FACILITY(IMSBRIDGE) command. No further messages are sent to IMS, but the responses for any outstanding transactions are received from IMS.

To start sending messages to IMS again, issue the RESUME QMGR FACILITY(IMSBRIDGE) command.

You can also use the MQSC command DISPLAY SYSTEM to display whether the bridge is suspended.

See [MQSC commands](#) for details of these commands.

For further information see:

- [“Starting and stopping the IMS bridge” on page 582](#)
- [“Controlling IMS connections” on page 583](#)
- [Controlling bridge queues](#)
- [“Resynchronizing the IMS bridge” on page 584](#)
- [Working with tpipe names](#)
- [Deleting messages from IMS](#)
- [Deleting tpipes](#)
- [“IMS Transaction Expiration” on page 586](#)

Starting and stopping the IMS bridge

Start the IBM MQ bridge by starting OTMA. Either use the IMS command:

```
/START OTMA
```

or start it automatically by specifying OTMA=YES in the IMS system parameters. If OTMA is already started, the bridge starts automatically when queue manager startup has completed. An IBM MQ event message is produced when OTMA is started.

Use the IMS command:

```
/STOP OTMA
```

to stop OTMA communication. When this command is issued, an IBM MQ event message is produced.

Controlling IMS connections

IMS provides these operator commands to control and monitor the connection to IBM MQ:

/DEQUEUE TMEMBER *tmember* TPIPE *tpipe*

Removes messages from a Tpipe. Specify PURGE to remove all messages or PURGE1 to remove the first message only.

/DISPLAY OTMA

Displays summary information about the OTMA server and clients, and client status.

/DISPLAY TMEMBER *name*

Displays information about an OTMA client.

/DISPLAY TRACE TMEMBER *name*

Displays information about what is being traced.

/SECURE OTMA

Sets security options.

/START OTMA

Enables communications through OTMA.

/START TMEMBER *tmember* TPIPE *tpipe*

Starts the named Tpipe.

/STOP OTMA

Stops communications through OTMA.

/STOP TMEMBER *tmember* TPIPE *tpipe*

Stops the named Tpipe.

/TRACE

Controls the IMS trace.

For more information about these commands, see the *IMS/ESA Operators Reference* manual for the level of IMS that you are using.

IMS command responses are sent to the terminal from which the command was issued. Authorization to issue IMS commands is based on IMS security.

Controlling bridge queues

To stop communicating with the queue manager with XCF member name *tmember* through the bridge, issue the following IMS command:

```
/STOP TMEMBER tmember TPIPE ALL
```

To resume communication, issue the following IMS command:

```
/START TMEMBER tmember TPIPE ALL
```

The Tpipes for a queue can be displayed using the MQ DISPLAY QUEUE command.

To stop communication with the queue manager on a single Tpipe, issue the following IMS command:

```
/STOP TMEMBER tmember TPIPE tpipe
```

One or two Tpipes are created for each active bridge queue, so issuing this command stops communication with the IBM MQ queue. To resume communication, use the following IMS command :

```
/START TMEMBER tmember TPIPE tpipe
```

Alternatively, you can alter the attributes of the IBM MQ queue to make it get inhibited.

Resynchronizing the IMS bridge

The IMS bridge is automatically restarted whenever the queue manager, IMS, or OTMA are restarted.

The first task undertaken by the IMS bridge is to resynchronize with IMS. This involves IBM MQ and IMS checking sequence numbers on every synchronized Tpipe. A synchronized Tpipe is used when persistent messages are sent to IMS from an IBM MQ - IMS bridge queue using commit mode zero (commit-then-send).

If the bridge cannot resynchronize with IMS, the IMS sense code is returned in message CSQ2023E and the connection to OTMA is stopped. If the bridge cannot resynchronize with an individual IMS Tpipe, the IMS sense code is returned in message CSQ2025E and the Tpipe is stopped. If a Tpipe has been cold started, the recoverable sequence numbers are automatically reset to 1.

If the bridge discovers mismatched sequence numbers when resynchronizing with a Tpipe, message CSQ2020E is issued. Use the IBM MQ command RESET TPIPE to initiate resynchronization with the IMS Tpipe. You need to provide the XCF group and member name, and the name of the Tpipe; this information is provided by the message.

You can also specify:

- A new recoverable sequence number to be set in the Tpipe for messages sent by IBM MQ, and to be set as the partner's receive sequence number. If you do not specify this, the partner's receive sequence number is set to the current IBM MQ send sequence number.
- A new recoverable sequence number to be set in the Tpipe for messages received by IBM MQ, and to be set as the partner's send sequence number. If you do not specify this, the partner's send sequence number is set to the current IBM MQ receive sequence number.

If there is an unresolved unit of recovery associated with the Tpipe, this is also notified in the message. Use the IBM MQ command RESET TPIPE to specify whether to commit the unit of recovery, or back it out. If you commit the unit of recovery, the batch of messages has already been sent to IMS, and is deleted from the bridge queue. If you back the unit of recovery out, the messages are returned to the bridge queue, to be later sent to IMS.

Commit mode 1 (send-then-commit) Tpipes are not synchronized.

Considerations for Commit mode 1 transactions

In IMS, commit mode 1 (CM1) transactions send their output replies before sync point.

A CM1 transaction might not be able to send its reply, for example because:

- The Tpipe on which the reply is to be sent is stopped
- OTMA is stopped
- The OTMA client (that is, the queue manager) has gone away
- The reply-to queue and dead-letter queue are unavailable

For these reasons, the IMS application sending the message pseudo-abends with code U0119. The IMS transaction and program are not stopped in this case.

These reasons often prevent messages being sent into IMS, as well as replies being delivered from IMS. A U0119 abend can occur if:

- The Tpipe, OTMA, or the queue manager is stopped while the message is in IMS
- IMS replies on a different Tpipe to the incoming message, and that Tpipe is stopped
- IMS replies to a different OTMA client, and that client is unavailable.

Whenever a U0119 abend occurs, both the incoming message to IMS and the reply messages to IBM MQ are lost. If the output of a CMO transaction cannot be delivered for any of these reasons, it is queued on the Tpipe within IMS.

Working with tpipe names

Many of the commands used to control the IBM MQ - IMS bridge require the *tpipe* name. Use this topic to understand how you can find further details of the tpipe name.

You need *tpipe* names for many of the commands that control the IBM MQ - IMS bridge. You can get the tpipe names from DISPLAY QUEUE command and note the following points:

- tpipe names are assigned when a local queue is defined
- a local queue is given two tpipe names, one for sync and one for non-sync
- tpipe names will not be known to IMS until after some communication between IMS and IBM MQ specific to that particular local queue takes place
- For a tpipe to be available for use by the IBM MQ - IMS bridge its associated queue must be assigned to a Storage Class that has the correct XCF group and member name fields completed

Deleting messages from IMS

A message that is destined for IBM MQ through the IMS bridge can be deleted if the Tmember/Tpipe is stopped. To delete one message for the queue manager with XCF member name *tmember*, issue the following IMS command:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE1
```

To delete all the messages on the Tpipe, issue the following IMS command:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE
```

Deleting tpipes

You cannot delete IMS tpipes yourself. They are deleted by IMS at the following times:

- Synchronized tpipes are deleted when IMS is cold started.

- Non-synchronized tpipes are deleted when IMS is restarted.

IMS Transaction Expiration

An expiration time is associated with a transaction; any IBM MQ message can have an expiration time associated with it. The expiration interval is passed from the application, to IBM MQ, using the MQMD.Expiry field. The time is the duration of a message before it expires, expressed as a value in tenths of a second. An attempt to perform the MQGET of a message, later than it has expired, results in the message being removed from the queue and expiry processing performed. The expiration time decreases as a message flows between queue managers on an IBM MQ network. When an IMS message is passed across the IMS bridge to OTMA, the remaining message expiry time is passed to OTMA as a transaction expiration time.

If a transaction has an expiration time specified, OTMA expires the input transactions in three different places in IMS:

- input message receiving from XCF
- input message enqueueing time
- application GU time

No expiration is performed after the GU time.

The transaction EXPRTIME can be provided by:

- IMS transaction definition
- IMS OTMA message header
- IMS DFSINSX0 user exit
- IMS CREATE or UPDATE TRAN commands

IMS indicates that it has expired a transaction by abending a transaction with 0243, and issuing a message. The message issued is either DFS555I in the non-shared-queues environment, or DFS2224I in the shared-queues environment.

z/OS

Operating Advanced Message Security on z/OS

The Advanced Message Security address space accepts commands using the z/OS MODIFY command.

Procedure

- Modify Advanced Message Security on z/OS.

To enter commands for the Advanced Message Security (AMS) address space, use the z/OS MODIFY command.

For example:

```
F qmgrAMSM, cmd
```

where *qmgr* is the prefix of the started task name.

The following table describes the MODIFY commands that are accepted:

Table 29. Advanced Message Security address space MODIFY commands		
Command	Option	Description
DISPLAY		Display version information
REFRESH	KEYRING POLICY ALL	Refresh the key ring certificates, security policies, or both.

Table 29. Advanced Message Security address space MODIFY commands (continued)		
Command	Option	Description
SMFAUDIT	SUCCESS FAILURE ALL	Set whether SMF auditing is required when AMS successfully protects or unprotects messages, when AMS fails to protect or unprotect messages, or both.
SMFTYPE	0 - 255	Set the SMF record type to be generated when AMS protects or unprotects messages. To disable SMF auditing specify a record type of 0.

Note: To specify an option it must be separated by a comma. For example:

```
F qmgrAMSM,REFRESH KEYRING
F qmgrAMSM,SMFAUDIT ALL
F qmgrAMSM,SMFTYPE 180
```

- Refresh Advanced Message Security on z/OS.

Changes that are made effective by issuing the **REFRESH** command apply to applications that issue MQOPEN after the **REFRESH** command has completed. Existing applications that have a queue open, continue to use the options from when the application opened the queue. To use the new values, the application has to close and reopen the queue.

- Start and stop AMS on z/OS.

You do not need to enter a command to start or stop the Advanced Message Security address space. The AMS address space is started automatically when the queue manager is started if AMS has been enabled with the **SPLCAP** parameter of CSQ6SYSP, and is stopped when the queue manager is stopped.

IBM MQ Internet Pass-Thru verwalten

In diesem Abschnitt wird die Verwaltung von IBM MQ Internet Pass-Thru (MQIPT) beschrieben.

Konfigurieren Sie MQIPT durch Änderung an der `mqipt.conf` Konfigurationsdatei, wie beschrieben in Konfigurieren IBM MQ Internet Pass-Thru. Verwenden Sie den Befehl **mqiptAdmin**, um MQIPT zu verwalten, einschließlich der Aktualisierung von MQIPT, damit Konfigurationsänderungen wirksam werden, ohne MQIPT erneut zu starten. Informationen zur Verwaltung von MQIPT mit dem Befehl **mqiptAdmin** finden Sie unter „MQIPT über die Befehlszeile verwalten“ auf Seite 590.

MQIPT starten und stoppen

MQIPT kann über die Befehlszeile oder beim Start des System automatisch gestartet werden. Sie können MQIPT mit dem Befehl **mqiptAdmin** stoppen.

MQIPT über die Befehlszeile starten

MQIPT wird in einem Installationsverzeichnis installiert, z. B.:

- **Windows** C:\MQIPT auf Windows -Systemen mit ausführbaren Scripts in C:\MQIPT\bin
- **Linux** **AIX** /opt/mqipt auf AIX and Linux -Systemen mit ausführbaren Scripts in /opt/mqipt/bin

MQIPT verwendet außerdem ein Ausgangsverzeichnis, das die Konfigurationsdatei `mqipt.conf` und alle Dateien enthält, die von MQIPT ausgegeben werden, wenn sie ausgeführt werden. Die folgenden Unterverzeichnisse des Ausgangsverzeichnisses von MQIPT werden automatisch erstellt, wenn MQIPT das erste Mal aufgerufen wird:

- Ein `errors` -Verzeichnis, in dem alle First Failure Support Technology (FFST) und Tracedateien geschrieben werden.
- Ein `logs` -Verzeichnis, in dem das Verbindungsprotokoll beibehalten wird.

Die Benutzer-ID, mit der MQIPT ausgeführt wird, muss über die Berechtigung zum Erstellen dieser Verzeichnisse verfügen oder die Verzeichnisse müssen bereits vorhanden sein und die Benutzer-ID muss zum Erstellen, Lesen und Schreiben von Dateien in diesen Verzeichnissen berechtigt sein. Wenn Sie außerdem eine Java security manager -Richtlinie verwenden, muss die Sicherheitsrichtlinie die erforderlichen Berechtigungen für diese Verzeichnisse erteilen. Weitere Informationen zu den Einstellungen von Security Manager-Richtlinien finden Sie unter [Java security manager](#).

Sie können das Installationsverzeichnis als Ausgangsverzeichnis verwenden. In diesem Fall müssen Sie sicherstellen, dass die Benutzer-ID, mit der MQIPT ausgeführt wird, über die entsprechenden Berechtigungen verfügt und alle Security Manager-Richtlinien korrekt konfiguriert sind.

Zum Starten von MQIPT verwenden Sie den Befehl `mqipt`, der sich im Verzeichnis `bin` des MQIPT-Installationsverzeichnisses befindet. Mit dem folgenden Befehl wird beispielsweise eine Instanz von MQIPT gestartet, die das Verzeichnis `C:\mqiptHome` als Ausgangsverzeichnis verwendet:

```
mqipt C:\mqiptHome
```

Weitere Informationen zum Befehl `mqipt` finden Sie im Abschnitt [mqipt \(MQIPT starten\)](#).




Mit dem Befehl `mqipt` können Sie einen Namen für die MQIPT-Instanz angeben, die gestartet wird. Mit dem Namen der MQIPT-Instanz werden lokale Instanzen von MQIPT mit dem Befehl `mqiptAdmin` verwaltet, ohne dass die Verwendung eines Befehlsports erforderlich ist. Wenn dieser Parameter nicht angegeben ist, wird der Name des MQIPT-Ausgangsverzeichnisses als Name der MQIPT-Instanz verwendet.

In Konsolennachrichten wird der Status von MQIPT angezeigt. Wenn ein Fehler auftritt, lesen Sie die Informationen im Abschnitt [Fehlerbehebung IBM MQ Internet Pass-Thru](#). Die folgenden Nachrichten sind ein Beispiel für die Ausgabe bei einem erfolgreichen Start von MQIPT:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is C:\mqiptHome
MQCPI021 Password checking has been enabled on the command port
MQCPI144 MQ Advanced capabilities not enabled
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1414 is starting and will forward messages to :
MQCPI034 ...examplehost(1414)
MQCPI035 ...using MQ protocols
MQCPI057 ...trace level 5 enabled
MQCPI078 Route 1414 ready for connection requests
```

MQIPT automatisch starten

Sie können MQIPT als Systemservice installieren, der beim Systemstart automatisch gestartet wird. Verwenden Sie den Befehl `mqiptService`, um den MQIPT-Service zu installieren und zu deinstallieren.

-  Auf Windows -Systemen installiert der `mqiptService` -Befehl MQIPT als Windows -Dienst.
-   Auf AIX and Linux -Systemen installiert der Befehl `mqiptService` MQIPT als System V-Initialisierungsservice, der beim Booten des Systems gestartet wird. On Linux systems which do not support System V init, use another method, such as `systemd`, to manage MQIPT as a service.

Beim Start des MQIPT-Service werden alle aktiven MQIPT-Routen gestartet. Wenn der Service gestoppt wird, werden alle Routen sofort beendet.

Sie können nur einen MQIPT-Service auf einem System installieren, auch wenn es mehrere Installationen von MQIPT auf dem System gibt.

Weitere Informationen zum Befehl **mqiptService** finden Sie unter [mqiptService \(MQIPT-Service verwalten\)](#).

MQIPT stoppen

Sie können MQIPT mithilfe des Befehls **mqiptAdmin** mit dem Parameter **-stop** stoppen.

Beispielsweise wird mit dem folgenden Befehl eine Instanz von MQIPT mit dem Namen `mqipt1` gestoppt, die lokal mit der gleichen Benutzer-ID wie der Befehl **mqiptAdmin** ausgeführt wird:

```
mqiptAdmin -stop -n ipt1
```

Der Befehl **mqiptAdmin** stellt mit einer der folgenden Methoden eine Verbindung zur aktiven Instanz von MQIPT her, damit diese verwaltet werden kann:

- über eine lokale Instanz von MQIPT ohne die Verwendung des Befehlsports.
- über eine Netzverbindung zu einem Befehlsport.

Der ferne Systemabschluss muss aktiviert werden, indem die Eigenschaft **RemoteShutDown** auf `true` gesetzt wird, bevor der Befehl **mqiptAdmin** zum Stoppen von MQIPT durch das Senden eines Befehls an einen Befehlsport verwendet wird.

Weitere Informationen zur Verwaltung von MQIPT mit dem Befehl **mqiptAdmin** finden Sie unter „[MQIPT über die Befehlszeile verwalten](#)“ auf Seite 590.

Schlüssel für die Kennwortverschlüsselung angeben

Wenn die MQIPT -Konfiguration Kennwörter enthält, die mit einem anderen Verschlüsselungsschlüssel als dem Standardschlüssel verschlüsselt sind, müssen Sie den Kennwortverschlüsselungsschlüssel in einer Datei angeben, die MQIPT beim Start lesen kann.

Schlüsseldatei für die Kennwortverschlüsselung

Kennwörter, die verschlüsselt sind, damit sie gespeichert und von MQIPT verwendet werden können, können mithilfe eines von Ihnen bereitgestellten Verschlüsselungsschlüssel verschlüsselt werden. Wenn Sie keinen Verschlüsselungsschlüssel bereitstellen, wird der Standardverschlüsselungsschlüssel verwendet. Sie müssen keinen Kennwortverschlüsselungsschlüssel angeben, aus Sicherheitsgründen wird dies allerdings empfohlen. Wenn Sie keinen eigenen Verschlüsselungsschlüssel angeben, wird der Standardverschlüsselungsschlüssel verwendet.

Wenn Sie einen Kennwortverschlüsselungsschlüssel bereitstellen, muss dieser in einer Datei gespeichert werden, auf die mit dem Befehl **mqiptPW** (zum Verschlüsseln von Kennwörtern) und mit MQIPT zugegriffen werden kann. Die Inhalte der Datei unterliegen nur der Einschränkung, dass sie mindestens ein Zeichen und nur eine Textzeile enthalten.

Anmerkung: Sie müssen sicherstellen, dass in der Schlüsseldatei für die Kennwortverschlüsselung die entsprechenden Dateiberechtigungen festgelegt sind, damit nur berechtigte Benutzer den Verschlüsselungsschlüssel lesen können. Nur der Benutzer, der den Befehl **mqiptPW** ausführt, und der Benutzer, unter dem MQIPT ausgeführt wird, benötigen die Berechtigung zum Lesen des Kennwortverschlüsselungsschlüssels.

Alle gespeicherten Kennwörter für eine Instanz von MQIPT werden mit dem gleichen Kennwortverschlüsselungsschlüssel verschlüsselt und entschlüsselt. Daher benötigen Sie für jede MQIPT-Installation nur eine einzelne Schlüsseldatei für die Kennwortverschlüsselung.

Wenn der Kennwortverschlüsselungsschlüssel für eine MQIPT-Installation geändert wird, müssen alle verschlüsselten Kennwörter mithilfe des neuen Verschlüsselungsschlüssels erneut verschlüsselt werden.

Starten MQIPT

Der Standardname der Kennwortverschlüsselungsschlüsseldatei ist `MQIPT_HOME_DIR/mqipt_cred.key`. Hierbei steht `MQIPT_HOME_DIR` für das Verzeichnis, in dem die `mqipt.conf` -Kon-

figurationsdatei gespeichert ist. Wenn Sie MQIPT als Service ausführen möchten, der automatisch gestartet wird, müssen Sie die Schlüsseldatei für die Kennwortverschlüsselung mit dem Standardnamen erstellen.

Wenn die Schlüsseldatei für die Kennwortverschlüsselung mit einem anderen Namen als dem Standardnamen erstellt wird, muss der Name der Datei beim Start von MQIPT bereitgestellt werden. Der Name der Schlüsseldatei für die Kennwortverschlüsselung kann mit einer der folgenden Methoden angegeben werden (nach Empfehlung geordnet):

1. Mit dem Parameter **-sf** im Befehl **mqipt** zum Starten von MQIPT.
2. Umgebungsvariable `MQS_MQIPTCRED_KEYFILE`.
3. Eigenschaft `com.ibm.mq.ipc.cred.keyfile` Java.

Wenn kein Name für die Schlüsseldatei für die Kennwortverschlüsselung bereitgestellt wird, wird der Standarddateiname verwendet, wenn die Datei vorhanden ist. Wenn die Standardschlüsseldatei für die Kennwortverschlüsselung nicht vorhanden ist, wird der standardmäßige Kennwortverschlüsselungsschlüssel verwendet.

MQIPT über die Befehlszeile verwalten

Mit dem Befehl **mqiptAdmin** in der Befehlszeile können Sie MQIPT verwalten.

Mit dem Befehl **mqiptAdmin** können Sie die folgenden Verwaltungsfunktionen ausführen:

- Auflistung der aktiven lokalen Instanzen von MQIPT.
- Aktualisierung einer Instanz von MQIPT, nachdem Änderungen an der Konfigurationsdatei vorgenommen wurden.
- Stoppen einer Instanz von MQIPT.

Der Befehl **mqiptAdmin** befindet sich im Unterverzeichnis `bin` des Installationsverzeichnisses von MQIPT.

Der Befehl **mqiptAdmin** stellt mit einer der folgenden Methoden eine Verbindung zur aktiven Instanz von MQIPT her, damit diese verwaltet werden kann:

- über eine Netzverbindung zu einem Befehlsport.
- über eine lokale Instanz von MQIPT ohne die Verwendung des Befehlsports.

Der Befehl **mqiptAdmin** ist mit vorherigen Versionen von MQIPT kompatibel, Sie können mit dem Befehl allerdings keine Versionen von MQIPT verwalten, die eine höhere Version als der Befehl **mqiptAdmin** haben. In einer Umgebung mit verschiedenen Versionen von MQIPT müssen Sie die aktuellste Version des Befehls **mqiptAdmin** verwenden.

Weitere Informationen zur Syntax des Befehls **mqiptAdmin** finden Sie unter [mqiptAdmin \(MQIPT verwalten\)](#).

Lokale Verwaltung ohne Befehlsport

Lokale Instanzen von MQIPT können ohne Verwendung eines Befehlsports verwaltet werden. Mit der lokalen Verwaltung können Sie MQIPT nur dann mit dem Befehl **mqiptAdmin** verwalten, wenn er auf dem gleichen System wie die MQIPT-Instanz ausgeführt wird, die Sie verwalten möchten.

Damit **mqiptAdmin** zur Verwaltung einer lokalen Instanz von MQIPT ohne Verwendung des Befehlsports berechtigt ist, muss die MQIPT-Instanz auf dem gleichen System und mit der gleichen Benutzer-ID wie **mqiptAdmin** ausgeführt werden. Unter AIX and Linux kann **mqiptAdmin** als `root` ausgeführt werden.

Die lokale Verwaltung ist standardmäßig aktiviert. Zum Inaktivieren der lokalen Verwaltung verwenden Sie die Konfigurationseigenschaft **LocalAdmin**. Weitere Informationen zur Eigenschaft **LocalAdmin** finden Sie unter [LocalAdmin](#).

Um lokale Instanzen von MQIPT verwalten zu können, müssen Sie jeder Instanz einen Namen zuordnen. Sie können einer Instanz von MQIPT einen Namen zuordnen, indem Sie den Parameter **-n** verwenden, wenn Sie MQIPT mit dem Befehl **mqipt** starten. Wenn Sie beim Start von MQIPT keinen Namen angeben,

wird der Name des Ausgangsverzeichnisses für die MQIPT-Instanz verwendet. Beispielsweise wird mit folgendem Befehl MQIPT gestartet und der Name `ipt1` wird der Instanz zugeordnet:

```
mqipt /opt/mqipt1 -n ipt1
```

Sobald die Instanz einen Namen hat, können Sie diese durch die Angabe des Namens im Befehl **mqiptAdmin** mit dem Parameter **-n** verwalten. Beispielsweise wird mit folgendem Befehl die lokale Instanz von MQIPT mit dem Namen `ipt1` gestoppt:

```
mqiptAdmin -stop -n ipt1
```

Sie können alle lokalen aktiven Instanzen von MQIPT auflisten, die mit dem Befehl **mqiptAdmin** ohne die Verwendung eines Befehlsports verwaltet werden können, indem Sie den Befehl **mqiptAdmin** mit dem Parameter **-list** verwenden. Beispielsweise werden mit dem folgenden Befehl alle lokalen aktiven Instanzen von MQIPT aufgeführt, zu deren Verwaltung der Benutzer, der den Befehl **mqiptAdmin** gestartet hat, berechtigt ist:

```
mqiptAdmin -list
```

Verwaltung mit einem Befehlsport

Sie können MQIPT mit einem Befehlsport konfigurieren, der nicht gesichert ist, und einem Befehlsport, der mit TLS gesichert ist. Mit diesen Befehlsports können Sie MQIPT wie jeden Benutzer verwalten, der sich im gleichen System wie die MQIPT-Instanz, die Sie verwalten möchten, oder in einem fernen System befindet.

In früheren Versionen von MQIPT wurden nur Verwaltungsbefehle akzeptiert, die an den nicht gesicherten Befehlsport ausgegeben wurden.

Anmerkung: Verbindungen zum nicht gesicherten Befehlsport sind nicht verschlüsselt und daher können Daten, die über das Netz an den nicht gesicherten Befehlsport gesendet werden (einschließlich dem MQIPT-Zugriffskennwort) für andere Benutzer im Netz sichtbar sein.

Damit MQIPT auf dem Befehlsport für Befehle empfangsbereit ist, die vom Befehl **mqiptAdmin** ausgegeben werden, muss für die Eigenschaften **CommandPort** oder **SSLCommandPort** im globalen Teil der Konfigurationsdatei `mqipt.conf` ein Wert angegeben werden.

Überprüfen Sie die Sicherheitsaspekte im Abschnitt [Weitere Sicherheitsaspekte](#), bevor Sie einen der MQIPT-Befehlsports aktivieren. Sie können auch die [Authentifizierung für Befehle](#) aktivieren, die von den Befehlsports empfangen werden. Weitere Informationen zur Authentifizierung von Befehlsports finden Sie unter [„Authentifizierung des Befehlsports“](#) auf Seite 595.

Um eine Instanz von MQIPT mithilfe eines Befehlsports zu verwalten, geben Sie die Netzadresse des Hosts, auf dem MQIPT ausgeführt wird, sowie die Portnummer des Befehls als Parameter für den Befehl **mqiptAdmin** an. Um beispielsweise die MQIPT-Instanz zu aktualisieren, die unter `mqipt.example.com` ausgeführt wird und deren nicht gesicherter Befehlsport so konfiguriert ist, dass er auf Port 1890 empfangsbereit ist, geben Sie den folgenden Befehl aus:

```
mqiptAdmin -refresh -r mqipt.example.com:1890
```

Wenn Sie keinen Hostnamen und keine Portnummer angeben, versucht der Befehl **mqiptAdmin**, eine Verbindung zu `localhost` an Port 1881 herzustellen.

Weitere Informationen zur Verwaltung von MQIPT mithilfe des TLS-Befehlsports finden Sie im Abschnitt [„MQIPT über den TLS-Befehlsport verwalten“](#) auf Seite 591.

MQIPT über den TLS-Befehlsport verwalten



MQIPT kann für die Verwendung eines TLS-Befehlsports konfiguriert werden, der für Verwaltungsbefehle empfangsbereit ist, die vom Befehl **mqiptAdmin** ausgegeben werden. Durch den TLS-Befehlsport werden sensible Daten wie das Kennwort für den MQIPT-Zugriff im Netz zwischen **mqiptAdmin** und MQIPT

geschützt. Konfigurieren Sie mit diesem Verfahren den TLS-Befehlsport und verwalten Sie MQIPT über den TLS-Befehlsport.

Informationen zu diesem Vorgang

Der TLS-Befehlsport muss mit einem Serverzertifikat konfiguriert werden, das entweder in einem PKCS #12 -Keystore oder in Verschlüsselungshardware gespeichert ist, die die PKCS #11 Cryptographic Token Interface unterstützt. Das Serverzertifikat des Befehlsports wird beim TLS-Handshake an den Befehl **mqiptAdmin** gesendet. In dieser Task wird vorausgesetzt, dass Sie ein neues Serverzertifikat von einer vertrauenswürdigen Zertifizierungsstelle anfordern und dieses Zertifikat in einer Datei an Sie zurückgegeben wird. Der Befehl **mqiptAdmin** überprüft das Zertifikat des Befehlsports mithilfe des CA-Zertifikats der Zertifizierungsstelle, die das Serverzertifikat unterzeichnet hat. Das CA-Zertifikat muss in einem PKCS #12 -Keystore gespeichert werden, auf den der Befehl **mqiptAdmin** zugreifen kann.

Die Clientzertifikatsauthentifizierung wird vom TLS-Befehlsport nicht unterstützt. Informationen zum Aktivieren der Authentifizierung für Verwaltungsbefehle, die von einem Befehlsport ausgegeben werden, finden Sie im Abschnitt „Authentifizierung des Befehlsports“ auf Seite 595.

Diese Prozedur beschreibt, wie die Keystores und digitalen Zertifikate, die für die Verwendung des TLS-Befehlsports erforderlich sind, mit dem Befehl   **mqiptKeytool** verwaltet werden. Weitere Informationen zur Verwaltung von Keystores, die MQIPT verwendet, finden Sie unter [MQIPT -Keystores verwalten](#).

Vorgehensweise

1. Führen Sie die folgenden Schritte aus, um den TLS-Befehlsport für die Instanz von MQIPT zu konfigurieren.

a) Erstellen Sie ein Paar aus öffentlichem und privatem Schlüssel und ein zugehöriges TLS-Befehlsportserverzertifikat in einem PKCS #12 -Keystore.



Geben Sie den folgenden Befehl ein, um den Keystore zu erstellen, der das Serverzertifikat für den TLS-Befehlsport enthält:

```
mqiptKeytool -genkeypair -keystore filename -storetype pkcs12 -storepass password
             -dname distinguished_name -alias label
             -keyalg key_algorithm -keysize key_size -sigalg sig_algorithm
```

Dabei gilt:

-keystore *dateiname*

Gibt den Namen des Keystores an

-storepass *Kennwort*

Gibt das Keystore-Kennwort an

-alias *Bezeichnung*

Gibt die Zertifikatsbezeichnung an.

-keyalg *Schlüsselalgorithmus*

Gibt den Algorithmus an, der zum Erstellen des Schlüsselpaars verwendet wird

-keysize *Schlüsselgröße*

Gibt die Schlüsselgröße an.

-sigalg *Algorithmus*

Gibt den Algorithmus an, der zum Signieren des Zertifikats verwendet wird.

-dname *definiertes_name*

Gibt den definierten X.500-Namen in doppelte Anführungszeichen an.

b) Erstellen Sie eine Zertifikatsanforderung für das von der Zertifizierungsstelle unterzeichnete Serverzertifikat für den TLS-Befehlsport.



Geben Sie den folgenden Befehl ein, um eine Zertifikatsanforderung zu erstellen:


```
mqiptykeytool -certreq -keystore filename -storetype pkcs12 -storepass password
               -alias label -file certreq_filename
```

Dabei gilt:

-keystore *dateiname*

Gibt den Namen des Keystores an

-storepass *Kennwort*

Gibt das Keystore-Kennwort an

-alias *Bezeichnung*

Gibt die Zertifikatsbezeichnung an.

-file *Dateiname_Zertanforderung*

Gibt den Dateinamen für die Zertifikatsanforderung an.

- c) Senden Sie die in Schritt „1.b“ auf Seite 592 erstellte Zertifikatsanforderungsdatei zur Unterzeichnung an Ihre Zertifizierungsstelle.
- d) Nachdem die Zertifizierungsstelle Ihnen das signierte Zertifikat gesendet hat, empfangen Sie das signierte Zertifikat im Keystore.

```
V 9.4.0 > V 9.4.0
```

Geben Sie den folgenden Befehl ein, um das signierte Zertifikat im Keystore zu empfangen:

```
mqiptykeytool -importcert -keystore cert_filename -storetype pkcs12 -storepass password
               -file cert_filename
```

Dabei steht *Zertifikatsdateiname* für den Namen der Datei, die das Zertifikat enthält, *Dateiname* für den Namen des Keystores und *Kennwort* für das Keystore-Kennwort.

- e) Verschlüsseln Sie das Keystore-Kennwort mit dem Befehl **mqiptyPW**.

Geben Sie den folgenden Befehl ein:

```
mqiptyPW -sf encryption_key_file
```

Dabei steht *Verschlüsselungsschlüsseldatei* für den Namen einer Datei, die den Kennwortverschlüsselungsschlüssel für Ihre MQIPT-Installation enthält. Sie müssen den Parameter **-sf** nicht angeben, wenn Ihre MQIPT-Installation den Standardkennwortverschlüsselungsschlüssel verwendet. Geben Sie das zu verschlüsselnde Keystore-Kennwort ein, wenn Sie dazu aufgefordert werden.

Weitere Informationen zum Befehl **mqiptyPW** finden Sie im Abschnitt [Schlüsselringkennwort verschlüsseln](#).

- f) Bearbeiten Sie die `mqipty.conf`-Konfigurationsdatei und geben Sie die folgenden Eigenschaften an, um den TLS-Befehlsport zu konfigurieren:
 - i) Setzen Sie den Wert der Eigenschaft **SSLCommandPort** auf die Portnummer des TLS-Befehls.
 - ii) Setzen Sie den Wert der Eigenschaft **SSLCommandPortKeyRing** auf den Dateinamen des in Schritt „1.a“ auf Seite 592 erstellten Keystores.
 - iii) Legen Sie als Wert von **SSLCommandPortKeyRingPW** die Zeichenfolgeausgabe des Befehls **mqiptyPW** in Schritt „1.e“ auf Seite 593 fest.
 - iv) Setzen Sie den Wert der Eigenschaft **SSLCommandPortSiteLabel** auf den Kennsatznamen des TLS-Befehlsportzertifikats, das beim Erstellen der Zertifikatsanforderung in Schritt „1.b“ auf Seite 592 angegeben wurde.
 - v) Wenn Sie die eingehenden Verbindungen zum TLS-Befehlsport auf die Verbindungen von einer bestimmten Netzschchnittstelle beschränken wollen, setzen Sie die Eigenschaft **SSLCommandPortListenerAddress** auf eine Netzadresse, die zu einer der Netzschchnittstellen auf dem System gehört, auf dem MQIPT ausgeführt wird. Wenn Sie beispielsweise eingehende Verbindungen zum TLS-Befehlsport auf die Verbindungen der lokalen Maschine beschränken möchten, setzen Sie den Wert der Eigenschaft **SSLCommandPortListenerAddress** auf `localhost`.
- g) Starten oder aktualisieren Sie MQIPT, um den TLS-Befehlsport zu aktivieren.

MQIPT gibt Konsolennachrichten wie die folgende aus, um die Konfiguration des TLS-Befehlsports anzuzeigen, die aktiv ist:

```
MQCPI155 Listening for control commands on port 1882 on local address * using TLS
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites <NULL>
MQCPI032 .....key ring file c:\\iptHome\\ssl\\commandport.p12
MQCPI072 .....and certificate label mqiptadmin
```

2. Führen Sie auf dem System, auf dem der Befehl **mqiptAdmin** zum Verwalten von MQIPT verwendet wird, diese Schritte aus, um **mqiptAdmin** für die Verbindung zum TLS-Befehlsport zu aktivieren.
 - a) Importieren Sie das CA-Zertifikat der Zertifizierungsstelle, die das TLS-Befehlsportzertifikat signiert hat, in einen PKCS #12 -Keystore, der vom Befehl **mqiptAdmin** als Truststore verwendet werden soll.



Geben Sie den folgenden Befehl ein, um das CA-Zertifikat zu importieren:

```
mqiptKeytool -importcert -keystore filename -storetype pkcs12 -storepass password
              -file cert_filename -alias certlabel
```

Dabei gilt:

Dateiname

Gibt den Namen des zu erstellenden Schlüsselspeichers an

Kennwort

Gibt das Keystore-Kennwort an.

Zertifikatsbezeichnung

Gibt die Bezeichnung an, die dem CA-Zertifikat zugeordnet werden soll.

Dateiname_Zertifikat

Gibt den Namen der Datei an, die das CA-Zertifikat enthält.

- b) Verschlüsseln Sie das Keystore-Kennwort mit dem Befehl **mqiptPW**.

Geben Sie den folgenden Befehl ein:

```
mqiptPW -sf encryption_key_file
```

Datei steht *Verschlüsselungsschlüsseldatei* für den Namen der Datei, die den Schlüssel für die Kennwortverschlüsselung enthält. Die Schlüsseldatei für die Kennwortverschlüsselung kann sich von der Schlüsseldatei unterscheiden, mit der Kennwörter in der MQIPT-Konfiguration verschlüsselt werden. Der Standardkennwortverschlüsselungsschlüssel wird verwendet, wenn Sie keine Chiffrierschlüsseldatei mit dem Parameter **-sf** angeben. Geben Sie das zu verschlüsselnde Keystore-Kennwort ein, wenn Sie dazu aufgefordert werden.

Weitere Informationen zum Befehl **mqiptPW** finden Sie im Abschnitt [Schlüsselringkennwort verschlüsseln](#).

- c) Erstellen Sie eine Eigenschaftendatei, die vom Befehl **mqiptAdmin** verwendet werden soll, und geben Sie die folgenden Eigenschaften an:

```
SSLClientCAKeyRing=key_ring_file_name
SSLClientCAKeyRingPW=key_ring_password
PasswordProtectionKeyFile=encryption_key_file
```

Dabei gilt:

Schlüsselringdateiname

ist der Name des in Schritt „2.a“ auf Seite 594 erstellten Keystores.

Schlüsselringkennwort

Die verschlüsselte Kennwortausgabe vom Befehl **mqiptPW** in Schritt „2.b“ auf Seite 594.

Verschlüsselungsschlüsseldatei

Der Name der Datei, die den Kennwortverschlüsselungsschlüssel enthält. Sie müssen die Eigenschaft **PasswordProtectionKeyFile** nur angeben, wenn eine Verschlüsselungsschlüs-

seldatei zum Verschlüsseln des Schlüsselspeicherkennworts in Schritt „2.b“ auf Seite 594 verwendet wurde.

- d) Geben Sie den Befehl **mqiPTAdmin** zur Verwaltung von MQIPT aus. Durch die Angabe des Parameters **-s** wird angezeigt, dass eine TLS-Verbindung erforderlich ist, und der Parameter **-p** gibt den Namen der Eigenschaftendatei an, die in Schritt „2.c“ auf Seite 594 erstellt wurde.

Geben Sie beispielsweise den folgenden Befehl ein, um eine Instanz von MQIPT zu aktualisieren, indem ein Aktualisierungsbefehl an den TLS-Befehlsport gesendet wird:

```
mqiPTAdmin -refresh -r hostname:port -s -p properties_file
```

Der Befehl **mqiPTAdmin** gibt eine Nachricht aus, die in etwa der folgenden Nachricht entspricht, um zu bestätigen, dass die Verbindung zu MQIPT mit TLS geschützt ist:

```
MQCAI109 The connection to MQIPT is secured with TLSv1.2.
```

Nächste Schritte

Um die Authentifizierung für Befehle zu aktivieren, die vom TLS-Befehlsport empfangen werden, folgen Sie den Schritten im Abschnitt „Authentifizierung des Befehlsports“ auf Seite 595.

Authentifizierung des Befehlsports

MQIPT kann so konfiguriert werden, dass die die vom nicht gesicherten Befehlsport und vom TLS-Befehlsport empfangenen Befehle mit einem Kennwort authentifiziert werden. Gehen Sie wie hier beschrieben vor, um die Authentifizierung des Befehlsports zu aktivieren.

Informationen zu diesem Vorgang

Mit dem Befehl **mqiPTAdmin** werden Benutzer zur Eingabe eines Kennworts aufgefordert, wenn der Befehl eine Verbindung zum Befehlsport einer Instanz von MQIPT herstellt, für die die Authentifizierung des Befehlsports aktiviert ist. MQIPT überprüft das im Befehl **mqiPTAdmin** eingegebene Kennwort mit dem Zugriffskennwort, das in der MQIPT-Konfiguration angegeben ist.

Die Eigenschaften, die Sie für die Befehlsportauthentifizierung festlegen, gelten für den TLS-Befehlsport und den nicht gesicherten Befehlsport.

Vorgehensweise

1. Verschlüsseln Sie das Zugriffskennwort für MQIPT mit dem Befehl **mqiPTPW**.

Geben Sie den folgenden Befehl ein:

```
mqiPTPW -sf encryption_key_file
```

Dabei steht *Verschlüsselungsschlüsseldatei* für den Namen der Datei, die den Kennwortverschlüsselungsschlüssel für Ihre MQIPT-Installation enthält. Sie müssen den Parameter **-sf** nicht angeben, wenn Ihre MQIPT-Installation den Standardkennwortverschlüsselungsschlüssel verwendet. Wenn Sie dazu aufgefordert werden, geben Sie das Zugriffskennwort ein, das verschlüsselt werden soll.

Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

2. Bearbeiten Sie die `mqiPT.conf`-Konfigurationsdatei und geben Sie die folgenden Eigenschaften an:

```
AccessPW=encrypted_password  
RemoteCommandAuthentication=auth_setting
```

Dabei gilt:

verschlüsseltes_Kennwort

Die Ausgabe für das verschlüsselte Kennwort vom Befehl **mqiPTPW** in Schritt „1“ auf Seite 595.

Authentifizierungseinstellung

Die Authentifizierungsanforderung. Die Authentifizierung des Befehlsport ist aktiviert, wenn diese Eigenschaft auf einen der folgenden Werte gesetzt ist:

optional

Es ist kein Kennwort erforderlich, aber wenn ein Kennwort bereitgestellt wird, muss es gültig sein. Diese Option kann beispielsweise bei einer Migration hilfreich sein.

erforderlich

Für jeden von einem Befehlsport empfangenen Befehl muss ein gültiges Kennwort bereitgestellt werden.

Weitere Informationen zu diesen Eigenschaften finden Sie im Abschnitt [Globale MQIPT-Eigenschaften](#).

3. Starten oder aktualisieren Sie MQIPT, damit die Änderungen wirksam werden.

MQIPT gibt eine Nachricht aus, in der angezeigt wird, ob die Authentifizierung des Befehlsport aktiviert ist. Wenn MQIPT beispielsweise so konfiguriert ist, dass bei jeder Ausführung des Befehls **mqiptAdmin** ein gültiges Kennwort erforderlich ist, wird die folgende Nachricht ausgegeben:

```
MQCPI021 Password checking has been enabled on the command port
```

Backups erstellen

Es gibt eine Reihe von MQIPT-Dateien, für die Sie als Teil Ihrer regelmäßigen Sicherungsprozeduren ein Backup erstellen sollten.

Erstellen Sie regelmäßig ein Backup der folgenden Dateien:

- Die Konfigurationsdatei `mqipt.conf`
- Die SSL-/TLS-Schlüsselringdateien, die durch die folgenden Eigenschaften in `mqipt.conf` angegeben werden:
 - **SSLClientKeyRing**
 - **SSLClientCAKeyRing**
 - **SSLServerKeyRing**
 - **SSLServerCAKeyRing**
 - **SSLCommandPortKeyRing**
- Die SSL-/TLS-Schlüsselringkennwortdateien, die durch die folgenden Eigenschaften in `mqipt.conf` angegeben werden:
 - **SSLClientKeyRingPW**
 - **SSLClientCAKeyRingPW**
 - **SSLServerKeyRingPW**
 - **SSLServerCAKeyRingPW**
- Die Schlüsseldatei für die Kennwortverschlüsselung, wenn die MQIPT-Konfiguration Kennwörter enthält, die mit einem anderen Verschlüsselungsschlüssel als dem Standardschlüssel verschlüsselt sind.
- Die durch **SecurityManagerPolicy** angegebene Richtliniendatei, falls diese Eigenschaft festgelegt wurde.
- Die Sicherheitsexitdateien und die Zertifikatsexitdateien, die durch die folgenden Eigenschaften in `mqipt.conf` angegeben werden:
 - **SecurityExitName**
 - **SSLExitName**
- Verbindungsprotokolldateien im Unterverzeichnis `log` des Ausgangsverzeichnisses von MQIPT, wenn diese für Prüfzwecke benötigt werden.

Leistungsoptimierung

Sie können die relative Leistung jeder MQIPT-Route mit einer Kombination aus einem Thread-Pool und einer Spezifikation für das Inaktivitätszeitlimit optimieren.

Verbindungsthreads

Jeder MQIPT-Route wird ein Arbeitspool mit gleichzeitig aktiven Threads zugeordnet, die eingehende Kommunikationsanforderungen verarbeiten. Bei der Initialisierung wird ein Thread-Pool erstellt (dessen Größe im Attribut `MinConnectionThreads` der Route angegeben ist) und es wird ein Thread zugeordnet, der die erste eingehende Anforderung bearbeiten soll. Wenn diese Anforderung ankommt, wird ein anderer Thread zugeordnet, der für die nächste eingehende Anforderung bereit ist. Wenn alle Threads für die Arbeit zugeordnet sind, wird ein neuer Thread erstellt, dem Arbeitspool hinzugefügt und einer Arbeit zugeordnet.

Auf diese Weise wächst der Pool, bis die maximale Anzahl an Threads (in `MaxConnectionThreads` angegeben) erreicht ist. Threads werden bei Beendigung eines Dialogs oder bei Ablauf des angegebenen Inaktivitätszeitlimits wieder in den Pool freigegeben. Wenn die maximale Anzahl an aktiven Threads erreicht ist, wartet die nächste eingehende Anforderung, bis wieder ein Thread in den Arbeitspool freigegeben wird.

Durch Erhöhen der Anzahl verfügbarer Threads können Sie die mögliche Wartezeit für Anforderungen reduzieren. Diese Erhöhung muss jedoch mit den verfügbaren Systemressourcen abgewogen werden.

Inaktivitätszeitlimit

Standardmäßig werden aktive Threads nicht aufgrund von Inaktivität beendet. Wenn ein Thread einem Dialog zugeordnet wurde, bleibt die Zuordnung zu dem Dialog bestehen, bis dieser normal geschlossen wird, die Route inaktiviert wird oder MQIPT beendet wird. Optional können Sie ein Inaktivitätszeitlimit (in Minuten) in der Eigenschaft `IdleTimeout` angeben, sodass Threads, die für den angegebenen Zeitraum inaktiv waren, gestoppt und erneut gestartet werden. Threads werden erneut zur Verwendung gestartet, indem sie wieder in den Arbeitspool zurückgestellt werden.

Setzen Sie bei nicht unterbrechungsfreier IBM MQ-Aktivität das Intervall der Überwachungssignale auf einen kleineren Wert als für das MQIPT-Zeitlimit festgelegt, damit die Threads nicht ständig gestoppt und erneut gestartet werden.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in diesem Dokument beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf Produkte, Programme oder Services von IBM bedeuten nicht, dass nur Produkte, Programme oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Défense
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East & Africa
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmieretechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos ohne Zahlung an IBM in jeder Form kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben sind. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

Informationen zu Programmierschnittstellen

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

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

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

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

Marken

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

Microsoft und Windows sind eingetragene Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Dieses Produkt enthält Software, die von Eclipse Project (<https://www.eclipse.org/>) entwickelt wurde.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.



Teilenummer:

(1P) P/N: