

7.5

*IBM WebSphere MQ -Übersicht*

**IBM**

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 239 gelesen werden.

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

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

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

---

# Inhaltsverzeichnis

<b>Informationen zu IBM WebSphere MQ.....</b>	<b>5</b>
Einführung in IBM WebSphere MQ.....	6
IBM WebSphere MQ - Lizenzinformationen.....	9
Einführung in IBM WebSphere MQ Telemetry.....	11
IBM WebSphere MQ Version 7.5 - Literaturübersicht.....	15
IBM WebSphere MQ Version 7.5 in der App IBM Documentation Offline.....	19
PDF-Dokumentation zu IBM WebSphere MQ 7.5.....	19
Neuerungen in IBM WebSphere MQ Version 7.5.....	20
IBM WebSphere MQ-Client für HP Integrity NonStop Server.....	22
Änderungen in IBM WebSphere MQ Version 7.5.....	25
Änderungen in IBM WebSphere MQ Version 7.5 -Fixpacks.....	26
Neuerungen und Änderungen in älteren Versionen.....	35
Zuordnung älterer IBM WebSphere MQ-Veröffentlichungen zur aktuellen Informationsstruktur.....	35
Einstieg für AIX.....	35
Einstieg für HP-UX.....	36
Einstieg für Linux.....	36
Einstieg für Solaris.....	36
Einstieg für Windows.....	37
Application Programming Guide.....	37
Application Programming Reference.....	37
Kunden.....	37
Konstanten.....	38
Intercommunication.....	38
Nachrichten und Codes.....	39
Migrationsinformationen.....	39
Überwachung.....	39
Programmable Command Formats and Administration Interface.....	40
Publish/Subscribe-Benutzerhandbuch.....	40
Cluster aus Warteschlangenmanagern.....	40
Referenz zu Scriptbefehlen (MQSC).....	40
Sicherheit.....	40
System Administration Guide.....	41
.NET verwenden.....	41
C++ verwenden.....	41
Using Java.....	41
Web-Services.....	42
Component Object Model-Schnittstelle verwenden.....	42
IBM WebSphere MQ Version 7.5, IBM i und z/OS.....	42
Bemerkungen.....	42
Technische Übersicht.....	44
Einführung in die Nachrichtenwarteschlangensteuerung.....	45
Konzepte der übergreifenden Kommunikation.....	53
IBM WebSphere MQ Telemetry.....	72
Objekte verwalten.....	111
IBM WebSphere MQ Multicast.....	137
Sicherheit.....	139
Clients und Server.....	139
Transaktionsmanagement und -unterstützung.....	147
Funktionen des Warteschlangenmanagers erweitern.....	149
IBM WebSphere MQ-Client für HP Integrity NonStop Server - Technische Übersicht.....	150
Szenarien.....	151
Erste Schritte mit IBM WebSphere MQ 7.5.....	151

Basisdateiübertragung mithilfe der Scripts.....	159
Basisdateiübertragung im Detail.....	167
Dateiübertragung zwischen zwei Computern mithilfe der Scripts.....	173
Dateiübertragung zwischen zwei Computern im Detail.....	180
Prüffunktion zur verwalteten Dateiübertragung hinzufügen.....	188
Glossar.....	191
A.....	192
B.....	195
C.....	196
D.....	201
E.....	203
F.....	204
G.....	206
H.....	206
I.....	207
J.....	209
K.....	210
L.....	210
M.....	212
N.....	216
O.....	217
P.....	219
Q.....	222
R.....	223
S.....	226
T.....	231
U.....	234
V.....	235
W.....	235
X.....	236
Behindertengerechte Bedienung.....	236
Eingabehilfen unter Windows.....	237
<b>Bemerkungen.....</b>	<b>239</b>
Informationen zu Programmierschnittstellen.....	240
Marken.....	241

# Informationen zu IBM WebSphere MQ

---

Dieser Abschnitt enthält einführende Informationen zur ersten Verwendung von IBM WebSphere MQ:

## **Zugehörige Tasks**

[WebSphere MQ-Architektur entwerfen](#)

## IBM WebSphere MQ

---

Willkommen bei der Produktdokumentation zu IBM WebSphere MQ Version 7.5 . Hier finden Sie ausführliche Anweisungen zur Ausführung der Tasks, die Sie zum Erstellen und Verwalten Ihrer MQ -Umgebung ausführen müssen. Darüber hinaus werden in der Dokumentation die zentralen Konzepte des Produkts erläutert und es wird beschrieben, wie das Produkt Sie beim Lösen bestimmter Geschäftsprobleme unterstützen kann.



IBM WebSphere MQ ist eine leistungsfähige, sichere und zuverlässige Messaging-Middleware. Mithilfe von Nachrichten und Warteschlangen ermöglicht sie den Austausch von Informationen zwischen Anwendungen, Systemen, Services und Dateien. Sie erleichtert und beschleunigt die Integration unterschiedlichster Anwendungen und Daten auf einer Vielzahl von Plattformen.

## **Zugehörige Konzepte**

[Informationen zu IBM WebSphere MQ](#)

## **Zugehörige Tasks**

[Planung](#)

[Migration und Upgrades](#)

[Installieren](#)

[Sicherheit](#)

[Konfiguration](#)

[Verwalten](#)

[Anwendungen entwickeln](#)

[Überwachung und Leistung](#)

[Fehlerbehebung und Unterstützung](#)

[WebSphere MQ Managed File Transfer](#)

[WebSphere MQ Explorer](#)

[Hilfe zum Assistenten für die grafische Benutzerschnittstelle von MQ](#)

[WebSphere MQ Internet Pass-Thru](#)

[Message Service Client for .NET](#)

[WebSphere MQ Hypervisor-Editionen](#)

[Mobile Messaging and M2M](#)

## **Zugehörige Verweise**

[Referenz](#)

## **Zugehörige Informationen**

[IBM MQ on Cloud](#)

# Einführung in IBM WebSphere MQ

---

Mithilfe von IBM WebSphere MQ können Sie es Anwendungen ermöglichen, zu verschiedenen Zeiten und in vielen verschiedenen IT-Umgebungen zu kommunizieren.

## Was ist IBM WebSphere MQ?

- IBM WebSphere MQ ist Messaging für Anwendungen. Es sendet Nachrichten über Netze aus unterschiedlichen Komponenten. Eine Anwendung stellt eine Verbindung zu IBM WebSphere MQ her, um eine Nachricht zu senden oder zu empfangen. IBM WebSphere MQ handhabt die unterschiedlichen Prozessoren, Betriebssysteme, Subsysteme und Kommunikationsprotokolle, die es bei der Übertragung der Nachricht vorfindet. Wenn eine Verbindung oder ein Prozessor vorübergehend nicht verfügbar ist, stellt IBM WebSphere MQ die Nachricht in eine Warteschlange und leitet sie weiter, sobald die Verbindung wieder online ist.
- Eine Anwendung kann aus verschiedenen Programmierschnittstellen und Programmiersprachen auswählen, um eine Verbindung zu IBM WebSphere MQ herzustellen.
- IBM WebSphere MQ ist eine Middleware für *Messaging* (Nachrichtenübertragung) und *Queuing* (Warteschlangensteuerung) und bietet die Betriebsarten *Punkt-zu-Punkt*, *Publish/Subscribe* und *Dateiübertragung*. Anwendungen können über *Multicasting* Nachrichten an viele Subskribenten veröffentlichen.

### **Messaging**

Programme kommunizieren miteinander, indem sie sich gegenseitig Daten in Nachrichten zusenden, statt sich direkt anzurufen.

### **Queuing**

Nachrichten werden in Warteschlangen gestellt, sodass Programme unabhängig voneinander mit unterschiedlichen Übertragungsgeschwindigkeiten und zu unterschiedlichen Zeiten an verschiedenen Standorten ausgeführt werden können, ohne dass eine direkte Verbindung zwischen ihnen besteht.

### **Punkt-zu-Punkt**

Anwendungen senden Nachrichten an eine Warteschlange oder an eine Liste mit Warteschlangen. Der Sender muss den Namen des Ziels kennen, aber nicht wissen, wo es sich befindet.

### **Publish/Subscribe**

Anwendungen veröffentlichen eine Nachricht zu einem Thema, z. B. das Ergebnis eines Spiels, das von einem Team gespielt wird. IBM WebSphere MQ sendet Kopien der Nachricht an Anwendungen, die das entsprechende Thema subskribiert haben. Diese empfangen die Nachricht mit den Ergebnissen von Spielen, die vom Team gespielt werden. Die Veröffentlichungskomponente kennt nicht die Namen der Subskribenten und weiß auch nicht, wo sie sich befinden.

### **Multicast**

Multicasting ist eine effiziente Form von Publish/Subscribe-Messaging, das für viele Subskribenten skaliert werden kann. Dabei wird der Aufwand für das Senden der Kopie einer Veröffentlichung an die einzelnen Subskribenten aus IBM WebSphere MQ in das Netz verlagert. Nachdem zwischen der Veröffentlichungskomponente und dem Subskribenten ein Pfad für die Veröffentlichung eingerichtet wurde, ist IBM WebSphere MQ nicht mehr an der Weiterleitung der Veröffentlichung beteiligt.

### **Dateiübertragung**

Dateien werden in Nachrichten übertragen. IBM WebSphere MQ File Transfer Edition steuert die Übertragung von Dateien und die Verwaltung zur Einrichtung automatisierter Übertragungen und zur Protokollierung der Ergebnisse. Die Dateiübertragung kann mit anderen Dateiübertragungssystemen, mit IBM WebSphere MQ-Messaging und dem World Wide Web integriert werden.

### **Telemetry**

IBM WebSphere MQ Telemetry ist Messaging für Geräte. IBM WebSphere MQ verbindet Geräte- und Anwendungs-Messaging miteinander. Es verbindet das Internet, Anwendungen, Services und Entscheidungsträger mit Netzen aus instrumentierten Geräten. IBM WebSphere MQ Telemetry verfügt über ein effizientes Messaging-Protokoll, das eine große Anzahl von Geräten über ein Netz verbindet. Das Messaging-Protokoll wird veröffentlicht, sodass es in Geräte integriert wer-

den kann. Sie haben auch die Möglichkeit, mithilfe einer der für das Protokoll veröffentlichten Programmierschnittstellen Geräteprogramme zu entwickeln.

### **Was kann die Software für mich tun?**

- IBM WebSphere MQ sendet und empfängt Daten zwischen Ihren Anwendungen und über Netze hinweg.
- Die Nachrichtenübermittlung ist *zuverlässig* und von der Anwendung *entkoppelt*. Zuverlässig, weil IBM WebSphere MQ Nachrichten transaktionsgesteuert austauscht, und entkoppelt, weil Anwendungen nicht überprüfen müssen, ob Nachrichten, die sie gesendet haben, sicher zugestellt werden.
- Sie können die Nachrichtenübermittlung zwischen Warteschlangenmanagern mit SSL/TLS schützen.
- Mithilfe von Advanced Message Security (AMS) können Sie Nachrichten zwischen dem Zeitpunkt des Einreichens durch eine Anwendung und dem Zeitpunkt des Abrufens durch eine andere Anwendung verschlüsseln und signieren.
- Anwendungsprogrammierer müssen dazu keine Kenntnisse über das Programmieren von Kommunikationsroutinen besitzen.

### **Wozu wende ich die Software an?**

- Erstellen und verwalten Sie IBM WebSphere MQ über die grafische Benutzerschnittstelle IBM WebSphere MQ Explorer oder indem Sie Befehle aus einem Befehlsfenster oder einer Anwendung ausführen.
- Programmieren Sie Anwendungen zum Senden und Empfangen von Nachrichten, indem Sie eine der Programmierschnittstellen aufrufen. Programmierschnittstellen werden für verschiedene Sprachen bereitgestellt, darunter die Standardprogrammierschnittstelle JMS JMS und Klassen für die Windows Communication Foundation.
- Senden und empfangen Sie IBM WebSphere MQ-Nachrichten in Browsern mit dem HTTP-Protokoll.

### **Wie funktioniert die Software?**

- Ein Administrator erstellt und startet einen Warteschlangenmanager mit Befehlen. Danach wird der Warteschlangenmanager üblicherweise automatisch gestartet, sobald das Betriebssystem gebootet wird. Anwendungen und andere Warteschlangenmanager können dann eine Verbindung zu ihm herstellen und Nachrichten senden und empfangen.
- Eine Anwendung oder ein Administrator erstellt eine Warteschlange oder ein Thema. Warteschlangen und Themen sind Objekte, die von einem *Warteschlangenmanager* als ihrem Eigner gespeichert werden.
- Wenn eine Anwendung Daten an eine andere Anwendung übertragen möchte, stellt sie die Daten in eine Nachricht. Sie reiht die Nachricht in eine Warteschlange ein oder veröffentlicht sie für ein Thema. Es gibt drei Möglichkeiten zum Abrufen der Nachricht:
  - Eine Punkt-zu-Punkt-Anwendung, die mit demselben Warteschlangenmanager verbunden ist, ruft die Nachricht aus der Warteschlange ab.

Beispiel 1: Eine Anwendung reiht Nachrichten in eine Warteschlange ein, um auf diese Weise temporäre oder persistente Daten zu speichern. Beispiel 2: Eine Anwendung nutzt Daten gemeinsam mit einer anderen Anwendung, die in einem anderen Prozess ausgeführt wird.

- Eine Punkt-zu-Punkt-Anwendung, die mit einem anderen Warteschlangenmanager verbunden ist, ruft die Nachricht aus einer anderen Warteschlange ab.

Anwendungen kommunizieren miteinander, indem sie Nachrichten über Warteschlangen austauschen. Der Hauptnutzen von IBM WebSphere MQ besteht darin, dass es Nachrichten sendet oder den Austausch von Nachrichten ermöglicht. Eine Anwendung stellt eine Nachricht in eine Warteschlange auf einem Computer und eine andere Anwendung erhält die Nachricht aus einer anderen Warteschlange auf einem anderen Computer. Die Warteschlangenmanager auf den beiden Computern arbeiten zusammen, um die Nachricht aus der ersten Warteschlange in die zweite Warteschlange zu übertragen. Nicht die Anwendungen kommunizieren miteinander, sondern die Warteschlangenmanager.

- Eine Subskribentenanwendung, die mit einem Warteschlangenmanager verbunden ist, ruft Nachrichten zu allgemeinen Themen ab.

Eine Veröffentlichungsanwendung erstellt eine Nachricht und veröffentlicht sie für ein Thema auf einem Computer. Beliebige viele Subskribentenanwendungen abonnieren dasselbe Thema auf anderen Computern. IBM WebSphere MQ übermittelt die Veröffentlichung an Warteschlangen, die zu den Warteschlangenmanagern gehören, mit denen die Subskribenten verbunden sind. Die Subskribenten rufen die Nachricht aus den Warteschlangen ab.

- *MQ-Kanäle* verbinden einen Warteschlangenmanager über ein Netz mit einem anderen Warteschlangenmanager. Sie können selbst MQ-Kanäle erstellen oder ein Warteschlangenmanager in einem Cluster aus Warteschlangenmanagern erstellt MQ-Kanäle, wenn sie benötigt werden.
- Einem Warteschlangenmanager können viele Warteschlangen und Themen zugeordnet sein.
- Auf einem Computer können sich mehrere Warteschlangenmanager befinden.
- Eine Anwendung kann auf demselben Computer wie der Warteschlangenmanager oder auf einem anderen Computer ausgeführt werden. Wird sie auf demselben Computer ausgeführt, handelt es sich um eine IBM WebSphere MQ-Serveranwendung. Wird sie auf einem anderen Computer ausgeführt, handelt es sich um eine IBM WebSphere MQ-Clientanwendung. Ob es ein IBM WebSphere MQ-Client oder -Server ist, macht für die Anwendung fast keinen Unterschied. Sie können eine Client/Server-Anwendung mit IBM WebSphere MQ-Clients oder -Servern erstellen.

### **Welche Tools und Ressourcen sind im Lieferumfang von IBM WebSphere MQ enthalten?**

- Steuerbefehle, die aus der Befehlszeile ausgeführt werden. Mit den Steuerbefehlen erstellen, starten und stoppen Sie Warteschlangenmanager. Auch Verwaltungs- und Problembestimmungsprogramme von IBM WebSphere MQ werden mit den Steuerbefehlen ausgeführt.
- IBM WebSphere MQ-Scriptbefehle (MQSC), die von einem Interpreter ausgeführt werden. Mit den Befehlen können Sie Warteschlangen und Themen erstellen und IBM WebSphere MQ konfigurieren und verwalten. Bearbeiten Sie die Befehle in einer Datei und übergeben Sie die Datei an das Programm **runmqsc**, das die Befehle interpretiert. Sie können den Interpreter auch auf einem Warteschlangenmanager ausführen, der die Befehle an einen anderen Computer sendet, um einen anderen Warteschlangenmanager zu verwalten.
- PCF-Befehle (Programmable Command Format), die Sie in eigenen Anwendungen zur Verwaltung von IBM WebSphere MQ aufrufen. Die PCF-Befehle bieten dieselbe Funktionalität wie die Scriptbefehle, sind jedoch einfacher zu programmieren.
- Beispielprogramme
- Auf Windows- und Linux® x86- und x86-64-Plattformen können folgende Dienstprogramme ausgeführt werden:
  - IBM WebSphere MQ Explorer. Mit dem Explorer können dieselben Verwaltungsaufgaben wie mit den Scriptbefehlen ausgeführt werden, er ist aber im Dialogbetrieb einfacher zu bedienen.
  - Anwendung *Postcard* zur Veranschaulichung der Messaging-Funktionalität und zur Überprüfung der Installation
  - Lernprogramme

### **Zugehörige Konzepte**

„Neuerungen in IBM WebSphere MQ Version 7.5“ auf Seite 20

In diesem Abschnitt werden die wichtigsten neuen Funktionen in IBM WebSphere MQ Version 7.5 beschrieben.

#### WebSphere MQ Multicast

IBM WebSphere MQ Multicast bietet eine geringe Latenzzeit, eine hohe Ausgabefächerung und eine zuverlässige Multicasting-Nachrichtenübertragung.

#### WebSphere MQ Telemetry

Privatpersonen, Unternehmen und Regierungen interessieren sich mehr und mehr für die Verwendung von IBM WebSphere MQ Telemetry, um schnell mit dem Lebens- und Arbeitsumfeld interagieren zu können. IBM WebSphere MQ Telemetry verbindet alle Arten von Geräten mit dem Internet und Unternehmen, wodurch sich die Kosten für die Erstellung von Anwendungen für Smart Devices verringern.



### Technische Einführung in Messaging und Queuing

Mit den WebSphere MQ-Produkten können Programme über ein Netz ungleicher Komponenten (Prozessoren, Betriebssysteme, Subsysteme und Kommunikationsprotokolle) unter Verwendung einer konsistenten Anwendungsprogrammierschnittstelle miteinander kommunizieren.

### Technische Einführung in WebSphere MQ-Clients und -Server

Dieser Abschnitt enthält allgemeine Hinweise zur IBM WebSphere MQ-Unterstützung von Client/Server-Konfigurationen für Anwendungen.

### Technische Einführung in die Warteschlangenmanager-Kommunikation

In WebSphere MQ bezeichnet übergreifende Kommunikation das Senden von Nachrichten von einem Warteschlangenmanager an einen anderen. Der empfangende Warteschlangenmanager kann sich auf demselben System oder auf einem anderen System ganz in der Nähe oder am anderen Ende der Welt befinden. Er kann auf derselben Plattform wie der lokale Warteschlangenmanager oder jeder anderen von WebSphere MQ unterstützten Plattform aktiv sein. Dies wird als *verteilte* Umgebung bezeichnet. WebSphere MQ wickelt die Kommunikation in einer verteilten Umgebung wie dieser mithilfe des verteilten Warteschlangenmanagements ab.

### **Zugehörige Tasks**

WebSphere MQ Advanced Messages Security (AMS)

WebSphere MQ Managed File Transfer

## **IBM WebSphere MQ - Lizenzinformationen**

---

Was Sie mit IBM WebSphere MQ kaufen können und zu welchen Installationen Sie bei jedem einzelnen Kauf berechtigt sind.

**Hinweis:** Dieses Lizenzhandbuch enthält ergänzende Informationen, die Sie bei der Implementierung der Programme unterstützen, die Sie über IBM innerhalb Ihrer erworbenen Berechtigung lizenziert haben. Ihre Lizenzvereinbarung (z. B. die Internationalen Nutzungsbedingungen für Programmpakete der IBM (IPLA) oder eine gleichwertige Angabe) und die zugehörigen Transaktionsdokumente, einschließlich der Lizenzinformationen für das IBM WebSphere MQ Produktangebot, stellen die einzige und vollständige Vereinbarung zwischen Ihnen und IBM in Bezug auf die Nutzung des Programms dar.

### **Was Sie mit IBM WebSphere MQ kaufen können**

#### **Verteilte Plattformen**

Für IBM WebSphere MQ auf verteilten Plattformen enthält das Produktangebot 11 gebührenpflichtige Komponenten, die unabhängig voneinander erworben werden können:

#### **5724-H72 IBM IBM WebSphere MQ**

IBM IBM WebSphere MQ (Server)

IBM IBM WebSphere MQ Telemetrie

IBM IBM WebSphere MQ Erweiterte Nachrichtensicherheit

IBM IBM WebSphere MQ Idle Standby

IBM IBM WebSphere MQ Advanced Message Security-Idle Standby

IBM IBM WebSphere MQ Erweitert

IBM IBM WebSphere MQ Advanced Idle Standby

IBM IBM WebSphere MQ Advanced für Entwickler

IBM IBM WebSphere MQ Managed File Transfer -Service

IBM IBM WebSphere MQ Managed File Transfer -service Idle Standby

Von IBM IBM WebSphere MQ Managed File Transfer verwalteter Endpunkt

### **Zu welchen Installationen ist mein Unternehmen berechtigt?**

Für IBM WebSphere MQ auf verteilten Plattformen sind die unten genannten Komponenten gleichbedeutend mit den Komponenten, die das IBM WebSphere MQ-Installationsprogramm installieren kann, d. h., es gibt eine einfache Zuordnung zwischen dem, was Sie gekauft haben, und dem, was Sie installieren können.

**Wichtig:** Die IBM WebSphere MQ-Installationsmedien enthalten alle Komponenten, aber Sie sollten nur die Untermenge installieren, für die Sie eine Berechtigung erworben haben.

#### **5724-H72 IBM IBM WebSphere MQ**

IBM WebSphere (Server)

Inhalt:

- ClientDevelopment Kit (SDK)
- IBM Global Security Kit (UNIX)
- IBM WebSphere MQExplorer
- Java .NET Messaging und Web Services
- Beispielprogramme
- Server / Runtime
- UNIX-Man-Pages

IBM IBM WebSphere MQ Telemetrie

Inhalt:

- Telemetrieservice

IBM IBM WebSphere MQ Erweiterte Nachrichtensicherheit

Inhalt:

- Erweiterte Nachrichtensicherheit

IBM IBM WebSphere MQ Managed File Transfer -Service

Inhalt:

- IBM WebSphere MQ Managed File Transfer Logger
- IBM WebSphere MQ Managed File Transfer Service
- IBM WebSphere MQ Managed File Transfer-Tools

IBM WebSphere IBM WebSphere MQ Managed File Transfer Verwalteter Endpunkt

Inhalt:

- IBM WebSphere MQ Managed File Transfer-Agent
- IBM WebSphere MQ Managed File Transfer Logger
- IBM WebSphere MQ Managed File Transfer-Tools

### **Was ist IBM IBM WebSphere MQ Advanced?**

IBM WebSphere MQ Advanced wurde eingeführt, um den Erwerb von Berechtigungen zu vereinfachen. Ihr Unternehmen bezahlt nur einen Preis und erhält die Berechtigung für mehrere IBM WebSphere MQ-Komponenten.

IBM WebSphere MQ Advanced beinhaltet folgende Komponenten:

#### **5724-H72 IBM IBM WebSphere MQ**

IBM IBM WebSphere MQ Erweitert

IBM IBM WebSphere MQ Advanced für Entwickler

Für IBM IBM WebSphere MQ auf verteilten Plattformen ist Ihr Unternehmen beim Kauf von 100 Prozessor-Value-Units (PVUs) von IBM WebSphere MQ Advanced zu folgenden Installationen berechtigt:

- 100 PVUs von IBM IBM WebSphere MQ (Server) **und**

- 100 PVUs von IBM IBM WebSphere MQ Advanced Message Security **und**
- 100 PVUs von IBM IBM WebSphere MQ Managed File Transfer Service **und**
- Unbegrenzte Installationen von IBM IBM WebSphere MQ Telemetry

Darüber hinaus kann Ihr Unternehmen IBM-Versionen beliebig kombinieren. Sie könnten also die durch den Kauf von 100 PVUs von IBM IBM WebSphere MQ (Server) erworbene Berechtigung in 50 PVUs für IBM WebSphere MQ Version 7.1 und 50 PVUs für IBM WebSphere MQ Version 7.5 dieser Komponente aufteilen.

IBM IBM WebSphere MQ Advanced for Developers umfasst die Berechtigung für alles, was in IBM IBM WebSphere MQ Advanced eingeschlossen ist, plus IBM IBM WebSphere MQ Managed File Transfer Managed Endpoint nur für Entwicklungszwecke.



**Achtung:** Die IBM-Lizenz definiert, was Entwicklungszwecke sind.

## Was sind Idle Standby-Komponenten?

Idle Standby-Komponenten wurden speziell für Hochverfügbarkeitsumgebungen eingeführt, in denen IBM WebSphere MQ auf dem passiven System installiert und verfügbar ist, dieses System aber keine IBM WebSphere MQ-Verarbeitung oder sonstige Aktivität ausführt, sondern nur auf dem aktuellen Stand der Konfiguration und Aktivität des aktiven Warteschlangenmanagers gehalten wird. In diesem Fall kann eine geringere Gebühr angemessen sein.

### Anmerkungen:

1. Für die Funktion des IBM WebSphere MQ-Multi-Instanz-Warteschlangenmanagers ist auch eine Idle Standby-Berechtigung erforderlich.
2. Es gibt keine Idle Standby-Komponente für IBM WebSphere MQ Telemetry. Für das aktive und das passive System muss die gleiche IBM IBM WebSphere MQ Telemetry-Komponente erworben werden, außer wenn Sie IBM IBM WebSphere MQ Advanced Idle Standby besitzen, denn dort ist diese Komponente eingeschlossen.
3. Auch für IBM WebSphere Managed File Transfer Managed Endpoint gibt es keine Standby-Komponente, weil der Endpunkt nicht Teil der Serverumgebung ist.

### Zugehörige Konzepte

„Neuerungen in IBM WebSphere MQ Version 7.5“ auf Seite 25

„IBM WebSphere MQ - Technische Übersicht“ auf Seite 44

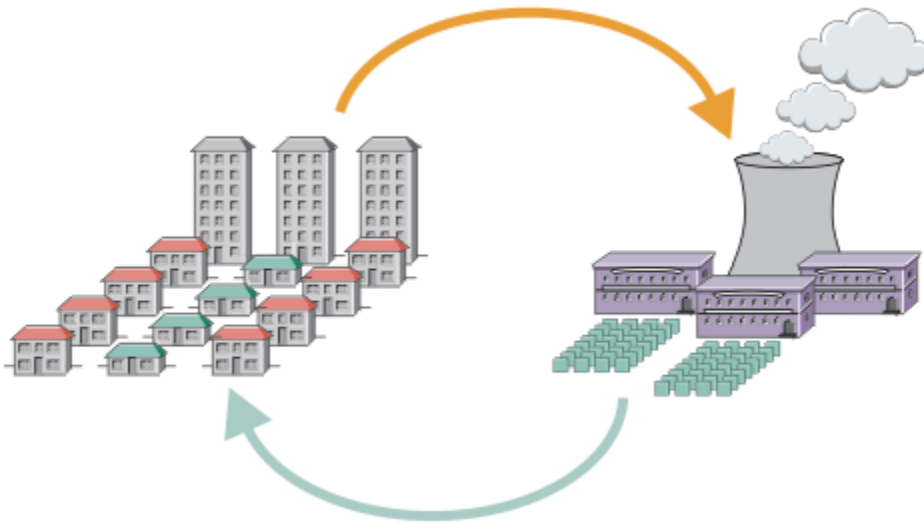
Mit IBM WebSphere MQ können Sie Verbindungen für Ihre Anwendungen herstellen und die Verteilung von Informationen in Ihrem Unternehmen verwalten.

## Einführung in IBM WebSphere MQ Telemetry

---

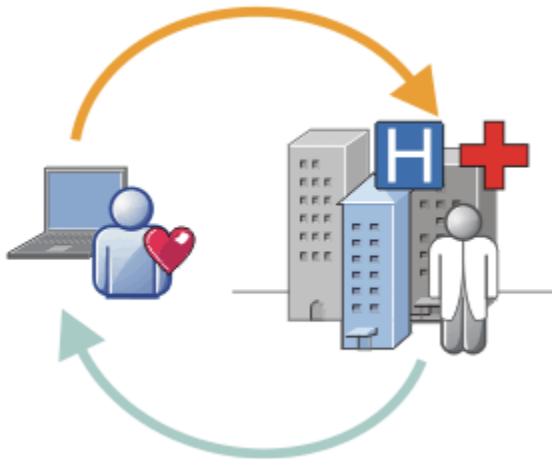
Privatpersonen, Unternehmen und Regierungen interessieren sich mehr und mehr für die Verwendung von IBM WebSphere MQ Telemetry, um schnell mit dem Lebens- und Arbeitsumfeld interagieren zu können. IBM WebSphere MQ Telemetry verbindet alle Arten von Geräten mit dem Internet und Unternehmen, wodurch sich die Kosten für die Erstellung von Anwendungen für Smart Devices verringern.

Die folgenden Diagramme zeigen Beispiele zur typischen Verwendung von IBM WebSphere MQ Telemetry:



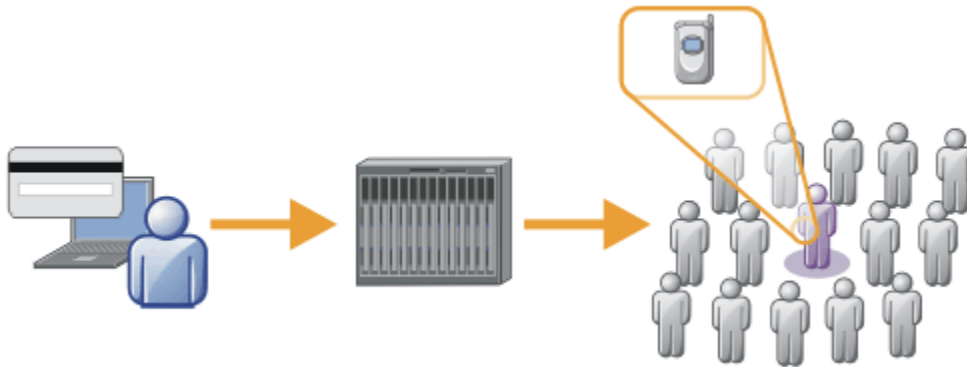
- Eine MQTT-Nachricht mit Energieverbrauchsdaten, die an den Serviceanbieter gesendet wird.
- Eine Telemetrieapplikation sendet Steuerbefehle, die auf einer Analyse von Energieverbrauchsdaten basieren.
- Weitere Informationen finden Sie unter „Telemetrieszenario: Überwachung und Steuerung des Energieverbrauchs in Privathaushalten“ auf Seite 78.

*Abbildung 1. Intelligente Elektrizitätsmessung*



- Eine Telemetrieapplikation sendet Gesundheitsdaten an ein Krankenhaus oder einen Arzt.
- Auf der Grundlage der Analyse Ihrer Gesundheitsdaten werden MQTT-Rückmeldungen mit Nachrichtenalerts oder Feedback gesendet.
- Weitere Informationen finden Sie unter „Telemetrieszenario: Überwachung nicht-stationärer Patienten“ auf Seite 76.

*Abbildung 2. Intelligente Gesundheitsüberwachung*



- Eine einfache Kartentransaktion wird an den Server einer Bank gesendet.
- IBM WebSphere MQ Telemetry ermittelt die betreffende Kontaktperson und warnt den Kunden, dass seine Karte verwendet wurde.
- IBM WebSphere MQ Telemetry kann mithilfe einfachster Eingabeinformationen die betreffende Kontaktperson ausfindig machen.

Abbildung 3. Telemetry: One in a Crowd

## Was ist WebSphere MQ Telemetry?

- Es ist eine Funktion von IBM WebSphere MQ , die den von IBM WebSphere MQ bereitgestellten Universal-Messaging-Backbone auf eine Vielzahl ferner Sensoren, Aktuatoren und Telemetriegeräte erweitert. IBM WebSphere MQ Telemetry erweitert IBM WebSphere MQ , sodass intelligente Unternehmensanwendungen, Services und Entscheidungsträger mit Netzen instrumentierter Geräte verbunden werden können.
- WebSphere MQ Telemetry setzt sich aus zwei zentralen Bestandteilen zusammen:
  1. Der IBM WebSphere MQ Telemetry -Service, der innerhalb des IBM WebSphere MQ -Servers ausgeführt wird.
  2. Den IBM WebSphere MQ Telemetry-Clients, die zusammen mit den Anwendungen an Geräte verteilt werden.

## Was kann die Software für mich tun?

- MQ Telemetry verwendet MQ Telemetry Transport (MQTT) zum Senden und Empfangen von Daten zwischen Ihren Anwendungen und dem IBM WebSphere MQ -Warteschlangenmanager.
- MQTT ist ein offenes Messaging-Protokoll, mit dem MQTT-Implementierungen für eine Vielzahl von Geräten erstellt werden können.
- MQTT-Clients können auf Geräten mit einem geringen Speicherbedarf ausgeführt werden, deren Ressourcen unter Umständen begrenzt sind.
- MQTT kann auf effiziente Weise in Netzen mit niedriger Bandbreite eingesetzt werden, bei denen das Versenden von Daten kostenintensiv ist, oder die möglicherweise störanfällig sind.
- Die Nachrichtenübermittlung wird von der Anwendung sichergestellt und voneinander entkoppelt.
- Anwendungsprogrammierer müssen dazu keine Kenntnisse über das Programmieren von Kommunikationsroutinen besitzen.
- Nachrichten können mit anderen Messaging-Anwendungen ausgetauscht werden, Dabei kann es sich um andere Telemetrieanwendungen, MQI-, JMS- oder Messaging-Anwendungen der Enterprise-Klasse handeln.

## Wozu wende ich die Software an?

- Mit IBM WebSphere MQ Explorer und den zugehörigen Tools können Sie die Funktion WebSphere MQ Telemetry von MQverwalten.
- Stellen Sie mithilfe der MQTT-Clients in Ihren Anwendungen eine Verbindung zu einem Warteschlangenmanager her und veröffentlichen und abonnieren Sie Nachrichten.
- Stellen Sie Ihre Anwendung mit dem MQTT-Client auf dem Gerät zur Verfügung, auf dem die Anwendung ausgeführt werden soll.

## Wie funktioniert die Software?

- Der MQ Telemetry-Service (MQXR) wandelt einen IBM WebSphere MQ -Warteschlangenmanager in einen MQTT-Server um.
- Der MQTT-Server erkennt das MQTT-Messaging-Protokoll und kann Nachrichten von MQTT-Clients empfangen sowie Nachrichten an diese senden.
- MQ Telemetry wird mit mehreren Telemetrieclients ausgeliefert, die das MQTT-Messaging-Protokoll implementieren. Diese werden häufig als MQTT-Clients bezeichnet.
- Ein einfacher Telemetrieclient funktioniert wie ein MQ-Standardclient, kann jedoch auf wesentlich mehr verschiedenen Plattformen und in deutlich mehr Netzen ausgeführt werden.
- Ein erweiterter Telemetrieclient agiert als Netzkonzentrator, um noch mehr MQTT-Clients mit einem einzelnen Warteschlangenmanager zu verbinden. Er kann außerdem ein Store-and-forward-Verfahren für kompakte Endgeräte zur Verfügung stellen, die bei kurzen Netzausfallzeiten Nachrichten nicht puffern können.
- IBM WebSphere MQ Telemetry daemon for devices ist ein erweiterter Telemetrieclient, der als eigene Komponente in IBM WebSphere MQ Telemetry bereitgestellt wird. Weitere Informationen finden Sie im Abschnitt „Telemetry-Dämon für Geräte“ auf Seite 104.
- MQTT ist ein Publish/Subscribe-Protokoll:
  - Eine MQTT-Clientanwendung kann Nachrichten auf einem MQTT-Server veröffentlichen.
  - Wenn ein IBM WebSphere MQ -Warteschlangenmanager als MQTT-Server agiert, können andere Anwendungen, die eine Verbindung zum Warteschlangenmanager herstellen, die Nachrichten vom MQTT-Client abonnieren und empfangen.
  - Ein MQTT-Client kann Nachrichten abonnieren, die von Anwendungen gesendet werden, die mit einem MQ-Warteschlangenmanager verbunden sind.
  - Der Warteschlangenmanager agiert als Router, der Nachrichten von veröffentlichenden Anwendungen an abonnierende Anwendungen verteilt.
  - Die Nachrichten können zwischen verschiedenen Arten von Clientanwendungen verteilt werden. Dies ist beispielsweise zwischen Telemetrieclients und JMS-Clients möglich.

IBM WebSphere MQ Telemetry ersetzt die SCADA-Knoten, die in Version 7 von WebSphere Message Broker zurückgezogen wurden und unter Windows, Linux und AIX ausgeführt werden können. Die Migration von Telemetrieanwendungen von WebSphere Message Broker Version 6 auf IBM WebSphere MQ Telemetry und WebSphere Message Broker Version 7.0 enthält Informationen, die Sie bei der Migration von Anwendungen von der Verwendung der SCADA-Knoten in WebSphere Message Broker V6 unterstützen. Telemetrieanwendungen, die WebSphere Message Broker Version 7 verwenden, abonnieren Themen, die für MQTT-Clients gelten. Sie empfangen Veröffentlichungen von MQTT-Clients mithilfe von MQInput-Knoten und veröffentlichen Informationen für MQTT-Clients über Veröffentlichungsknoten.

## Zugehörige Konzepte

„Telemetrikonzepte und -szenarios im Bereich der Überwachung und Steuerung“ auf Seite 74

Telemetrie ist die automatisierte Erkennung und Messung von Daten ferner Geräte sowie die Steuerung dieser fernen Geräte. Der Schwerpunkt liegt in der Datenübertragung von fernen Geräten zu einem zentralen Kontrollpunkt. Die Telemetrie beinhaltet auch das Senden von Konfigurations- und Steuerinformationen an Einheiten.

### **Zugehörige Tasks**

[WebSphere MQ Telemetry installieren](#)

[WebSphere MQ Telemetry verwalten](#)

[Migration von Telemetrieapplikationen von einer Verwendung von WebSphere Message Broker Version 6 hin zur Verwendung von WebSphere MQ Telemetry und WebSphere Message Broker Version 7.0](#)

[Anwendungen für WebSphere MQ Telemetry entwickeln](#)

[Fehlerbehebung für WebSphere MQ Telemetry](#)

### **Zugehörige Verweise**

[WebSphere MQ Telemetry-Referenz](#)

### **Zugehörige Informationen**

[„IBM WebSphere MQ Telemetry“ auf Seite 72](#)

## **IBM WebSphere MQ Version 7.5 - Literaturübersicht**

---

Die Literaturübersicht enthält Links zu einer Vielzahl von IBM WebSphere MQ Version 7.5-Ressourcen.

Diese Übersicht fasst Informationen aus verschiedenen Quellen zusammen, damit Sie mehr über einen bestimmten Bereich von IBM WebSphere MQ in Erfahrung bringen können. Klicken Sie auf die Links in den einzelnen Abschnitten der Übersicht, um zu sehen, welche Ressourcen verfügbar sind.

- [Produktübersicht](#)
- [Technische Übersicht](#)
- [Szenarios](#)
- [Planung](#)
- [Migration und Upgrades](#)
- [Installation](#)
- [Sicherheit](#)
- [Konfiguration](#)
- [Verwaltung](#)
- [Anwendungsentwicklung](#)
- [Überwachung und Leistung](#)
- [Fehlerbehebung und Support](#)
- [Referenzinformationen](#)

Tabelle 1. Tabelle für IBM WebSphere MQ -Literaturübersicht



Kategorie	Informationsquellen
Produktübersicht	<p>Übersicht über den allgemeinen Zweck, die Leistungsmerkmale und neue Funktionen von IBM WebSphere MQ</p> <p> <b>„Informationen zu IBM WebSphere MQ“ auf Seite 5</b>  Die Informationen in diesem Abschnitt helfen Ihnen beim Einstieg in IBM WebSphere MQ Version 7.5. Sie erhalten hier eine Einführung in das Produkt sowie einen Überblick über die Neuerungen und Änderungen in dieser Version.</p> <p><b><u>IBM WebSphere MQ Version 7.5</u></b></p> <p>Die IBM Redbooks-Veröffentlichung deckt die wichtigsten funktionalen Erweiterungen in IBM WebSphere MQ Version 7.5 und die Konzepte ab, mit denen Sie vertraut sein müssen.</p> <p><b><u>IBM WebSphere MQ-Produktwebseite</u></b>  Produktwebseite mit Links zu Ressourcen und zusätzlichen Informationen</p> <p><b><u>IBM WebSphere MQ Systemvoraussetzungen</u></b>  Webseite mit Links zu den Systemvoraussetzungen für die verschiedenen Releases von IBM WebSphere MQ.</p> <p><b>„IBM WebSphere MQ Version 7.5 in der App IBM Documentation Offline“ auf Seite 19</b>  Sie können die Messaging-Dokumentation zu IBM WebSphere MQ Version 7.5 als lokal installierte Offlineversion des IBM Documentation herunterladen.</p>
Technische Übersicht	<p> <b>„IBM WebSphere MQ - Technische Übersicht“ auf Seite 44</b></p> <p>Hier finden Sie Informationen zur Steuerung von Nachrichtenwarteschlangen und zu anderen Funktionen, die IBM WebSphere MQ Version 7.5 bereitstellt.</p>



Tabelle 1. Tabelle für IBM WebSphere MQ -Literaturübersicht (Forts.)

Kategorie	Informationsquellen
Szenarios	<p>Jedes Szenario führt Sie durch eine Reihe von Aufgaben und hilft Ihnen bei der Konfiguration einer wichtigen Produktfunktion. Die Szenarios enthalten hilfreiche Links zu anderen Inhalten, die Ihnen ein besseres Verständnis zu dem Bereich vermitteln, für den Sie sich interessieren.</p> <p><b>❄ „Szenarien“ auf Seite 151</b> Das Szenario <i>Erste Schritte</i> beschreibt die ersten Schritte mit IBM WebSphere MQ. Verwenden Sie dieses Szenario, wenn IBM WebSphere MQ für Sie neu ist und Sie möglichst schnell damit arbeiten möchten. Weitere Szenarios helfen Ihnen bei der Konfiguration oder Verwendung der Produktfunktionen, indem sie Sie durch die entsprechenden Aufgabenschritte führen.</p> <p><b>❄ <u>WebSphere Application Server mit IBM WebSphere MQverbinden</u></b> Enthält Informationen, die Sie durch die wichtigsten Aufgaben zur Herstellung einer Verbindung von WebSphere Application Server mit IBM WebSphere MQ in einer Vielzahl von Szenarios führen.</p> <p><b>❄ <u>WebSphere Application Server Liberty Profile mit IBM WebSphere MQverbinden</u></b> Enthält Informationen, die Sie durch die wichtigsten Aufgaben zur Herstellung einer Verbindung eines WebSphere Application Server Liberty-Profiles mit IBM WebSphere MQ in einer Vielzahl von Szenarios führen.</p> <p><b>❄ <u>IBM MessageSight mit IBM WebSphere MQ und WebSphere Application Serververbinden</u></b> Enthält Informationen, die Sie durch die wichtigsten Tasks führen, die erforderlich sind, um eine Verbindung zwischen IBM MessageSight und IBM WebSphere MQ und WebSphere Application Server in verschiedenen Szenarios herzustellen.</p>
Planung	<p><b>❄ <u>Planung</u></b> Beachten Sie bei der Planung einer IBM WebSphere MQ-Umgebung die Unterstützung, die IBM WebSphere MQ für Architekturen mit einzelnen oder mehreren Warteschlangenmanagern sowie für Punkt-zu-Punkt- und Publish/Subscribe-Messaging bereitstellt. Planen Sie auch den Ressourcenbedarf und die Nutzung von Protokollierungs- und Sicherheitsfunktionen.</p>
Migration und Upgrades	<p><b>❄ <u>Migration und Upgrades</u></b> Um einen Warteschlangenmanager auf eine neue Codeversion zu migrieren, müssen Sie zunächst ein Upgrade für IBM WebSphere MQ durchführen, um die neue Codeversion zu installieren. Nachdem Sie überprüft haben, dass die Aktualisierung erfolgreich war, können Sie den Warteschlangenmanager und alle zugehörigen Anwendungen und Ressourcen migrieren. Erstellen Sie, bevor Sie diesen Prozess starten, einen Migrationsplan anhand der Informationen in dieser Migrationsanleitung. Wenn Sie eine Wartung durchführen, ist keine Migration erforderlich. Sie sollten Anwendungen jedoch mit der neuen Version des IBM WebSphere MQ -Codes testen.</p> <p><b>❄ <u>WebSphere / IBM MQ Migrationshandbuch</u></b> Dieses Handbuch enthält Informationen, die Ihnen bei der Planung der Migration von einer früheren auf eine neue Produktversion helfen. Sie können das Handbuch entweder im Web-Browser aufrufen oder als PDF-Datei herunterladen.</p>

Tabelle 1. Tabelle für IBM WebSphere MQ -Literaturübersicht (Forts.)

Kategorie	Informationsquellen
Installation	<p>❄ <b><u>Installieren und Deinstallieren</u></b>                      Hier finden Sie Informationen, die Ihnen bei der Vorbereitung der Installation, bei der Installation des Produkts und bei der Überprüfung der Installation helfen. Sie finden auch Informationen zur Deinstallation des Produkts.</p>
Sicherheit	<p>❄ <b><u>Sicherheit</u></b>                      Sicherheitsaspekte, die in einer IBM WebSphere MQ-Installation zu beachten sind, einschließlich Identifikation und Authentifizierung, Berechtigung, Protokollierung, Vertraulichkeit und Datenintegrität.</p>
Konfiguration	<p>❄ <b><u>Konfiguration</u></b>                      Erstellen Sie einen oder mehrere Warteschlangenmanager auf einem oder mehreren Computern und konfigurieren Sie sie und die zugehörigen Ressourcen auf Ihren Entwicklungs-, Test- und Produktionssystemen für die Verarbeitung von Nachrichten mit Ihren Geschäftsdaten.</p>
Verwaltung	<p>❄ <b><u>Verwaltung IBM WebSphere MQ</u></b>                      Verwalten Sie Ihre Warteschlangenmanager und die zugehörigen Ressourcen.</p>
Anwendungs-entwicklung	<p>❄ <b><u>Anwendungen entwickeln</u></b>                      Entwickeln Sie Anwendungen zum Senden und Empfangen von Nachrichten und zum Verwalten Ihrer Warteschlangenmanager und der zugehörigen Ressourcen. IBM WebSphere MQ unterstützt Anwendungen, die in prozeduralen Programmiersprachen sowie objektorientierten Sprachen und Frameworks geschrieben sind.</p>
Überwachung und Leistung	<p>❄ <b><u>Überwachung und Leistung</u></b>                      Hier finden Sie Überwachungsinformationen und -anweisungen zur Verbesserung der Leistung eines Warteschlangenmanagernetzes und Optimierungstipps zur Verbesserung der Leistung des Warteschlangenmanagernetzes.</p>
Fehlerbehebung und Support	<p>❄ <b><u>Fehlerbehebung und Unterstützung</u></b>                      Hier finden Sie Informationen zu Verfahren, die Ihnen bei der Diagnose und Behebung von Problemen mit Ihrem Warteschlangenmanagernetz und Ihren IBM WebSphere MQ-Anwendungen helfen.</p> <p><b><u>IBM Support Assistant Webseite</u></b>                      IBM Support Assistant (ISA) hilft Ihnen bei der Lösung von Fragen und Problemen mit IBM Softwareprodukten durch die Bereitstellung von Zugriff auf Supportinformationen und Fehlerbehebungstools.</p> <p><b><u>IBM Support-Portal-Webseite</u></b>                      IBM Support Portal für IBM WebSphere MQ.</p>
Referenzinformationen	<p>❄ <b><u>Referenz</u></b>                      Referenzinformationen für Konfiguration, Verwaltung, Entwicklungsanwendungen, Telemetrie, Sicherheit, Überwachung, Fehlerbehebung und Support sowie Diagnose-nachrichten</p>

# IBM WebSphere MQ Version 7.5 in der App IBM Documentation Offline

Wenn Sie sich in einer Airgap-Umgebung ohne Zugriff auf das Internet befinden, verwenden Sie unsere Dark Shop-App "IBM Documentation Offline", um Downloads der IBM WebSphere MQ Version 7.5 -Produktdokumentation anzuzeigen.

IBM Documentation Offline verfügt über zwei Komponenten:

- **Die IBM Documentation Offline -App.** Dies ist eine lokal installierbare Offlineversion des IBM Documentation.
- **Die Dokumentationspakete, die Sie in der IBM Documentation Offline -App installieren.** Diese Pakete enthalten dieselbe Dokumentation, die auch online im IBM Documentation veröffentlicht ist.

Um die App und das IBM WebSphere MQ Version 7.5 -Dokumentationspaket herunterzuladen, müssen Sie sich bei IBM Documentation anmelden. Weitere Informationen finden Sie unter [IBM Documentation Offline](#).

## PDF-Dokumentation zu IBM WebSphere MQ Version 7.5

Die Produktdokumentation zu IBM WebSphere MQ Version 7.5 kann in einer Reihe von PDF-Dateien heruntergeladen werden.

Die IBM WebSphere MQ Version 7.5 -PDF-Dateien enthalten dieselben Informationen wie die in IBM Documentation veröffentlichte HTML-Version. Sie können diese Dateien durch Klicken auf die Links in der folgenden Tabelle herunterladen.

**Anmerkung:** Sie können die IBM WebSphere MQ Version 7.5-Produktdokumentation auch im HTML-Format in eine Offline-Version des IBM Documentation herunterladen, die Sie lokal installieren. Weitere Informationen finden Sie im Abschnitt „IBM WebSphere MQ Version 7.5 in der App IBM Documentation Offline“ auf Seite 19.

<b>PDF-Dateiname und Download-Link</b>	<b>Äquivalenter Abschnitt in IBM Documentation</b>
<a href="#">mq75.overview.pdf</a>	Informationen zu IBM WebSphere MQ
<a href="#">mq75.plan.pdf</a>	Planung
<a href="#">mq75.migrate.pdf</a>	Migration und Upgrades
<a href="#">mq75.install.pdf</a>	Installieren
<a href="#">mq75.secure.pdf</a>	Sicherheit
<a href="#">mq75.configure.pdf</a>	Konfiguration
<a href="#">mq75.administer.pdf</a>	Verwalten
<a href="#">mq75.develop.pdf</a>	Anwendungen entwickeln
<a href="#">mq75.monitor.pdf</a>	Überwachung und Leistung
<a href="#">mq75.troubleshoot.pdf</a>	Fehlerbehebung und Unterstützung
<a href="#">mq75.reference.pdf</a>	Referenz
<a href="#">mq75.refcon.pdf</a>	Konfigurationsreferenz
<a href="#">mq75.refadmin.pdf</a>	Referenzinformationen zur Verwaltung
<a href="#">mq75.refdev.pdf</a>	Referenzinformationen zum Entwickeln von Anwendungen

Tabelle 2. PDF-Dateien für die Produktdokumentation und die Übereinstimmung mit den Abschnitten in IBM Documentation (Forts.)

PDF-Dateiname und Download-Link	Äquivalenter Abschnitt in IBM Documentation
<a href="#">mq75.mft.pdf</a>	IBM WebSphere MQ Managed File Transfer
<a href="#">mq75.explorer.pdf</a>	IBM WebSphere MQ Explorer
<a href="#">mq75.guiwizard.pdf</a>	Hilfe zum Assistenten für die grafische Benutzerschnittstelle von MQ
<a href="#">mq75.xms.pdf</a>	Message Service Client for .NET
<a href="#">mq75.hve.pdf</a>	WebSphere MQ Hypervisor-Editionen
<a href="#">mq75.mmtc.pdf</a>	Mobile Messaging and M2M
<a href="#">MQ_Migration_Guide.pdf</a>	WebSphere / IBM MQ Migrationshandbuch

**Anmerkung:** Damit die Links zwischen den PDF-Dateien funktionieren, müssen sich die PDF-Dateien im gleichen Ordner befinden.

## Neuerungen in IBM WebSphere MQ Version 7.5

In diesem Abschnitt werden die wichtigsten neuen Funktionen in IBM WebSphere MQ Version 7.5 beschrieben.

- „IBM WebSphere MQ Managed File Transfer“ auf Seite 20.
- „IBM WebSphere MQ Erweiterte Nachrichtensicherheit“ auf Seite 20.
- „Überwachung des Nachrichtenkanalagenten (MCA)“ auf Seite 21.
- „Mehrere Clusterübertragungswarteschlangen“ auf Seite 21.
- „Erweiterte transaktionsorientierte Funktionalität ist jetzt Teil des Kernclients“ auf Seite 21.
- „Verbindung mit einem Warteschlangenmanager durch Festlegung eines Anwendungsnamens angeben“ auf Seite 21.
- „Richtlinien zur Zertifikatsprüfung“ auf Seite 21.
- „Mehr Transparenz bei transaktionsorientierter Verarbeitung“ auf Seite 21.
- „Szenarios“ auf Seite 21.
- „IBM WebSphere MQ Explorer“ auf Seite 22.

### IBM WebSphere MQ Managed File Transfer

IBM WebSphere MQ Managed File Transfer verwendet IBM WebSphere MQ zur Übertragung von Dateien zwischen Warteschlangenmanagern. Sie können seine Reichweite auf Workstations und Server ausdehnen, die keinen Warteschlangenmanager besitzen. Sie können es auf die Verwendung von Dateiübertragungsagenten und Apache Ant erweitern und mit IBM Sterling Commerce:Direct, Web Gateways und SFTP-, FTP- oder FTPS-Protokollservern integrieren.

Mit IBM WebSphere MQ Managed File Transfer können Sie die Übertragung von Dateien automatisieren, steuern, schützen und prüfen (siehe [IBM WebSphere MQ Managed File Transfer-Einführung](#)).

### IBM WebSphere MQ Erweiterte Nachrichtensicherheit

IBM WebSphere MQ Advanced Message Security (AMS) ist eine separat installierte Komponente, die separat gebührenpflichtig ist. Es bietet ein hohes Maß an Schutz für sensible Daten, die über das IBM WebSphere MQ-Netz ausgetauscht werden. Sie müssen bestehende Anwendungen nicht ändern, um die Vorteile von AMS nutzen zu können (siehe [IBM WebSphere MQ Advanced Message Security](#)).

## Überwachung des Nachrichtenkanalagenten (MCA)

Die MCA-Abfangfunktion ermöglicht einem Warteschlangenmanager, der unter IBM WebSphere MQ mit einer lizenzierten Installation von Erweiterte Nachrichtensicherheit ausgeführt wird, die selektive Aktivierung von Richtlinien für Serververbindungskanäle. Durch MCA-Abfangen können Clients, die außerhalb von IBM WebSphere MQ AMS bleiben, immer noch mit einem Warteschlangenmanager verbunden sein und ihre Nachrichten verschlüsselt und entschlüsselt werden. Siehe [Abfangen durch den Nachrichtenkanalagenten \(MCA\)](#).

## Mehrere Clusterübertragungswarteschlangen

Durch die Änderung des neuen Warteschlangenmanager-Attributs **DEFCLXQ** können Sie jedem Cluster-senderkanal eine andere Clusterübertragungswarteschlange zuordnen. Die von den einzelnen Cluster-senderkanälen weiterzuleitenden Nachrichten werden in separate Clusterübertragungswarteschlangen gestellt (siehe [Clusterübertragungswarteschlangen und Clustersenderkanäle](#)). Sie können Clusterübertragungswarteschlangen auch manuell konfigurieren, indem Sie das neue Warteschlangenattribut **CLCHNAME** festlegen. Sie können festlegen, welche Clustersenderkanäle welche Übertragungswarteschlangen gemeinsam nutzen, welche über separate Übertragungswarteschlangen verfügen und welche die Clusterübertragungswarteschlange(n) verwenden (siehe [Clustering: Konfiguration von Clusterübertragungswarteschlangen planen](#)). Die Änderung unterstützt Systemadministratoren bei der Verwaltung von Nachrichten, die zwischen Warteschlangenmanagern, die einem Cluster angehören, ausgetauscht werden.

## Erweiterte transaktionsorientierte Funktionalität ist jetzt Teil des Kernclients

Die erweiterte transaktionsorientierte Funktionalität ist jetzt in den IBM WebSphere MQ-Kernclient integriert. Sie müssen keine separate Lizenz für einen erweiterten transaktionsorientierten Client erwerben und keine separate erweiterte transaktionsorientierte Clientkomponente installieren (siehe [„Was ist ein erweiterter Transaktionsclient?“](#) auf Seite 144).

## Verbindung mit einem Warteschlangenmanager durch Festlegung eines Anwendungsnamens angeben

Eine Anwendung kann einen Namen festlegen, der die Verbindung zum Warteschlangenmanager angibt. Zeigen Sie den Anwendungsnamen mit dem Befehl **DISPLAY CONN** an. Der Name wird im Feld **APPL-TAG** zurückgegeben. Sie können den Namen auch im IBM WebSphere MQ Explorer-Fenster **Application Connections** (Anwendungsverbindungen) anzeigen. Das Feld heißt **App name**; siehe [Einrichten der WebSphere MQ -Umgebung für WebSphere MQ -Klassen für Java](#). Sie können den Namen einer Anwendungs-Verbindung auf allen Plattformen außer z/OS festlegen.

## Richtlinien zur Zertifikatsprüfung

Unter UNIX, Linux, and Windows können Sie angeben, wie streng die Validierung der Zertifikatskette dem Branchensicherheitsstandard RFC 5280 entspricht (siehe [Zertifikatsprüfrichtlinien in WebSphere MQ](#)).

## Mehr Transparenz bei transaktionsorientierter Verarbeitung

Für den Befehl `dspmqrtrn` gibt es die beiden neuen Parameter **-a** und **-q**, um weitere Informationen bereitzustellen, wenn ein asynchroner Rollback auftritt. Zwei neue Nachrichten, [AMQ7486](#) und [AMQ7487](#), enthalten Informationen zu der Transaktion, die zurückgesetzt wird, und dazu, ob die Transaktion einer Verbindung zugeordnet ist.

## Szenarios

„Szenarios“ auf Seite 151 zeigen, wie neue IBM WebSphere MQ Version 7.5-Funktionen schneller und einfacher verwendet und kombiniert werden können. Die Szenarios enthalten hilfreiche Links zu Inhalten der Produktdokumentation, die Ihnen ein besseres Verständnis zu dem Themenbereich vermitteln, für den Sie sich interessieren.

## IBM WebSphere MQ Explorer

Zu den neuen Funktionen in IBM WebSphere MQ Explorer Version 7.5 gehören die Integration von IBM WebSphere MQ Managed File Transfer und IBM WebSphere MQ Advanced Message Security, eine verbesserte Unterstützung mehrerer Versionen und ein geringerer Systemaufwand.

Weitere Informationen zu Neuerungen in IBM WebSphere MQ Explorer finden Sie unter [Neuerungen und Änderungen in WebSphere MQ Explorer](#).

### Zugehörige Konzepte

„[Neuerungen in IBM WebSphere MQ Version 7.5](#)“ auf Seite 25

„[Änderungen in IBM WebSphere MQ Version 7.5 -Fixpacks](#)“ auf Seite 26

In diesem Abschnitt werden Änderungen der Funktionen und Ressourcen in Fixpacks von Version 7.5 beschrieben.

## IBM WebSphere MQ-Client für HP Integrity NonStop Server

IBM WebSphere MQ unterstützt jetzt den Client für die Plattform HP Integrity NonStop Server.

### Übersicht

Eine Übersicht über IBM WebSphere MQ-Clients, einschließlich des Clients für die Plattform HP Integrity NonStop Server, finden Sie im Abschnitt [„Überblick über IBM WebSphere MQ MQI-Clients“](#) auf Seite 140.

Eine technische Übersicht über den IBM WebSphere MQ-Client für die Plattform HP Integrity NonStop Server finden Sie im Abschnitt [„IBM WebSphere MQ-Client für HP Integrity NonStop Server - Technische Übersicht“](#) auf Seite 150.

Details zu Umgebungen und Funktionen, die vom IBM WebSphere MQ-Client für HP Integrity NonStop Server unterstützt werden, finden Sie im Abschnitt [„IBM WebSphere MQ-Client für HP Integrity NonStop Server-unterstützte Umgebungen und Funktionen“](#) auf Seite 150.

### Planung

Wenn Sie Hilfe bei der Planung Ihres IBM WebSphere MQ-Clients für die HP Integrity NonStop Server-Umgebung benötigen, lesen Sie den Abschnitt [Planung der IBM WebSphere MQ-Clientumgebung unter HP Integrity NonStop Server](#).

### Installieren

Hilfe bei der Installation des IBM WebSphere MQ-Clients für HP Integrity NonStop Server:

- Auswahl der Installationskomponenten - siehe [IBM WebSphere MQ-Clientkomponenten für HP Integrity NonStop Server](#).
- Planung der Installation - siehe [Installation auf HP Integrity NonStop Server planen](#).
  - [Dateisystem](#)
- Hardware- und Softwarevoraussetzungen - siehe [Hardware- und Softwarevoraussetzungen auf HP Integrity NonStop Server-Systemen](#).
- Überprüfung auf richtige Software - siehe [Systemsoftwarevoraussetzungen überprüfen](#)
- Vorbereitung Ihres Systems - siehe [Benutzer und Gruppe unter HP Integrity NonStop Server einrichten](#).
- Informationen zur Installation des Clients finden Sie unter [IBM WebSphere MQ -Client auf HP Integrity NonStop Server -Systemen installieren](#).
- Überprüfung der Installation - siehe [Clientinstallation überprüfen](#)
- Deinstallation - siehe [IBM WebSphere MQ unter HP Integrity NonStop Server deinstallieren](#).

## HP Integrity NonStop Server-Clientbefehle

Folgende Befehle sind auf den IBM WebSphere MQ-Client für OSS- und Guardian-Umgebungen von HP Integrity NonStop Server anwendbar:

- [dspmqver](#)
- [endmqtrc](#)
- [mqrc \(MQ-Rückkehrcode\)](#)
- [runmqgras](#)
- [runmqtmc](#)
- [strmqtrc](#)

Der folgende Befehl ist auf den IBM WebSphere MQ-Client für die OSS-Umgebung von HP Integrity NonStop Server anwendbar:

- [dspmqtrc](#)

Neue Produkt-ID MQNC wurde zum Befehl [DISPLAY CHSTATUS](#) hinzugefügt, Tabelle [Werte für Produkt-ID](#).

## Sicherheit

Die folgenden Abschnitte enthalten Informationen zur Sicherheit des IBM WebSphere MQ-Clients für die HP Integrity NonStop Server-Umgebung:

- Informationen dazu, wie sich der IBM WebSphere MQ -Client für HP Integrity NonStop Server beim Warteschlangenmanager identifiziert, die zu [Authentifizierung für eine Clientanwendung planen](#) hinzugefügt wurden.
- [Sicherheit unter HP Integrity NonStop Server einrichten](#)
  - [OpenSSL](#)
  - [Entropiedämon](#)
- [IBM WebSphere MQ-Unterstützung für SSL und TLS](#)
- [Unter HP Integrity NonStop Server mit SSL oder TLS arbeiten](#)
  - [Zertifikatsmanagement](#)
  - [Persönlicher Zertifikatsspeicher](#)
  - [Zertifikat-Truststore](#)
  - [Kennphrasen-Stashdatei](#)
  - [Zertifikatswiderrufslistendatei](#)

## Transaction Management Facility

Die folgenden Abschnitte enthalten Informationen zur Transaction Management Facility (TMF):

- [IBM WebSphere MQ -Clientumgebung unter HP Integrity NonStop Server planen](#)
  - [HP Integrity NonStop Server-Umgebung vorbereiten](#)
  - [IBM WebSphere MQ und HP NonStop TMF](#)
  - [HP NonStop TMF verwenden](#)
    - [Globale Arbeitseinheiten verwenden](#)
    - [Transaktionen mit langer Laufzeit vermeiden](#)
    - Der Abschnitt [Globale Arbeitseinheiten beenden](#) wurde um Informationen zur Warteschlangenmanagerkonfiguration bezüglich des Ablaufs globaler Arbeitseinheiten nach einem vorkonfigurierten Inaktivitätsintervall erweitert.
- [HP Integrity NonStop Server](#)

- [Übersicht über den Gateway-Prozess](#)
- [Gateway für die Ausführung unter Pathway konfigurieren](#)
- [TMF- und TMF/Gateway-Zeilengruppen](#)
- [Clientinitialisierungsdatei konfigurieren](#)
- [Kanälen Berechtigungen erteilen](#)
- [VerwaltungHP Integrity NonStop Server](#)
  - [TMF-Gateway manuell aus Pathway starten](#)
  - [TMF-Gateway manuell aus Pathway stoppen](#)
- [Fehlerbehebung für IBM WebSphere MQ -Client für HP Integrity NonStop Server](#)

## Anwendungen entwickeln

Informationen zur Entwicklung von Anwendungen für den IBM WebSphere MQ-Client auf der Plattform HP Integrity NonStop Server finden Sie in folgenden Abschnitten:

- [Anwendung auf HP Integrity NonStop Server erstellen](#)
  - [OSS- und Guardian-Header und öffentliche Bibliotheken](#)
  - [C-Programme in HP Integrity NonStop Server vorbereiten](#)
  - [COBOL-Programme vorbereiten](#)
  - [pTAL-Programme vorbereiten](#)
- [Codierung in pTAL.](#)
- [JMS-Programme für den IBM WebSphere MQ-Client für HP Integrity NonStop Server vorbereiten.](#)

## Neue Nachrichten

Es gibt folgende neue Nachrichten für den IBM WebSphere MQ-Client für HP Integrity NonStop Server:

- [AMQ5000-5999: Installierbare Services](#)
  - [AMQ5370](#)
  - [AMQ5371](#)
  - [AMQ5372](#)
  - [AMQ5373](#)
  - [AMQ5374](#)
  - [AMQ5375](#)
  - [AMQ5376](#)
  - [AMQ5377](#)
  - [AMQ5378](#)
  - [AMQ5379](#)
  - [AMQ5380](#)
  - [AMQ5390](#)
  - [AMQ5391](#)
  - [AMQ5392](#)
  - [AMQ5393](#)
  - [AMQ5394](#)
  - [AMQ5395](#)
  - [AMQ5396](#)
  - [AMQ5397](#)



- [AMQ5398](#)
- [AMQ5399](#)
- AMQ9000-9999: Fernzugriff
  - [AMQ9816](#)
  - [AMQ9817](#)
  - [AMQ9818](#)
  - [AMQ9819](#)
  - [AMQ9820](#)
  - [AMQ9821](#)
  - [AMQ9823](#)
  - [AMQ9824](#)

### Geänderte API-Ursachencodes

Die folgenden bestehenden API-Ursachencodes schließen jetzt HP Integrity NonStop Server ein:

- [2354 \(0932\) \(RC2354\): MQRC\\_UOW\\_ENLISTMENT\\_ERROR](#)
- [2355 \(0933\) \(RC2355\): MQRC\\_UOW\\_MIX\\_NOT\\_SUPPORTED](#)
- [2072 \(0818\) \(RC2072\): MQRC\\_SYNCPOINT\\_NOT\\_AVAILABLE](#)
- [2003 \(07D3\) \(RC2003\): MQRC\\_BACKED\\_OUT](#)

### Beispielanwendungen

Weitere Informationen zu den von den Beispielprogrammen veranschaulichten Verfahren finden Sie im Abschnitt [Beispiele für IBM WebSphere MQ-Client für HP Integrity NonStop Server](#).

### Fehlerbehebung und Unterstützung

Die folgenden Abschnitte enthalten Fehlerbehebungs- und Unterstützungsinformationen für den IBM WebSphere MQ-Client für HP Integrity NonStop Server:

- [Fehlerbehebung für IBM WebSphere MQ -Client für HP Integrity NonStop Server](#)
- [Fehlerprotokolle unter HP Integrity NonStop Server](#)
- [Trace unter HP Integrity NonStop Server verwenden](#)
- [FFST: IBM WebSphere MQ für HP Integrity NonStop Server](#)

## Neuerungen in IBM WebSphere MQ Version 7.5

---

Überprüfen Sie die Liste der Änderungen sorgfältig, bevor Sie ein Upgrade der Warteschlangenmanager auf IBM WebSphere MQ Version 7.5 durchführen. Überlegen Sie, ob Sie bestehende Anwendungen, Scripts und Verfahren ändern müssen, bevor Sie mit der Migration Ihrer Systeme auf Version 7.5 beginnen.

Die folgenden Links führen zu Informationen im Abschnitt Migration und Upgrades der Produktdokumentation. Neue Funktionen und Änderungen, die keine Auswirkungen auf bestehende Anwendungen, Verwaltungsverfahren und Verwaltungsscripts haben, sind hier nicht aufgeführt. Beachten Sie hierzu den Abschnitt [„Neuerungen in IBM WebSphere MQ Version 7.5“](#) auf Seite 20.

[Liste der Änderungen nach Versions-, Release- und Wartungsstufe](#)

- [Änderungen zwischen Version 7.1 und Version 7.5](#)

### Zugehörige Konzepte

[„Neuerungen in IBM WebSphere MQ Version 7.5“](#) auf Seite 20

In diesem Abschnitt werden die wichtigsten neuen Funktionen in IBM WebSphere MQ Version 7.5 beschrieben.

„Änderungen in IBM WebSphere MQ Version 7.5 -Fixpacks“ auf Seite 26

In diesem Abschnitt werden Änderungen der Funktionen und Ressourcen in Fixpacks von Version 7.5 beschrieben.

## Änderungen in IBM WebSphere MQ Version 7.5 -Fixpacks

---

In diesem Abschnitt werden Änderungen der Funktionen und Ressourcen in Fixpacks von Version 7.5 beschrieben.

- **V7.5.0.1** „Version 7.5.0, Fix Pack 1: Unterstützung für MQTT over WebSockets“ auf Seite 27
- **V7.5.0.4** „Version 7.5.0, Fix Pack 4: Inaktivieren Sie IBM WebSphere MQ Advanced Message Security auf dem Client mithilfe von AMQ\_DISABLE\_CLIENT\_AMS“ auf Seite 28
- **V7.5.0.5** „Version 7.5.0, Fix Pack 5: Inaktivieren Sie IBM WebSphere MQ Advanced Message Security auf dem Client mithilfe der Datei mqclient.ini.“ auf Seite 28
- **Windows** **V7.5.0.6** „Version 7.5.0, Fix Pack 6: Benutzer-ID und Kennwort für verwaltete .NET-Clientanwendungen“ auf Seite 28
- **V7.5.0.6** „Version 7.5.0, Fix Pack 6: Nicht weiter unterstützte CipherSpecs“ auf Seite 28
- **V7.5.0.6** „Version 7.5.0, Fix Pack 6: Erweiterungen für die Wartungsfreundlichkeit für IBM WebSphere MQ Managed File Transfer“ auf Seite 29
- **V7.5.0.6** „Version 7.5.0, Fix Pack 6 Neue IBM WebSphere MQ Managed File Transfer-Agenteneigenschaft failTransferOnFirstFailure“ auf Seite 29
- **V7.5.0.7** „Version 7.5.0, Fix Pack 7: Nicht weiter unterstützte CipherSpecs“ auf Seite 29
- **Windows** **V7.5.0.7** „Version 7.5.0, Fix Pack 7: Unterstützung für JDBC-Verbindungen zu einer Oracle 12c-Datenbank“ auf Seite 30
- **Windows** **V7.5.0.7** „Version 7.5.0, Fix Pack 7: Entfernung der Einschränkung bei Verwendung von .NET für MQCNO\_CLIENT\_BINDING und MQCNO\_LOCAL\_BINDING“ auf Seite 30
- **V7.5.0.8** „Version 7.5.0, Fix Pack 8: Einschränkung bei Verwendung von Topic-Aliaswarteschlangen in Verteilerlisten“ auf Seite 30
- **V7.5.0.8** „Version 7.5.0, Fix Pack 8: GSKit-Version aktualisiert“ auf Seite 30
- **V7.5.0.8** „Version 7.5.0, Fix Pack 8: Nicht weiter unterstützte CipherSpecs“ auf Seite 30
- **V7.5.0.8** „Version 7.5.0, Fix Pack 8: Neue Konstante JMS\_IBM\_SUBSCRIPTION\_USER\_DATA zur JmsConstants-Schnittstelle hinzugefügt“ auf Seite 31
- **V7.5.0.8** „Version 7.5.0, Fix Pack 8: Aktualisierungen des JMS-Listeners für Ausnahmbedingungen“ auf Seite 31
- **V7.5.0.8** „Version 7.5.0, Fix Pack 8: Unterstützung für Zulassungslisten für Klassennamen in IBM WebSphere MQ classes for JMS ObjectMessage“ auf Seite 31
- **V7.5.0.8** „Version 7.5.0, Fix Pack 8: Neue IBM WebSphere MQ Managed File Transfer-Agenteneigenschaft additionalWildcardSandboxChecking“ auf Seite 32
- **V7.5.0.8** „Version 7.5.0, Fix Pack 8: Änderung des Verhaltens des Befehls IBM WebSphere MQ Managed File Transfer fteCleanAgent“ auf Seite 32
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Neue Umgebungsvariable AMQ\_SSL\_LDAP\_SERVER\_VERSION“ auf Seite 32

- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Beispielkonfigurationsdatei für IBM WebSphere MQ classes for JMS“ auf Seite 32
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Änderungen am Befehl dmpmqcfcg“ auf Seite 32
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Änderungen am Befehl strmqm“ auf Seite 33
- **V7.5.0.9** „Funktionale Erweiterungen des Dienstprogramms 'runmqras'“ auf Seite 33
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Abgleichsprüfung von Übertragungs-ID und Wert des Attributs groupId für Übertragungen aus Nachrichten in Dateien wiederhergestellt“ auf Seite 34
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Ressourcenadapteraktualisierungen für Eigenschaft targetClientMatching in einer Aktivierungsspezifikation“ auf Seite 33
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Neuer Parameter -caseno für den Befehl runmqras“ auf Seite 33
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Änderung bei der Behandlung von Fehlern in Verbindung mit beschädigten Zeichendaten innerhalb von internen MQ-Befehlsnachrichten, die von der Warteschlangenmanager-Clustering-Komponente verwendet werden“ auf Seite 33
- **V7.5.0.9** **Windows** „Version 7.5.0, Fix Pack 9: Hinzufügen von Microsoft Visual Studio 2013 C/C++ -Laufzeiten unter Windows“ auf Seite 34
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Abgleichsprüfung von Übertragungs-ID und Wert des Attributs groupId für Übertragungen aus Nachrichten in Dateien wiederhergestellt“ auf Seite 34
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9: Abgleichsprüfung von Übertragungs-ID und Wert des Attributs groupId für Übertragungen aus Nachrichten in Dateien wiederhergestellt“ auf Seite 34
- **V7.5.0.9** „Version 7.5.0, Fix Pack 9 sowie vorläufiger Fix für APAR IT26482: Änderung der Berechtigungen, die erforderlich sind, damit IBM WebSphere MQ classes for JMS die BackoutThreshold und BackoutRequestQueueName einer Clusterwarteschlange abfragen kann“ auf Seite 34

## Version 7.5.0, Fix Pack 1: Unterstützung für MQTT over WebSockets

### **V7.5.0.1**

Es wurde ein neuer Kommunikationsprotokollparameter (PROTOCOL) zur MQTT-Kanaldefinition (DEFINE CHANNEL (MQTT)) hinzugefügt:

- Wenn der Parameter auf MQTTV3 gesetzt wird, akzeptiert der Kanal nur Verbindungen von Clients mit Version 3 des MQ Telemetry-Übertragungsprotokolls. Dies war das einzige Protokoll, das vor IBM WebSphere MQ Version 7.5.0Fixpack 1 unterstützt wurde.
- Wenn der Parameter auf HTTP gesetzt wird, akzeptiert der Kanal nur HTTP-Anforderungen für Seiten oder WebSockets-Verbindungen zu IBM WebSphere MQ Telemetry.
- Wenn der Parameter auf MQTTV3, HTTP gesetzt wird, akzeptiert der Kanal Verbindungen von Clients, die eins dieser beiden Protokolle verwenden. Dies ist das Standardverhalten für neue MQTT-Kanäle, die mit IBM WebSphere MQ Version 7.5.0Fixpack 1 und höheren Versionen erstellt werden.

Weitere Informationen finden Sie im Abschnitt MQTT-Messaging-Client für JavaScript über SSL und WebSockets verbinden.

Wenn ein Client über SSL eine Verbindung zu einem MQTT -Kanal herstellt, bestimmt der Parameter SSLCAUTH, ob IBM WebSphere MQ ein Zertifikat vom Client erfordert (siehe DEFINE CHANNEL (MQTT) ). Vor IBM WebSphere MQ Version 7.5.0Fixpack 1kann dieser Parameter entweder REQUIRED oder OPTIONAL für MQTT -Kanäle sein:

- REQUIRED bedeutet, dass IBM WebSphere MQ ein Zertifikat vom Client anfordert und der Client ein gültiges Zertifikat liefern muss.

- OPTIONAL bedeutet, dass IBM WebSphere MQ ein Zertifikat vom Client anfordert, aber der Client kein Zertifikat liefern muss. Die Clientverbindung wird zugelassen, wenn der Client ein gültiges Zertifikat liefert, aber auch wenn er kein Zertifikat liefert. Die Clientverbindung wird nur dann nicht zugelassen, wenn der Client ein ungültiges Zertifikat liefert.

In IBM WebSphere MQ Version 7.5.0Fixpack 1 und höher kann der Parameter SSLCAUTH auf NEVER für MQTT-Kanäle gesetzt werden. NEVER bedeutet, dass IBM WebSphere MQ nie ein Zertifikat vom Client anfordert. Der neue Wert wurde im Rahmen der Unterstützung für Clients hinzugefügt, die den MQTT-Messaging-Client für JavaScript verwenden. Er dient zur Anpassung an das Verhalten einiger Web-Browser, die die Anforderung für ein Clientzertifikat wie einen Protokollfehler behandeln.

## Version 7.5.0, Fix Pack 4: Inaktivieren Sie IBM WebSphere MQ Advanced Message Security auf dem Client mithilfe von AMQ\_DISABLE\_CLIENT\_AMS

### V7.5.0.4

Ab Version 7.5.0, Fix Pack 4 verwenden Sie die Umgebungsvariable AMQ\_DISABLE\_CLIENT\_AMS, um IBM WebSphere MQ Advanced Message Security (AMS) in IBM WebSphere MQ classes for JMS -und IBM WebSphere MQ classes for Java -Clients zu inaktivieren, um Fehler zu vermeiden, wenn sie eine Verbindung zu Warteschlangenmanagern herstellen, die in früheren Versionen des Produkts ausgeführt werden. Alternativ und auch ab Version 7.5.0, Fix Pack 4 können Sie die Java -Systemeigenschaft com.ibm.mq.cfg.AMQ\_DISABLE\_CLIENT\_AMS verwenden, um AMS auf dem Client zu inaktivieren.

Weitere Informationen finden Sie unter [IBM WebSphere MQ Advanced Message Security auf dem Client inaktivieren](#).

## Version 7.5.0, Fix Pack 5: Inaktivieren Sie IBM WebSphere MQ Advanced Message Security auf dem Client mithilfe der Datei mqclient.ini.

### V7.5.0.5

Ab Version 7.5.0, Fix Pack 4 können Sie IBM WebSphere MQ Advanced Message Security (AMS) auf dem Client inaktivieren, um Fehler zu verhindern, wenn sie eine Verbindung zu Warteschlangenmanagern herstellen, die in früheren Versionen des Produkts ausgeführt werden, indem Sie die Konfigurationsvariable AMQ\_DISABLE\_CLIENT\_AMS festlegen. Ab Version 7.5.0, Fix Pack 5 können Sie AMS auch mithilfe der AMS-Eigenschaft DisableClient unter der Zeilengruppe **Security** in der Datei mqclient.ini inaktivieren. Weitere Informationen finden Sie unter [IBM WebSphere MQ Advanced Message Security auf dem Client inaktivieren](#).



## Version 7.5.0, Fix Pack 6: Benutzer-ID und Kennwort für verwaltete .NET-Clientanwendungen



### Windows V7.5.0.6

Ab IBM WebSphere MQ Version 7.5.0Fixpack 6 werden die mit der verwalteten .NET-Clientanwendung angegebene Benutzer-ID und das Kennwort in der, IBM WebSphere MQ-Klasse .NET MQChannelDefinition festgelegt, die an den Clientsicherheitsexit übergeben wird. Weitere Informationen finden Sie im Abschnitt [Kanalexits in IBM WebSphere MQ .NET verwenden](#).

## Version 7.5.0, Fix Pack 6: Nicht weiter unterstützte CipherSpecs

Die folgenden CipherSpecs sind ab IBM WebSphere MQ Version 7.5.0Fixpack 6 veraltet:

- DES\_SHA\_EXPORT
-  DES\_SHA\_EXPORT1024
-  FIPS\_WITH\_DES\_CBC\_SHA
- NULL\_MD5
- NULL\_SHA

- TLS\_RSA\_WITH\_DES\_CBC\_SHA
- ECDHE\_ECDSA\_NULL\_SHA256
-  ECDHE\_RSA\_NULL\_SHA256
-  TLS\_RSA\_WITH\_NULL\_NULL
- TLS\_RSA\_WITH\_NULL\_SHA256

Weitere Informationen finden Sie im Abschnitt [Nicht weiter unterstützte CipherSpecs](#).

## Version 7.5.0, Fix Pack 6: Erweiterungen für die Wartungsfreundlichkeit für IBM WebSphere MQ Managed File Transfer



In IBM WebSphere MQ Version 7.5.0Fixpack 6 wurden für IBM WebSphere MQ Managed File Transfer folgende Änderungen vorgenommen:

- Der Standardwert für die Prioritätseigenschaft `commandMessageIn` der Datei `installation.properties` wurde in 8 geändert. Weitere Informationen finden Sie im Abschnitt [Datei installation.properties](#).
- Der Standardwert der Eigenschaft 'logTransferRecovery' in der Datei `agent.properties` wurde in 'true' geändert. Weitere Informationen finden Sie unter [Datei 'agent.properties'](#).
- Bei Verwendung des Parameters **-d** im Befehl **fteShowAgentDetails** muss nicht mehr die Eigenschaft `enableFunctionalFixPack` in der Datei `installation.properties` festgelegt werden. Weitere Informationen finden Sie im Abschnitt [fteShowAgentDetails](#).
- Wenn ein Agent einen nicht behebbaren Fehler feststellt, wird ein First-Failure Data Capture (FDC) generiert.





## Version 7.5.0, Fix Pack 6 Neue IBM WebSphere MQ Managed File Transfer-Agenteneigenschaft failTransferOnFirstFailure

Ab IBM WebSphere MQ Version 7.5.0Fixpack 6 können Sie einen Agenten so konfigurieren, dass eine verwaltete Übertragung fehlschlägt, sobald ein Übertragungselement innerhalb der verwalteten Übertragung fehlschlägt. Hierzu geben Sie in der Datei `agent.properties` die Eigenschaft 'failTransferOnFirstFailure' an. Weitere Informationen finden Sie unter [Datei 'agent.properties'](#).

## Version 7.5.0, Fix Pack 7: Nicht weiter unterstützte CipherSpecs

Die folgenden CipherSpecs sind ab IBM WebSphere MQ Version 7.5.0Fixpack 7veraltet:

- RC2\_MD5\_EXPORT
- RC4\_MD5\_EXPORT
- RC4\_MD5\_US
- RC4\_SHA\_US
-  RC4\_56\_SHA\_EXPORT1024
-  ECDHE\_ECDSA\_RC4\_128\_SHA256
-  ECDHE\_RSA\_RC4\_128\_SHA256
-  TLS\_RSA\_WITH\_RC4\_128\_SHA256

Weitere Informationen finden Sie im Abschnitt [Nicht weiter unterstützte CipherSpecs](#).

## Version 7.5.0, Fix Pack 7: Unterstützung für JDBC-Verbindungen zu einer Oracle 12c-Datenbank

Windows > V 7.5.0.7

Ab IBM WebSphere MQ Version 7.5.0 Fixpack 7 wird eine neue Datei namens `jdbcora12.dll` mit dem Installationsimage des IBM WebSphere MQ Windows -Servers bereitgestellt, um JDBC -Verbindungen zu einer Oracle 12c -Datenbank zu unterstützen (siehe [JTA/JDBC -Koordination unter Windows konfigurieren](#)).

## Version 7.5.0, Fix Pack 7: Entfernung der Einschränkung bei Verwendung von .NET für MQCNO\_CLIENT\_BINDING und MQCNO\_LOCAL\_BINDING

Windows > V 7.5.0.7

Ab IBM WebSphere MQ Version 7.5.0 Fixpack 7 wurde der angepasste IBM WebSphere MQ-Kanal für Microsoft Windows Communication Foundation (WCF) aktualisiert, sodass bei der Ausführung einer reinen Clientinstallation die richtige Konfiguration für die Clientverbindung verwendet wird. Weitere Informationen finden Sie im Abschnitt [Verbindung zu einem Warteschlangenmanager über MQCONNX-Aufruf herstellen](#).

## Version 7.5.0, Fix Pack 8: Einschränkung bei Verwendung von Topic-Aliaswarteschlangen in Verteilerlisten

V 7.5.0.8

Verteilerlisten unterstützen nicht die Verwendung von Aliaswarteschlangen, die auf Themenobjekte verweisen. Wenn ab Version 7.5.0, Fix Pack 8 eine Aliaswarteschlange auf ein Topic-Objekt in einer Verteilerliste zeigt, gibt IBM WebSphere MQ den Fehler `MQRC_ALIAS_BASE_Q_TYPE_ERROR` zurück.

## Version 7.5.0, Fix Pack 8: GSKit-Version aktualisiert

V 7.5.0.8


Die GSKit-Version wurde aktualisiert. Die neue Version von Global Security Kit (GSKit) ändert das Stashdateiformat, das verwendet wird, wenn Sie eine Datei mit der Erweiterung `.sth` verwenden, um das Kennwort für die Schlüsseldatenbank verdeckt zu speichern. Stashdateien, die mit dieser GSKit-Version generiert werden, sind für ältere Versionen von Global Security Kit nicht lesbar. Um sicherzustellen, dass die mit Version 7.5.0, Fix Pack 8 oder höheren Versionen generierten Stashdateien mit Ihren Anwendungen und sonstigen IBM WebSphere MQ-Installationen kompatibel sind, müssen Sie das System mit einer IBM WebSphere MQ-Version aktualisieren, die eine kompatible Version von Global Security Kit enthält. Für IBM WebSphere MQ Version 7.5 ist dies Version 7.5.0, Fix Pack 8.

Wenn Sie Ihre Anwendungen oder sonstigen IBM WebSphere MQ-Installationen nicht aktualisieren können, können Sie ein Stashdateiformat anfordern, das mit einer älteren Version kompatibel ist. Wenn Sie die Befehle `runmqakm` oder `runmqckm` zusammen mit der Option `-stash` oder `-stashpw` verwenden, müssen Sie den Befehlszeilenparameter `-v1stash` einschließen. Sie können für die Generierung einer Stashdatei, die mit einer früheren Version kompatibel ist, nicht die grafische iKeyman-Schnittstelle verwenden.

## Version 7.5.0, Fix Pack 8: Nicht weiter unterstützte CipherSpecs

V 7.5.0.8

Ab Version 7.5.0, Fix Pack 8 werden die folgenden CipherSpecs nicht mehr verwendet:

-  FIPS\_WITH\_3DES\_EDE\_CBC\_SHA
- TRIPLE\_DES\_SHA\_US
- TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA

- ECDHE\_ECDSA\_3DES\_EDE\_CBC\_SHA256
- ECDHE\_RSA\_3DES\_EDE\_CBC\_SHA256

Weitere Informationen finden Sie im Abschnitt [Nicht weiter unterstützte CipherSpecs](#).

## Version 7.5.0, Fix Pack 8: Neue Konstante `JMS_IBM_SUBSCRIPTION_USER_DATA` zur `JmsConstants`-Schnittstelle hinzugefügt

V 7.5.0.8

Ab Version 7.5.0, Fix Pack 8 werden die IBM WebSphere MQ classes for JMS wie folgt aktualisiert: Wenn eine Warteschlange, die einen RFH2-Header mit dem MQPS-Ordner enthält, eine Nachricht verarbeitet, wird der dem Sud-Schlüssel (sofern vorhanden) zugeordnete Wert als Zeichenfolgeeigenschaft (String) zu dem JMS-Nachrichtenobjekt hinzugefügt, das an die IBM WebSphere MQ classes for JMS-Anwendung zurückgegeben wird. Damit eine Anwendung diese Eigenschaft aus der Nachricht abrufen kann, wird eine neue Konstante, `JMS_IBM_SUBSCRIPTION_USER_DATA`, zur `JmsConstants`-Schnittstelle hinzugefügt. Diese neue Eigenschaft kann zusammen mit der Methode `javax.jms.Message.getStringProperty(java.lang.String)` verwendet werden, um die Subskriptionsbenutzerdaten abzurufen. Weitere Informationen finden Sie in den Abschnitten [Abruf von Benutzersubskriptionsdaten](#) und [DEFINE SUB](#).

## Version 7.5.0, Fix Pack 8: Aktualisierungen des JMS-Listeners für Ausnahmebedingungen

V 7.5.0.8

Um das Verhalten für aktuelle JMS-Anwendungen, die einen JMS `MessageListener` und einen JMS `ExceptionListener` konfigurieren, beizubehalten, wurde der Standardwert für die `JMS-ConnectionFactory`-Eigenschaft `ASYNC_EXCEPTIONS` ab IBM WebSphere MQ Version 7.5.0Fixpack 8 für die IBM WebSphere MQ classes for JMS für Version 7.5 in `ASYNC_EXCEPTIONS_CONNECTIONBROKEN` geändert. Deshalb werden standardmäßig nur Ausnahmebedingungen, die Fehlercodes für Verbindungsunterbrechungen entsprechen, an den JMS `ExceptionListener` einer Anwendung übergeben. Ausnahmebedingungen aufgrund von Verbindungsunterbrechungen werden an den Listener für Ausnahmebedingungen übergeben, wenn Nachrichten synchron oder asynchron verarbeitet werden.

Die IBM WebSphere MQ classes for JMS für Version 7.5 wurden zudem dahingehend aktualisiert, dass JMS-Ausnahmebedingungen, die sich auf Fehler aufgrund von Verbindungsunterbrechungen beziehen und während der Nachrichtenübermittlung an asynchrone Nachrichtenkonsumenten auftreten, an einen registrierten `ExceptionListener` übergeben werden, wenn die Eigenschaft `ASYNC_EXCEPTIONS` der von der Anwendung verwendeten `JMS ConnectionFactory` auf den Wert `ASYNC_EXCEPTIONS_ALL` gesetzt ist.

Weitere Informationen finden Sie in den Abschnitten [JMS: Änderungen des Listeners für Ausnahmebedingungen in Version 7.5](#) und [Ausnahmebedingungen in IBM WebSphere MQ-Klassen for JMS](#).

## Version 7.5.0, Fix Pack 8: Unterstützung für Zulassungslisten für Klassennamen in IBM WebSphere MQ classes for JMS `ObjectMessage`

V 7.5.0.8

Ab IBM WebSphere MQ Version 7.5.0Fixpack 8 unterstützt IBM WebSphere MQ classes for JMS die Zulassungsliste von Klassen in der Implementierung der JMS-Schnittstelle `ObjectMessage`. In der Zulassungsliste wird definiert, welche Java-Klassen mit `ObjectMessage.setObject()` serialisiert und mit `ObjectMessage.getObject()` deserialisiert werden können.

Weitere Informationen finden Sie unter [Class name allowlisting in JMS ObjectMessage](#) und [Running IBM WebSphere MQ Classes for JMS-Anwendungen unter Java Security Manager](#).

## Version 7.5.0, Fix Pack 8: Neue IBM WebSphere MQ Managed File Transfer-Agenteneigenschaft `additionalWildcardSandboxChecking`

V 7.5.0.8

Ab IBM WebSphere MQ Version 7.5.0 Fixpack 8 gilt Folgendes: Wenn ein Agent mit einer Benutzer- oder Agentensandbox konfiguriert wurde, um die Positionen zu beschränken, an die der Agent Dateien übertragen und von der er Dateien empfangen kann, können Sie angeben, dass für den betreffenden Agenten zusätzliche Prüfungen bei Platzhalterübertragungen vorgenommen werden. Hierfür müssen Sie die Eigenschaft `'additionalWildcardSandboxChecking'` auf `'true'` setzen. Weitere Informationen finden Sie unter [Zusätzliche Prüfungen bei Platzhalterübertragungen](#) und [Datei 'agent.properties'](#).

## Version 7.5.0, Fix Pack 8: Änderung des Verhaltens des Befehls IBM WebSphere MQ Managed File Transfer `fteCleanAgent`

V 7.5.0.8

Ab IBM WebSphere MQ Version 7.5.0 Fixpack 8 wurde der Befehl `fteCleanAgent` aktualisiert, sodass Sie angeben müssen, welcher IBM WebSphere MQ Managed File Transfer-Status gelöscht werden soll. Hierfür müssen Sie die entsprechenden Parameter an den Befehl übergeben und einen Agentennamen bereitstellen. Durch diese Verhaltensänderung ist sichergestellt, dass `fteCleanAgent` standardmäßig nicht alle laufenden und anstehenden Übertragungen, Ressourcenüberwachungsdefinitionen und geplanten Übertragungsdefinitionen für den angegebenen Agenten löscht.

Falls erforderlich, können Sie zum vorherigen Verhalten von `fteCleanAgent` zurückkehren, indem Sie die neue Eigenschaft `failCleanAgentWithNoArguments` in der Datei `command.properties` auf `'false'` setzen.

Weitere Informationen finden Sie unter [fteCleanAgent \(bereinigt einen Managed File Transfer Agent\)](#) und [Datei 'command.properties'](#).

## Version 7.5.0, Fix Pack 9: Neue Umgebungsvariable `AMQ_SSL_LDAP_SERVER_VERSION`

V 7.5.0.9

Ab Version 7.5.0, Fix Pack 9 können Sie die Umgebungsvariable `AMQ_SSL_LDAP_SERVER_VERSION` setzen, um sicherzustellen, dass IBM WebSphere MQ die erforderliche Lightweight Directory Access Protocol-(LDAP-)Version, entweder LDAP v2 oder LDAP v3, an GSKit übergibt, wenn ein Zertifikat anhand einer höheren Version des LDAP Certificate Revocation List-(CRL-)Servers geprüft wird. Weitere Informationen finden Sie im Abschnitt [Umgebungsvariablen](#).

## Version 7.5.0, Fix Pack 9: Beispielkonfigurationsdatei für IBM WebSphere MQ classes for JMS

V 7.5.0.9

Ab Version 7.5.0, Fix Pack 9 wird eine Beispielkonfigurationsdatei namens `mqjava.config` im Unterverzeichnis `bin` des Installationsverzeichnis von IBM WebSphere MQ classes for JMS bereitgestellt. Weitere Informationen finden Sie im Abschnitt [Konfigurationsdatei für IBM MQ-Klassen für Java](#).

## Version 7.5.0, Fix Pack 9: Änderungen am Befehl `dmpmqcfg`

V 7.5.0.9

Ab Version 7.5.0, Fix Pack 9 wird Befehlszeilenoption `-w` zum Befehl `dmpmqcfg` hinzugefügt. Mit dieser Option können Sie die Zeit (in Sekunden) verlängern, die `dmpmqcfg` auf eine Antwort vom Befehlsserver wartet.

Weitere Informationen finden Sie im Abschnitt [dmpmqcfg](#).



## Version 7.5.0, Fix Pack 9: Änderungen am Befehl `strmqm`

### V 7.5.0.9

Ab Version 7.5.0, Fix Pack 9 prüft der Befehl `strmqm` die Syntax der Zeilengruppen CHANNELS und SSL in der Datei `qm.ini`, bevor er den Warteschlangenmanager vollständig startet. Dies macht es einfacher, Fehler zu erkennen und schnell zu korrigieren, falls `strmqm` feststellt, dass die Datei `qm.ini` Fehler enthält. Weitere Informationen finden Sie im Abschnitt [strmqm](#).

## Funktionale Erweiterungen des Dienstprogramms 'runmqras'

### V 7.5.0.9

Ab Version 7.5.0, Fix Pack 9 sind die folgenden Erweiterungen für das Dienstprogramm `runmqras` verfügbar:

- **AIX** **Solaris** **Linux** Die Informationen zur Umgebungsvariable werden standardmäßig abgerufen.
- **distributed** Die Verzeichnisliste mit dem Warteschlangenmanagerdaten wird standardmäßig abgerufen.
- Die folgenden beiden Abschnitte werden dem Befehl `runmqras` hinzugefügt:
  - **Linux** **UNIX** Der Abschnitt `leak`, um Informationen zur Nutzung von IBM WebSphere MQ-Prozessressourcen zu erfassen.
  - **distributed** Der Abschnitt `mft`, um die durch den Befehl `fteRas` abgerufenen Daten zu erfassen.

Weitere Informationen hierzu finden Sie unter [runmqras](#).

## Version 7.5.0, Fix Pack 9: Ressourcenadapteraktualisierungen für Eigenschaft `targetClientMatching` in einer Aktivierungsspezifikation

### V 7.5.0.9

Der IBM WebSphere MQ-JCA-Ressourcenadapter (MQ-RA) wurde so aktualisiert, dass die Eigenschaft `targetClientMatching` für eine Aktivierungsspezifikation konfiguriert werden kann. Sie können die Eigenschaft so konfigurieren, dass ein MQRFH2-Header in Antwortnachrichten eingeschlossen wird, wenn Anforderungsnachrichten keinen MQRFH2-Header enthalten. Dies bedeutet, dass alle Nachrichteneigenschaften, die eine Anwendung in einer Antwortnachricht definiert, beim Senden der Nachricht eingeschlossen werden. Weitere Informationen finden Sie im Abschnitt [Eigenschaft targetClientMatching für eine Aktivierungsspezifikation konfigurieren](#).

## Version 7.5.0, Fix Pack 9: Neuer Parameter `-caseno` für den Befehl `runmqras`

### V 7.5.0.9

Der Parameter `-caseno` entspricht funktional dem Parameter `-pmrno`, bietet jedoch zusätzlich die Möglichkeit, eine gültige Salesforce-Fallnummer anzugeben. `-caseno` und `-pmrno` sind optionale Parameter, die jedoch nicht zusammen angegeben werden dürfen. Weitere Informationen finden Sie im Abschnitt [runmqras \(IBM MQ-Diagnoseinformationen erfassen\)](#).

## Version 7.5.0, Fix Pack 9: Änderung bei der Behandlung von Fehlern in Verbindung mit beschädigten Zeichendaten innerhalb von internen MQ-Befehlsnachrichten, die von der Warteschlangenmanager-Clustering-Komponente verwendet werden

### V 7.5.0.9

Ab Version 7.5.0, Fix Pack 9 wird die Art und Weise geändert, in der der Warteschlangenmanager Fehler behandelt, die sich auf beschädigte Zeichendaten in internen MQ-Befehlsnachrichten beziehen, die von der Clustering-Komponente des Warteschlangenmanagers verwendet werden. Felder, die möglicherweise

nicht konvertierbare Zeichen enthalten, werden ausgeblendet und die Konvertierung wird wiederholt. Ist die Konvertierung erfolgreich, wird die Aktualisierung mit einer leeren Beschreibung gespeichert und die Fehlernachrichten AMQ6174 und AMQ9880 werden ausgegeben, um den Administrator zu informieren. Wenn die Nachricht weiterhin nicht konvertierbar ist, wird sie in die Warteschlange für nicht zustellbare Nachrichten verschoben und es wird eine Fehlernachricht ausgegeben.

## **Version 7.5.0, Fix Pack 9: Hinzufügen von Microsoft Visual Studio 2013 C/C++-Laufzeiten unter Windows**

**V 7.5.0.9** **Windows**

Ab Version 7.5.0, Fix Pack 9 erfordern einige Elemente von IBM WebSphere MQ die Microsoft Visual Studio 2013 C/C++-Laufzeiten (VS2013). Diese Laufzeiten werden mit jeder neuen Installation von IBM WebSphere MQ Version 7.5.0 Fixpack 9 installiert.

Falls die VS2013-Laufzeiten nicht auf dem System vorhanden sind, werden sie installiert, wenn Sie einen neuen Client installieren, einen Client auf die Wartungsstufe 7.5.0.9 aktualisieren oder das Fixpack 7.5.0.9 anwenden. In den meisten Fällen erfordert dieser Prozess keine Benutzeraktion. Wenn jedoch bei der Anwendung des Server-Fixpacks ein Installationsfehler auftreten sollte, gibt es eine Umgebungsvariable, die Sie möglicherweise angeben müssen, damit Sie die Laufzeiten manuell installieren können. Weitere Informationen finden Sie im Abschnitt [Installation von Microsoft Visual Studio 2013 C/C++-Laufzeiten in einem Server-Fixpack](#).

## **Version 7.5.0, Fix Pack 9: Abgleichsprüfung von Übertragungs-ID und Wert des Attributs groupId für Übertragungen aus Nachrichten in Dateien wiederhergestellt**

**V 7.5.0.9**

Ab Version 7.5.0, Fix Pack 9 wird IBM WebSphere MQ Managed File Transfer aktualisiert, um die zuvor durch APAR IT18213 bei Version 7.5.0, Fix Pack 6 entfernte Abgleichsprüfung der Übertragungs-ID und des Werts des Attributs 'groupId' innerhalb der XML-Nutzdaten der Übertragungsanforderung wiederherzustellen. 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. Siehe auch [Daten aus Nachrichten in Dateien übertragen](#).

## **Version 7.5.0, Fix Pack 9 sowie vorläufiger Fix für APAR IT26482: Änderung der Berechtigungen, die erforderlich sind, damit IBM WebSphere MQ classes for JMS die BackoutThreshold und BackoutRequeueQName einer Clusterwarteschlange abfragen kann**

**V 7.5.0.9**

Für Version 7.5.0, Fix Pack 9 und einen vorläufigen Fix für APAR IT26482 wurden die IBM WebSphere MQ classes for JMS aktualisiert, sodass nur Abfragezugriff erforderlich ist, um die **BackoutThreshold** und **BackoutRequeueQName** einer Clusterwarteschlange abzufragen. Für alle anderen Versionen sind auch Browse- und Abrufzugriff erforderlich. Weitere Informationen finden Sie im Abschnitt [Behandlung von nicht verarbeitbaren Nachrichten in IBM MQ-Klassen für JMS](#).

### **Zugehörige Konzepte**

„Neuerungen in IBM WebSphere MQ Version 7.5“ auf Seite 20

In diesem Abschnitt werden die wichtigsten neuen Funktionen in IBM WebSphere MQ Version 7.5 beschrieben.

„Neuerungen in IBM WebSphere MQ Version 7.5“ auf Seite 25

### **Zugehörige Informationen**

[Systemvoraussetzungen für IBM WebSphere MQ](#)

[Webseite der Produkt-Readmes für IBM MQ, WebSphere MQ und MQSeries](#)

[Empfohlene Fixes für WebSphere MQ](#)

## Neuerungen und Änderungen in älteren Versionen

---

Hier finden Sie Links zu Informationen über neue Features und Änderungen an Funktionen und Ressourcen, die in Produktversionen vor IBM WebSphere MQ Version 7.5 vorgenommen wurden. Dies betrifft auch Stabilisierungen, Einstellungen der Unterstützung und Entfernungen.

Informationen zu Neuerungen und Änderungen in einer älteren Version des Produkts finden Sie im betreffenden Abschnitt in der Produktdokumentation für die jeweilige Version.

### IBM WebSphere MQ Version 7.1

- [Neuerungen in IBM WebSphere MQ Version 7.1](#)
- [Änderungen in IBM WebSphere MQ Version 7.1](#)
- [Änderungen in IBM WebSphere MQ Version 7.1-Fixpacks](#)

### IBM WebSphere MQ Version 7.0.1 und früher

Informationen zu älteren Versionen von Produkten, bei denen die Dokumentation außerhalb von IBM Documentation bereitgestellt wird, finden Sie in der [Documentation für ältere Versionen von WebSphere MQ](#).

#### Zugehörige Konzepte

„[Neuerungen in IBM WebSphere MQ Version 7.5](#)“ auf Seite 20

In diesem Abschnitt werden die wichtigsten neuen Funktionen in IBM WebSphere MQ Version 7.5 beschrieben.

„[Neuerungen in IBM WebSphere MQ Version 7.5](#)“ auf Seite 25

„[Änderungen in IBM WebSphere MQ Version 7.5 -Fixpacks](#)“ auf Seite 26

In diesem Abschnitt werden Änderungen der Funktionen und Ressourcen in Fixpacks von Version 7.5 beschrieben.

[Einführung in die Migration von WebSphere MQ](#)

## Zuordnung älterer IBM WebSphere MQ-Veröffentlichungen zur aktuellen Informationsstruktur

---

Die Informationen in IBM Documentation sind entsprechend einer Gruppe generischer Aktivitäten strukturiert. Hierzu gehören beispielsweise die Installation, Entwicklung, Verwaltung, Konfiguration, Sicherung. Frühere Veröffentlichungen (die "MQ-Bücher") waren etwas anders strukturiert. Dieser Abschnitt enthält eine Zuordnung zwischen diesen früheren IBM WebSphere MQ -Veröffentlichungen und der aktuellen Informationsstruktur.

### Einstieg für AIX

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Einstieg für AIX" zur neuen Produktdokumentationsstruktur:

- [AIX: Migration auf höhere Version planen](#)
- [Anforderungen überprüfen](#)
- [Das System vorbereiten](#)
- [IBM WebSphere MQ -Server unter AIX installieren](#)
- [Serverinstallation überprüfen](#)
- [IBM WebSphere MQ-Client auf AIX-Systemen installieren](#)
- [Clientinstallation überprüfen](#)

- [AIX: Wartungsstufenupgrades auf IBM WebSphere MQ Version 7.5 anwenden](#)
- [IBM WebSphere MQ unter AIXdeinstallieren](#)

## Einstieg für HP-UX

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Einstieg für HP-UX" zur neuen Produktdokumentationsstruktur:

- [HP-UX: Migration auf höhere Version planen](#)
- [Anforderungen überprüfen](#)
- [Das System vorbereiten](#)
- [IBM WebSphere MQ -Server unter HP-UXinstallieren](#)
- [Serverinstallation überprüfen](#)
- [IBM WebSphere MQ-Client auf HP-UX-Systemen installieren](#)
- [Clientinstallation überprüfen](#)
- [HP-UX: Wartungsstufenaktualisierungen auf IBM WebSphere MQ Version 7.5](#)
- [IBM WebSphere MQ unter HP-UXdeinstallieren](#)

## Einstieg für Linux

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Einstieg für Linux " zur aktuellen Produktdokumentationsstruktur:

- [Linux: Migration von IBM WebSphere MQ Version 7.0.1 in IBM WebSphere MQ Version 7.5planen](#)
- [Anforderungen überprüfen](#)
- [Das System vorbereiten](#)
- [IBM WebSphere MQ -Server unter Linuxinstallieren](#)
- [Serverinstallation überprüfen](#)
- [WebSphere MQ -Client unter Linuxinstallieren](#)
- [Clientinstallation überprüfen](#)
- [Linux: Aktualisierungen der Wartungsstufe auf IBM WebSphere MQ Version 7.5anwenden](#)
- [IBM WebSphere MQ unter Linuxdeinstallieren](#)

## Einstieg für Solaris

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Einstieg für Solaris" zur neuen Produktdokumentationsstruktur:

- [Solaris: Migration auf höhere Version planen](#)
- [Anforderungen überprüfen](#)
- [Das System vorbereiten](#)
- [IBM WebSphere MQ-Server unter Solaris installieren](#)
- [Serverinstallation überprüfen](#)
- [IBM WebSphere MQ -Client auf Solaris](#)
- [Clientinstallation überprüfen](#)
- [Solaris: Wartungsstufenaktualisierungen auf IBM WebSphere MQ Version 7.5](#)
- [IBM WebSphere MQ unter Solaris deinstallieren](#)

## Einstieg für Windows

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Einstieg für Windows" zur neuen Produktdokumentationsstruktur:

- [Windows: Migration auf höhere Version planen](#)
- [Anforderungen überprüfen](#)
- [Das System vorbereiten](#)
- [IBM WebSphere MQ -Server unter Windows installieren](#)
- [Serverinstallation überprüfen](#)
- [IBM WebSphere MQ-Client auf Windows-Systemen installieren](#)
- [Clientinstallation überprüfen](#)
- [Windows: Wartungsstufenupgrades auf IBM WebSphere MQ Version 7.5 anwenden](#)
- [IBM WebSphere MQ unter Solaris deinstallieren](#)

## Application Programming Guide

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Application Programming Guide" zur neuen Produktdokumentationsstruktur:

- [Konzepte für die Anwendungsentwicklung](#)
- [IBM WebSphere MQ-Anwendungen entwerfen](#)
- [Anwendung zur Warteschlangensteuerung schreiben](#)
- [WebSphere MQ-Musterprogramme](#)
- [Beispiele für Programmiersprache C](#)
- [COBOL-Beispiele](#)
- [System/390-Assemblerbeispiele](#)
- [WebSphere MQ-Datendefinitionsdateien](#)
- [Codierungsstandards auf 64-Bit-Plattformen](#)

## Application Programming Reference

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Application Programming Reference" zur neuen Produktdokumentationsstruktur:

- [Im MQI verwendete Datentypen](#)
- [Funktionsaufrufe](#)
- [Attribute von Objekten](#)
- [Rückgabecodes](#)
- [Regeln zur Überprüfung von MQI-Optionen](#)
- [Berichtsoptionen und Nachrichtenattribute](#)
- [Berichtsoptionen und Nachrichtenattribute](#)
- [Datenkonvertierung](#)
- [Als MQRFH2-Elemente angegebene Eigenschaften](#)
- [Codepage-Konvertierung](#)

## Kunden

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Clients" zur neuen Produktdokumentationsstruktur:

- [„Überblick über IBM WebSphere MQ MQI-Clients“ auf Seite 140](#)

- [„Plattformunterstützung für IBM WebSphere MQ-Clients“ auf Seite 143](#)
- [IBM WebSphere MQ-Client installieren](#)
- [Verbindungen zwischen Server und Client konfigurieren](#)
- [Erweiterten transaktionsorientierten Client konfigurieren](#)
- [Clientinstallation überprüfen](#)
- [WebSphere MQ-Clientsicherheit einrichten](#)
- [„Kanäle“ auf Seite 126](#)
- [MQI-Kanäle definieren](#)
- [Serververbindungs- und Clientverbindungsdefinitionen auf verschiedenen Plattformen erstellen](#)
- [Serververbindungs- und Clientverbindungsdefinitionen auf dem Server erstellen](#)
- [Kanalexitprogramme für MQI-Kanäle](#)
- [Verbindung zwischen einem Client und einer Gruppe mit gemeinsamer Warteschlange herstellen](#)
- [Client mithilfe einer Konfigurationsdatei konfigurieren](#)
- [WebSphere MQ-Umgebungsvariablen verwenden](#)
- [Schnittstelle für Nachrichtenwarteschlangen \(MQI\) in einer Clientanwendung verwenden](#)
- [Anwendungen für WebSphere MQ-Clients erstellen](#)
- [Anwendungen in der WebSphere MQ-Clientumgebung ausführen](#)
- [CICS- und Tuxedo-Anwendungen vorbereiten und ausführen](#)
- [Microsoft Transaction Server-Anwendungen vorbereiten und ausführen](#)
- [WebSphere MQ-JMS-Anwendungen vorbereiten und ausführen](#)
- [Probleme mit IBM WebSphere MQ-MQI-Clients beheben](#)
- [Mithilfe eines PreConnect-Exits in einem Repository nach Verbindungsdefinitionen suchen](#)

## Konstanten

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Konstanten" zur neuen Produktdokumentationsstruktur:

- [WebSphere MQ-Dateien vom Typ COPY, Header, Include und Modul](#)
- [WebSphere MQ-Dateien vom Typ COPY, Header, Include und Modul](#)
- [Konstanten](#)

## Intercommunication

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Intercommunication" zur neuen Produktdokumentationsstruktur:

### Einführung

- [„Konzepte der übergreifenden Kommunikation“ auf Seite 53](#)
- [Anwendungen mithilfe der verteilten Steuerung von Warteschlangen verbinden](#)
- [Netze und Netzplanung](#)
- [WebSphere MQ-Verfahren für verteilte Nachrichtenübermittlung](#)
- [Einführung in verteiltes Warteschlangenmanagement](#)
- [Kanalattribute](#)
- [Beispielkonfigurationsdaten](#)

 **Verteiltes Warteschlangenmanagement in WebSphere MQ für Windows- und UNIX-Plattformen**

- [Kanäle auf den Plattformen Windows, UNIX and Linux überwachen und steuern](#)

- [Übertragungswarteschlange erstellen](#)
- [Kanäle auslösen](#)
- [Kanalprogramme](#)
- [Sicherheit bei fernem Messaging](#)
- [Sonstige Aspekte beim verteilten Warteschlangenmanagement](#)
- [Kommunikation für Windows einrichten](#)
- [Beispielkonfiguration - IBM WebSphere MQ for Windows](#)
- [Beispielkonfiguration - IBM WebSphere MQ for AIX](#)
- [Beispielkonfiguration - IBM WebSphere MQ for HP-UX](#)
- [Beispielkonfiguration - IBM WebSphere MQ for Solaris](#)
- [Beispielkonfiguration- IBM WebSphere MQ für Linux](#)
- [Beispiel für Nachrichtenkanalplanung für verteilte Plattformen](#)

#### **Weitere Hinweise zur übergreifenden Kommunikation (Intercommunication)**

- [Kanalexitprogramme für Nachrichtenkanäle](#)
- [Kanalexitaufrufe und Datenstrukturen](#)
- [Auflösung des Warteschlangennamens](#)

## **Nachrichten und Codes**

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Messages and Codes" zur neuen Produktdokumentationsstruktur:

- [Diagnosenachrichten: AMQ4000-9999](#)
- [API-Beendigungs- und Ursachencodes](#)
- [PCF-Ursachencodes](#)
- [Secure Sockets Layer \(SSL\) und Transport Layer Security \(TLS\) - Rückkehrcodes](#)
- [Ausnahmebedingungen für den benutzerdefinierten WCF-Kanal](#)

## **Migrationsinformationen**

Die Struktur der Migrationsabschnitte für Version 7.5 hat sich seit Version 7.1 nicht geändert.

Für die Migration auf Version 7.5 wurden Themen hinzugefügt, während für die Migration auf Version 7.1 Themen entfernt wurden. Informationen zur Migration auf Version 7.1 und frühere Releases finden Sie unter [Wo finden Sie ein Thema zu einem bestimmten Migrationspfad?](#)

## **Überwachung**

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Monitoring" zur neuen Produktdokumentationsstruktur:

- [Ereignisüberwachung](#)
- [Nachrichtenüberwachung](#)
- [Abrechnungs- und Statistiknachrichten](#)
- [Echtzeitüberwachung](#)
- [Strukturdatentypen](#)
- [Objektattribute für Ereignisdaten](#)

## Programmable Command Formats and Administration Interface

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Programmable Command Formats and Administration Interface" zur neuen Produktdokumentationsstruktur:

- [Einführung in Programmable Command Formats](#)
- [Einführung in WebSphere MQ Administration Interface \(MQAI\)](#)

## Publish/Subscribe-Benutzerhandbuch

Dieser Abschnitt enthält eine Zuordnung des alten Publish/Subscribe-Benutzerhandbuchs zur neuen Produktdokumentationsstruktur:

- [Einführung in das Publish/Subscribe-Messaging von WebSphere MQ](#)
- [Verteiltes Publish/Subscribe](#)
- [Publish/Subscribe-Anwendungen schreiben](#)
- [Publish/Subscribe-Sicherheit](#)
- [Publish/Subscribe-Migration von Version 6.0](#)
- [Migration des Publish/Subscribe-Brokers in WebSphere Event Broker und WebSphere Message Broker](#)

## Cluster aus Warteschlangenmanagern

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Queue Manager Clusters" zur neuen Produktdokumentationsstruktur:

- [„Wie funktionieren Cluster?“ auf Seite 68](#)
- [Warteschlangenmanager-Cluster konfigurieren](#)
- [IBM WebSphere MQ-Cluster verwalten](#)
- [Nachrichten an und von Clustern weiterleiten](#)
- [Cluster für das Workload-Management verwenden](#)
- [Cluster schützen](#)
- [Mit MQI und Clustern arbeiten](#)
- [WebSphere MQ-Clusterbefehle](#)
- [Probleme mit Warteschlangenmanagerclustern beheben](#)

## Referenz zu Scriptbefehlen (MQSC)

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Scriptbefehlsreferenz (MQSC)" zur neuen Produktdokumentationsstruktur:

- [Scriptbefehle \(MQSC\)](#)
- [Generische Werte und Zeichen mit besonderer Bedeutung](#)
- [Befehlsscripts erstellen](#)
- [„Regeln für die Benennung von IBM WebSphere MQ-Objekten“ auf Seite 132](#)
- [Syntaxdiagramme](#)
- [WebSphere MQ-Scriptbefehle](#)

## Sicherheit

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Sicherheit" zur neuen Produktdokumentationsstruktur:

- [Sicherheit](#)



## System Administration Guide

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "System Administration Guide" zur neuen Produktdokumentationsstruktur:

- [„IBM WebSphere MQ - Technische Übersicht“ auf Seite 44](#)
- [Verwaltung IBM WebSphere MQ](#)
- [Lokale WebSphere MQ-Objekte verwalten](#)
- [Verwaltung mithilfe von IBM WebSphere MQ Explorer](#)
- [Verwendung der WebSphere MQ Taskleistanwendung \(nur Windows\)](#)
- [WebSphere MQ-Steuerbefehle](#)

### Konfiguration und Verwaltung

- [Konfigurationsdaten von IBM WebSphere MQ und von Warteschlangenmanagern ändern](#)
- [Dateisystemunterstützung planen](#)
- [Sicherheit auf Systemen mit Windows, UNIX and Linux einrichten](#)
- [Transaktionsunterstützung](#)
- [Verarbeitung nicht zugestellter Nachrichten durch die WebSphere MQ-Steerroutine der Warteschlange für nicht zustellbare Nachrichten](#)
- [Verfügbarkeit, Wiederherstellung und Neustart](#)
- [Fehlerbehebung und Support](#)
- [IPC-Ressourcen von WebSphere MQ und UNIX System V](#)
- [WebSphere MQ- und UNIX-Prozesspriorität](#)
- [Benutzerexits, API-Exits und installierbare WebSphere MQ-Services](#)

## .NET verwenden

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Using .NET" zur neuen Produktdokumentationsstruktur:

- [Verwendung von .NET](#)
- [.NET-Programme von WebSphere MQ schreiben und implementieren](#)
- [.NET-Klassen und -Schnittstellen von WebSphere MQ](#)
- [Benutzerdefinierter IBM WebSphere MQ-Kanal für Microsoft Windows Communication Foundation \(WCF\)](#)

## C++ verwenden

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Using C++" zur neuen Produktdokumentationsstruktur:

- [Verwendung von C++](#)
- [WebSphere MQ C++-Klassen](#)

## Using Java

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Using Java" zur neuen Produktdokumentationsstruktur:

- [Sollte ich WebSphere MQ Classes for Java oder WebSphere MQ Classes for JMS verwenden?](#)
- [WebSphere MQ Classes for Java verwenden](#)
- [WebSphere MQ-Klassen für JMS](#)
- [WebSphere MQ Classes for JMS verwenden](#)

- [WebSphere MQ-Klassen für Java](#)

## Web-Services

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Web Services" zur neuen Produktdokumentationsstruktur:

- [WebSphere MQ Transport for SOAP](#)
- [WebSphere MQ Bridge for HTTP](#)

## Component Object Model-Schnittstelle verwenden

Dieser Abschnitt enthält eine Zuordnung des alten Handbuchs "Verwendung der Component Object Model-Schnittstelle" zur neuen Produktdokumentationsstruktur:

- [Verwendung der Component Object Model-Schnittstelle \(WebSphere MQ Automation Classes for ActiveX\)](#)

## IBM WebSphere MQ Version 7.5 - IBM i und z/OS

---

IBM WebSphere MQ Version 7.5 ist für IBM i und z/OS nicht verfügbar.

Diese Plattformen sind in höheren Versionen des Produkts verfügbar.

Informationen zu den neuesten Versionen von IBM MQ for IBM i und z/OS finden Sie auf der [Website von IBM MQ](#).

## Bemerkungen

---

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf 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 dieser Dokumentation 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.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 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. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

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.

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 unabhängig voneinander 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  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
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 diesem Dokument 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. Leistungen in anderen Betriebsumgebungen können von diesen Daten erheblich abweichen. Möglicherweise wurden Messungen noch mit Systemen im Entwicklungsstadium vorgenommen, deren Ergebnisse sich nicht in jedem Fall auf öffentlich verfügbare Systeme übertragen lassen. Gelegentlich werden Messergebnisse auch durch Extrapolation ermittelt. Die tatsächlichen Ergebnisse können von diesen Ergebnissen 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 zu künftigen Plänen und Absichten von IBM sind unverbindlich und können sich jederzeit und ohne Ankündigung ändern.

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 Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, IBM Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle des Betriebssystems kompatibel sind, für das diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielprogramme entstehen.

Sämtliche Kopien, Ausschnitte und Revisionen dieser Beispielprogramme müssen folgenden Copyrightvermerk enthalten: © (Name Ihres Unternehmens) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. Beispielprogramme. © Copyright IBM Corp. 1993, 2024. Alle Rechte vorbehalten.

In der Softcopy-Version dieser Veröffentlichung werden Fotografien und farbige Abbildungen unter Umständen nicht korrekt dargestellt.

## Marken

IBM, das IBM -Logo, ibm.com, Passport Advantage, WebSphere, MQSeries und z/OS sind Marken oder eingetragene Marken der International Business Machines Corp. in vielen Ländern weltweit registriert. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter <https://www.ibm.com/legal/copytrade.shtml>.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sind Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Die in diesem Produkt enthaltene Oracle Outside In Technology wird mit einer eingeschränkten Lizenz bereitgestellt und kann nur in Verbindung mit dieser Anwendung verwendet werden.

Intel ist eine Marke oder eingetragene Marke der Intel Corporation oder deren Tochtergesellschaften in den USA und/oder anderen Ländern.

Die eingetragene Marke Linux wird gemäß einer Unterlizenz der Linux Foundation, dem exklusiven Lizenznehmer von Linus Torvalds, dem weltweit Inhaber der Marke, verwendet.

## IBM WebSphere MQ - Technische Übersicht

---

Mit IBM WebSphere MQ können Sie Verbindungen für Ihre Anwendungen herstellen und die Verteilung von Informationen in Ihrem Unternehmen verwalten.

IBM WebSphere MQ ermöglicht Programmen, über ein Netz unähnlicher Komponenten (Prozessoren, Betriebssysteme, Subsysteme und Übertragungsprotokolle) mithilfe einer konsistenten Anwendungsprogrammierschnittstelle miteinander zu kommunizieren. Anwendungen, die mit dieser Schnittstelle entworfen und geschrieben wurden, werden als Message-Queuing-Anwendungen bezeichnet (Anwendungen zur Steuerung von Nachrichtenwarteschlangen).

In den folgenden Abschnitten erhalten Sie Informationen zur Steuerung von Nachrichtenwarteschlangen und anderen Features von IBM WebSphere MQ.

### Zugehörige Konzepte

[„Einführung in IBM WebSphere MQ“ auf Seite 6](#)

Mithilfe von IBM WebSphere MQ können Sie es Anwendungen ermöglichen, zu verschiedenen Zeiten und in vielen verschiedenen IT-Umgebungen zu kommunizieren.

### Zugehörige Tasks

[WebSphere MQ-Architektur entwerfen](#)

[WebSphere MQ Managed File Transfer](#)

### Zugehörige Verweise

„Hauptmerkmale und Vorteile des Message-Queuing“ auf Seite 47

In diesem Abschnitt werden einige Merkmale und Vorteile des Message-Queuing hervorgehoben. Sie finden hier beispielsweise eine Beschreibung der Message-Queuing-Merkmale in Bezug auf Sicherheit und Datenintegrität.

## Einführung in die Nachrichtenwarteschlangensteuerung

Mit den WebSphere MQ-Produkten können Programme über ein Netz ungleicher Komponenten (Prozessoren, Betriebssysteme, Subsysteme und Kommunikationsprotokolle) unter Verwendung einer konsistenten Anwendungsprogrammierschnittstelle miteinander kommunizieren.

Anwendungen, die über diese Schnittstelle entwickelt und geschrieben wurden, werden als *Message-Queuing*-Anwendungen bezeichnet, da sie *Messaging* und *Queuing* (Warteschlangensteuerung) verwenden:

<b>Messaging</b>	Programme kommunizieren miteinander, indem sie sich gegenseitig Daten in Nachrichten zusenden, anstatt sich direkt anzurufen.
<b>Queuing</b>	Nachrichten werden in Warteschlangen gespeichert. Programme können somit unabhängig voneinander mit unterschiedlicher Verarbeitungsgeschwindigkeit und zu unterschiedlichen Zeiten an verschiedenen Standorten ausgeführt werden, ohne dass eine logische Verbindung zwischen ihnen bestehen muss.

Message-Queuing ist in der Datenverarbeitung seit vielen Jahren im Einsatz. Heutzutage wird es im Allgemeinen in E-Mails verwendet. Ohne Queuing (Warteschlangensteuerung) muss beim Senden einer elektronischen Nachricht über weite Entfernungen jeder einzelne Knoten auf der Route zur Weiterleitung von Nachrichten verfügbar sein. Außerdem müssen die Empfänger angemeldet sein und wissen, dass Sie ihnen eine Nachricht senden möchten. In einem Warteschlangensystem werden die Nachrichten an Zwischenknoten gespeichert, bis das System zu ihrer Weiterleitung bereit ist. Am Zielort werden sie in einer elektronischen Mailbox gespeichert, bis der Empfänger bereit ist, sie zu lesen.

Trotz alledem werden viele komplexe Geschäftstransaktionen heutzutage ohne Queuing verarbeitet. In einem großen Netz kann das System möglicherweise viele Tausende Verbindungen sofort einsatzfähig halten. Tritt in einem Teil des Systems ein Problem auf, können viele Systemteile nicht mehr eingesetzt werden.

Sie können sich Message-Queuing als E-Mail für Programme vorstellen. In einer Message-Queuing-Umgebung führt jedes Programm einer Anwendungssuite eine klar strukturierte eigenständige Funktion als Antwort auf eine bestimmte Anforderung aus. Für die Kommunikation mit einem anderen Programm muss ein Programm eine Nachricht in eine vordefinierte Warteschlange einreihen. Diese Nachricht wird von dem anderen Programm aus der Warteschlange abgerufen und die Anforderungen und Angaben aus der Nachricht werden verarbeitet. Message-Queuing stellt somit eine Art der Kommunikation zwischen Programmen dar.

Queuing bezeichnet den Mechanismus, mit dem Nachrichten zurückbehalten werden, bis sie von einer Anwendung verarbeitet werden können. Queuing bietet folgende Möglichkeiten:

- Kommunikation zwischen Programmen (die sogar in verschiedenen Umgebungen ausgeführt werden können), ohne dass Kommunikationscode geschrieben werden muss.
- Auswahl der Reihenfolge, in welcher ein Programm Nachrichten verarbeitet.
- Ausgleich der Arbeitslast in einem System durch Festlegung mehrerer Programme, die für eine Warteschlange zuständig sind, wenn die Anzahl der Nachrichten einen Grenzwert überschreitet.

- Erhöhte Verfügbarkeit der Anwendungen, durch Festlegung eines alternativ für die Warteschlangen zuständigen Systems, wenn das primäre System nicht verfügbar ist.

## Was ist unter einer Nachrichtenwarteschlange zu verstehen?

Bei einer Nachrichtenwarteschlange, die auch einfach als Warteschlange bezeichnet wird, handelt es sich um ein benanntes Ziel, an das Nachrichten gesendet werden können. Nachrichten sammeln sich in Warteschlangen an, bis sie von Programmen abgerufen werden, die für diese Warteschlangen zuständig sind.

Warteschlangen befinden sich auf einem Warteschlangenmanager, von dem sie auch verwaltet werden (siehe Abschnitt „Terminologie zum Message-Queuing“ auf Seite 49). Die physische Natur einer Warteschlange hängt von dem Betriebssystem ab, unter dem der Warteschlangenmanager ausgeführt wird. Warteschlangen können flüchtige Pufferbereiche im Hauptspeicher eines Computers oder aber Datenmengen in einer permanenten Speichereinheit (wie z. B. einer Festplatte) sein. Die physische Verwaltung der Warteschlangen ist Aufgabe der Warteschlangenmanager, für die beteiligten Anwendungsprogramme ist sie nicht offensichtlich.

Programme greifen auf Warteschlangen lediglich über die externen Services des Warteschlangenmanagers zu. Sie können eine Warteschlange öffnen, Nachrichten darin einreihen und daraus abrufen und die Warteschlange schließen. Außerdem können sie die Warteschlangenattribute festlegen und abfragen.

## Verschiedene Arten des Message-Queuing

### Punkt-zu-Punkt

Eine Nachricht wird in die Warteschlange eingereiht und dann von einer Anwendung empfangen.

Beim Punkt-zu-Punkt-Messaging muss eine sendende Anwendung Informationen zu einer anderen empfangenden Anwendung erhalten, damit sie eine Nachricht an diese Anwendung senden kann. Die sendende Anwendung muss beispielsweise den Namen der Warteschlange kennen, an die die Informationen gesendet werden sollen, und möglicherweise muss auch der Name eines Warteschlangenmanagers angegeben werden.

### Publish/Subscribe

Jeder interessierten Anwendung wird eine Kopie jeder Nachricht bereitgestellt, die von einer publizierenden Anwendung veröffentlicht wird. Dabei ist die Anzahl der interessierten Anwendungen - keine, eine oder mehrere - unbedeutend. Beim Publish/Subscribe wird eine interessierte Anwendung als Subskribent bezeichnet, und die Nachrichten werden in eine durch eine Subskription festgelegte Warteschlange eingereiht.

Mithilfe des Publish/Subscribe-Messaging kann der Informationsanbieter von den Nutzern der Informationen abgekoppelt werden. Die Sender- und Empfängeranwendungen müssen zur Übertragung der Informationen nichts übereinander wissen. Weitere Informationen zum Publish/Subscribe-Messaging finden Sie im Abschnitt [Einführung in das WebSphere MQ-Publish/Subscribe-Messaging](#)

## Vorteile des Message-Queuing für Anwendungsentwickler

In WebSphere MQ können Anwendungsprogramme *Message-Queuing* einsetzen und somit an einer nachrichtengesteuerten Verarbeitung teilnehmen. Unter Verwendung der entsprechenden Message-Queuing-Softwareprodukte können Anwendungsprogramme über verschiedene Plattformen hinweg miteinander kommunizieren. So ist beispielsweise eine Kommunikation von HP-UX- und z/OS-Anwendungen mithilfe von WebSphere MQ for HP-UX und WebSphere MQ for z/OS möglich. Die Anwendungen sind von den Mechanismen der zugrunde liegenden Kommunikation abgeschirmt. Einige weitere Vorteile des Message-Queuing:

- Es ist möglich, für die Anwendungsentwicklung kleine Programme einzusetzen, die von vielen Anwendungen gemeinsam genutzt werden können.
- Durch die Wiederverwendung dieser logischen Bausteine können schnell neue Anwendungen erstellt werden.

- Anwendungen, die für die Verwendung von Message-Queuing-Verfahren geschrieben wurden, sind von Änderungen in der Funktionsweise der Warteschlangenmanager nicht betroffen.
- Es werden keine Kommunikationsprotokolle benötigt. Alle Aspekte der Kommunikation werden vom Warteschlangenmanager übernommen.
- Programme, die Nachrichten empfangen, müssen nicht aktiv sein, wenn ihnen Nachrichten zugesendet werden. Die Nachrichten verbleiben in Warteschlangen.

Geringere Kosten für die Anwendungen, da eine schnellere Entwicklung mit weniger Entwicklern möglich ist und weniger Programmierkenntnisse erforderlich sind als für Anwendungen, die Message-Queuing nicht einsetzen.

Überall, wo die Anwendungen ausgeführt werden, implementiert WebSphere MQ eine einheitliche Anwendungsprogrammierschnittstelle, das sogenannte *Message Queue Interface* (MQI). Auf diese Weise können Anwendungsprogramme leichter auf andere Plattformen portiert werden.

Details zum Message Queue Interface finden Sie im Abschnitt [Übersicht über das Message Queue Interface](#).

## Hauptmerkmale und Vorteile des Message-Queuing

In diesem Abschnitt werden einige Merkmale und Vorteile des Message-Queuing hervorgehoben. Sie finden hier beispielsweise eine Beschreibung der Message-Queuing-Merkmale in Bezug auf Sicherheit und Datenintegrität.

Anwendungen, die Message-Queuing-Verfahren einsetzen, weisen folgende Hauptmerkmale auf:

- Es bestehen keine Direktverbindungen zwischen Programmen.
- Die Kommunikation zwischen Programmen kann zeitunabhängig erfolgen.
- Die Arbeit kann von kleinen eigenständigen Programmen ausgeführt werden.
- Die Kommunikation kann ereignisgesteuert sein.
- Anwendungen können Nachrichten Prioritäten zuweisen.
- Sicherheit.
- Datenintegrität.
- Unterstützung bei der Wiederherstellung.

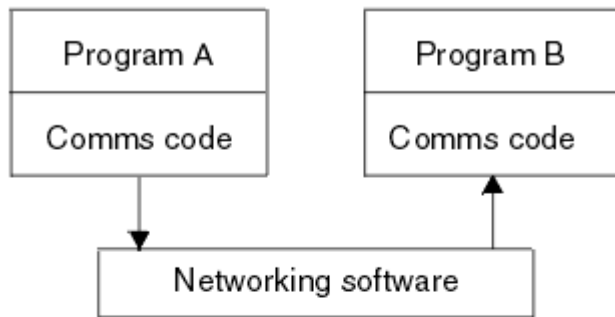
### Keine Direktverbindungen zwischen Programmen

Message-Queuing stellt ein Verfahren für die indirekte Kommunikation zwischen Programmen dar. Dieses Verfahren kann in jeder Anwendung eingesetzt werden, in der Programme miteinander kommunizieren. Die Kommunikation erfolgt, indem ein Programm Nachrichten in eine (einem Warteschlangenmanager zugehörige) Warteschlange einreicht und ein anderes Programm die Nachrichten aus der Warteschlange abrufft.

Programme können Nachrichten abrufen, die von anderen Programmen in eine Warteschlange eingereicht wurden. Die anderen Programme können mit demselben Warteschlangenmanager wie das empfangende Programm oder aber mit einem anderen Warteschlangenmanager verbunden sein. Dieser andere Warteschlangenmanager kann sich auf einem anderen System, in einem anderen Computersystem oder sogar in einem anderen Unternehmen befinden.

Zwischen Programmen, die Nachrichtenwarteschlangen für die Kommunikation miteinander einsetzen, bestehen keine physischen Verbindungen. Ein Programm sendet Nachrichten an eine Warteschlange eines Warteschlangenmanagers und ein anderes Programm ruft Nachrichten aus der Warteschlange ab (siehe Abschnitt [Abbildung 4 auf Seite 48](#)).

### Traditional communication between programs



### Communication by message queuing

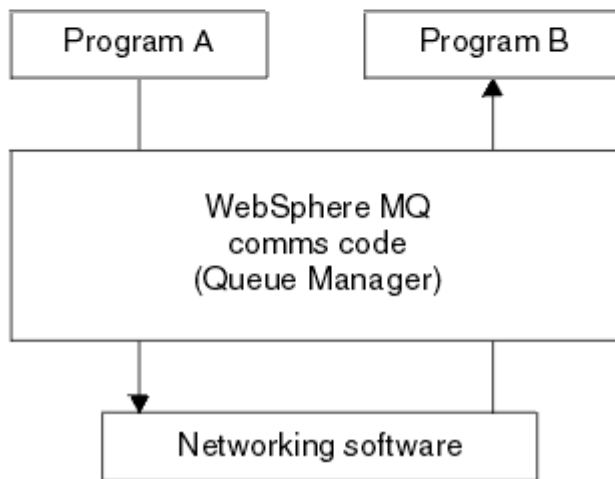


Abbildung 4. Message-Queuing im Vergleich zur konventionellen Kommunikation

Wie bei E-Mail durchqueren die einzelnen zu einer Transaktion gehörigen Nachrichten ein Netz auf Basis des Store-and-forward-Verfahrens. Schlägt eine Verbindung zwischen Knoten fehl, bleibt die Nachricht erhalten, bis die Verbindung wiederhergestellt ist oder der Bediener bzw. das Programm die Nachricht umleitet.

Der Mechanismus, mit dem Nachrichten von einer Warteschlange zur nächsten gelangen, bleibt vor den Programmen verborgen. Dies vereinfacht die Programme.

#### **Zeitunabhängige Kommunikation**

Programme, die andere Programme zur Ausführung von Arbeiten auffordern, müssen nicht auf die Antwort zu der Anforderung warten. Sie können weitere Arbeiten erledigen und die Antwort bei Erhalt oder zu einem späteren Zeitpunkt verarbeiten. Beim Schreiben einer Messaging-Anwendung ist es irrelevant, wann ein Programm eine Nachricht sendet bzw. wann das Ziel die Nachricht empfangen kann. Die Nachricht geht nicht verloren, sie wird vom Warteschlangenmanager beibehalten, bis sie vom Ziel verarbeitet werden kann. Die Nachricht bleibt in der Warteschlange, bis sie von einem Programm daraus entfernt wird. Die sendende und die empfangende Anwendung sind also voneinander abgekoppelt. Der Sender kann seine Verarbeitung fortsetzen und muss nicht warten, bis der Empfänger den Erhalt der Nachricht bestätigt. Die Zielanwendung muss nicht einmal aktiv sein, wenn die Nachricht gesendet wird. Sie kann die Nachricht nach dem Start abrufen.

#### **Kleine Programme**

Bei Einsatz des Message-Queuing können Sie die Vorteile kleiner unabhängiger Programme nutzen. Anstelle eines einzelnen umfangreichen Programms, das nacheinander alle Teile eines Jobs ausführt, kann der Job auf mehrere kleinere unabhängige Programme verteilt werden. Das aufrufende Programm sendet Nachrichten an jedes einzelne dieser separaten Programme und fordert diese zur



Ausführung ihrer jeweiligen Funktion auf. Sobald die einzelnen Programme mit der Verarbeitung fertig sind, werden die Ergebnisse in Form einer oder mehrerer Nachrichten zurückgesendet.

### **Nachrichtengesteuerte Verarbeitung**

Wenn Nachrichten in einer Warteschlange ankommen, können sie mithilfe der *Auslösefunktion* automatisch eine Anwendung starten. Gegebenenfalls können die Anwendungen nach erfolgter Verarbeitung der Nachricht(en) gestoppt werden.

### **Ereignisgesteuerte Verarbeitung**

Programme können entsprechend dem Warteschlangenstatus gesteuert werden. So können Sie beispielsweise festlegen, dass bei Eintreffen einer Nachricht in einer Warteschlange ein Programm gestartet werden soll, oder Sie können angeben, dass das Programm erst gestartet werden soll, wenn sich in der Warteschlange beispielsweise 10 Nachrichten über einer gewissen Priorität oder aber mit einer beliebigen Priorität befinden.

### **Nachrichtenpriorität**

Wenn ein Programm eine Nachricht in eine Warteschlange einreicht, kann es der Nachricht eine Priorität zuordnen. Diese legt fest, an welcher Stelle in der Warteschlange die neue Nachricht hinzugefügt wird.

Programme können entweder die Nachrichten der Reihenfolge nach aus einer Warteschlange abrufen oder aber eine bestimmte Nachricht abrufen. (So könnte es z. B. sein, dass ein Programm eine bestimmte Nachricht abrufen möchte, wenn es auf die Antwort auf eine zuvor gesendete Anforderung wartet.)

### **Sicherheit**

Für jede Ressource werden unter Verwendung der vom WebSphere MQ-Administrator eingerichteten und verwalteten Tabellen Berechtigungsprüfungen vorgenommen.

- Verwenden Sie Security Server (bisher unter der Bezeichnung RACF) oder andere externe Sicherheitsmanager unter WebSphere MQ für z/OS.
- Wenn WebSphere MQ auf UNIX-, Linux-, Windows- und IBM i-Systemen ausgeführt wird, ist ein Sicherheitsmanager, der sogenannte Objektberechtigungsmanager, als installierbarer Service verfügbar. Standardmäßig ist der Objektberechtigungsmanager aktiv.

### **Datenintegrität**

Datenintegrität wird über Arbeitseinheiten bereitgestellt. Die Synchronisation des Anfangs und Endes von Arbeitseinheiten wird bei jedem Aufruf MQGET oder MQPUT als Option vollständig unterstützt, sodass die Ergebnisse der Arbeitseinheit festgeschrieben bzw. rückgängig gemacht werden können. Die Synchronisationspunktunterstützung funktioniert entweder WebSphere MQ-intern oder -extern, je nachdem, welche Art der Synchronisationspunktorganisation für die Anwendung ausgewählt wurde.

### **Unterstützung bei der Wiederherstellung**

Um eine Wiederherstellung zu ermöglichen, werden alle permanenten WebSphere MQ-Updates protokolliert. Sollte eine Wiederherstellung erforderlich sein, werden alle persistenten Nachrichten zurückgeschrieben, für alle unvollständigen Transaktionen erfolgt ein Rollback und alle Synchronisationspunktfestschreibungen und -Backouts werden nach dem für den zuständigen Synchronisationspunktmanager normalen Verfahren bearbeitet. Weitere Informationen zu persistenten Nachrichten finden Sie unter [Nachrichtenpersistenz](#).

**Anmerkung:** Für WebSphere MQ-Clients und -Server gilt Folgendes: Es ist keine Änderung einer Serveranwendung erforderlich, damit weitere WebSphere MQ MQI-Clients auf neuen Plattformen unterstützt werden können. In ähnlicher Weise kann der WebSphere MQ MQI-Client ohne Änderung mit weiteren Servertypen eingesetzt werden.

## **Terminologie zum Message-Queuing**

In diesem Abschnitt werden einige Begriffe im Zusammenhang mit Message-Queuing erläutert.

Dazu gehören:

- [Nachricht](#)
- [Nachrichtendeskriptor](#)

- [Warteschlangen](#)
- [Warteschlangenmanager](#)
- [Kanäle](#)
- [Nachrichtenkanalagent](#)
- [Cluster](#)
- [WebSphere MQ MQI-Client](#)
- [Punkt-zu-Punkt](#)
- [Publish/Subscribe](#)
- [Thema](#)
- [Subskription](#)

## Nachricht

Beim Message-Queuing bezeichnet der Begriff 'Nachricht' eine Datensammlung, die von einem Programm gesendet wurde und an ein anderes Programm gerichtet ist. Siehe [IBM WebSphere MQ-Nachrichten](#). Informationen zu Nachrichtentypen finden Sie im Abschnitt [Nachrichtentypen](#).

## Nachrichtendeskriptor

Eine IBM WebSphere MQ-Nachricht besteht aus Steuerinformationen und Anwendungsdaten.

Die Steuerinformationen sind in einer Nachrichtendeskriptorstruktur (MQMD) definiert und umfassen u. a. folgende Angaben:

- Nachrichtentyp
- ID für die Nachricht
- Zustellungspriorität der Nachricht

Struktur und Inhalt der Anwendungsdaten werden durch die beteiligten Programme und nicht durch IBM WebSphere MQ bestimmt.

## Warteschlange

Ein benanntes Ziel, an das Nachrichten gesendet werden können. Nachrichten sammeln sich in Warteschlangen an, bis sie von Programmen abgerufen werden, die für diese Warteschlangen zuständig sind.

## Warteschlangenmanager

Ein *Warteschlangenmanager* ist ein Systemprogramm, welches Warteschlangensteuerungsservices für Anwendungen bereitstellt.

Der Warteschlangenmanager stellt eine Anwendungsprogrammierschnittstelle zur Verfügung, über welche Programme Nachrichten in Warteschlangen einreihen und daraus abrufen können. Darüber hinaus bietet der Warteschlangenmanager weitere Funktionen, mit denen Administratoren neue Warteschlangen erstellen, die Eigenschaften bestehender Warteschlangen ändern und den Betrieb des Warteschlangenmanagers steuern können.

Die Message-Queueing-Services von IBM WebSphere MQ sind auf einem System nur verfügbar, wenn ein Warteschlangenmanager aktiv ist. Auf einem einzelnen System können mehrere Warteschlangenmanager aktiv sein (beispielsweise um ein Testsystem von einem *Livesystem* zu trennen). Jeder Warteschlangenmanager wird gegenüber einer Anwendung über eine *Verbindungskennung (Hconn)* identifiziert.

Die Services des Warteschlangenmanagers können von vielen verschiedenen Anwendungen, die komplett unabhängig voneinander sein können, gleichzeitig verwendet werden. Damit ein Programm die Services eines Warteschlangenmanagers verwenden kann, muss es eine Verbindung zu dem betreffenden Warteschlangenmanager herstellen.

Anwendungen können Nachrichten nur dann an Anwendungen senden, die mit anderen Warteschlangenmanagern verbunden sind, wenn die betreffenden Warteschlangenmanager in der Lage sind,

miteinander zu kommunizieren. Für die sichere Nachrichtenübermittlung zwischen solchen Anwendungen implementiert IBM WebSphere MQ ein *Store-and-forward*-Protokoll.

## Kanäle

*Kanäle* sind Objekte, die einen Kommunikationspfad von einem Warteschlangenmanager zu einem anderen bereitstellen. Kanäle werden bei der verteilten Steuerung von Warteschlangen für die Übertragung von Nachrichten zwischen Warteschlangenmanagern verwendet und schirmen Anwendungen von den verwendeten Kommunikationsprotokollen ab. Die Warteschlangenmanager können sich dabei auf derselben oder auf verschiedenen Plattformen befinden.

## Nachrichtenkanalagent

Ein Nachrichtenkanalagent verschiebt Nachrichten von einem Warteschlangenmanager auf einen anderen.

Im Zusammenhang mit Berichtsnachrichten wird darauf verwiesen und auch bei der Anwendungsentwicklung sind sie zu berücksichtigen. Weitere Informationen finden Sie im Abschnitt [Eigene Nachrichtenkanalagenten schreiben](#).

## Cluster

Ein *Cluster* ist ein Netz von Warteschlangenmanagern, die logisch in gewisser Weise zusammengehören. Clustering ist für Warteschlangenmanager in IBM WebSphere MQ Version 7.0 und neueren Versionen verfügbar.

In einem IBM WebSphere MQ-Netz mit verteilter Steuerung von Warteschlangen ohne Clustering ist jeder Warteschlangenmanager unabhängig. Wenn ein Warteschlangenmanager Nachrichten an einen anderen Warteschlangenmanager senden muss, müssen eine Übertragungswarteschlange sowie ein Kanal zu dem fernen Warteschlangenmanager definiert sein.

Für die Verwendung von Clustern gibt es zwei Gründe: geringerer Systemverwaltungsaufwand sowie verbesserte Verfügbarkeit und besserer Lastausgleich.

Schon bei Einrichtung eines sehr kleinen Clusters profitieren Sie von einer vereinfachten Systemverwaltung. Zu einem Cluster gehörige Warteschlangenmanager benötigen weniger Definitionen, wodurch sich das Risiko von Fehlern bei den Definitionen verringert.

Weitere Informationen zum Clustering finden Sie im Abschnitt [„Wie funktionieren Cluster?“](#) auf Seite 68.

## IBM WebSphere MQ MQI client

IBM WebSphere MQ MQI-*Clients* sind unabhängig voneinander installierbare Komponenten von IBM WebSphere MQ. Auf einem MQI-Client können IBM WebSphere MQ-Anwendungen über ein Kommunikationsprotokoll ausgeführt werden, es ist eine Interaktion mit einem oder mehreren MQI-Servern (MQI = Message Queue Interface) auf anderen Plattformen möglich und es kann eine Verbindung zu den zugehörigen Warteschlangenmanagern hergestellt werden.

Ausführliche Informationen zur Installation und Verwendung von IBM WebSphere MQ MQI client-Komponenten finden Sie in den Abschnitten [IBM WebSphere MQ MQI client installieren](#) und [Verbindungen zwischen Server und Client konfigurieren](#).

## Punkt-zu-Punkt-Messaging

Beim Punkt-zu-Punkt-Messaging wird jede Nachricht von einer generierenden Anwendung an eine konsumierende Anwendung übertragen. Zur Übertragung stellt die generierende Anwendung die Nachricht in eine Warteschlange, aus der die konsumierende Anwendung die Nachricht abrufen.

## Publish/Subscribe-Messaging

Beim Publish/Subscribe-Messaging wird jeder interessierten Anwendung eine Kopie jeder Nachricht bereitgestellt, die von einer publizierenden Anwendung veröffentlicht wird. Dabei ist die Anzahl der interessierten Anwendungen - keine, eine oder mehrere - unbedeutend. Beim Publish/Subscribe wird eine interessierte Anwendung als Subskribent bezeichnet, und die Nachrichten werden in eine durch eine Subskription festgelegte Warteschlange eingereiht. Weitere Informationen zu Publish/Subscribe finden Sie im Abschnitt [Einführung in das IBM WebSphere MQ-Publish/Subscribe-Messaging](#).

## Thema

Ein Thema ist eine Zeichenfolge, die den Gegenstand der Informationen beschreibt, die in einer Publish/Subscribe-Nachricht veröffentlicht werden.

Themen sind wichtig für die erfolgreiche Nachrichtenübermittlung in einem Publish/Subscribe-System. Anstatt eine bestimmte Zieladresse in die einzelnen Nachrichten einzufügen, ordnet ein Publisher jeder Nachricht ein Thema zu. Der Warteschlangenmanager gleicht das Thema mit einer Liste von Abonnenten ab, die das Thema abonniert haben, und stellt jedem dieser Abonnenten die Nachricht zu.

## Subskription

Eine Publish/Subscribe-Anwendung kann ihr Interesse an Nachrichten zu bestimmten Themen registrieren lassen. Mit der Registrierung wird die Anwendung zu einem Subskribenten. Der Begriff 'Subskription' definiert dabei, auf welche Weise passende Nachrichten zur Verarbeitung in Warteschlangen eingereicht werden.

Eine Subskription enthält Informationen zur Identität des Subskribenten und der Zielwarteschlange, auf der Veröffentlichungen eingereicht werden sollen. Sie enthält außerdem Informationen darüber, wie eine Veröffentlichung in der Zielwarteschlange eingereicht wird.

## Nachrichten und Warteschlangen

Nachrichten und Warteschlangen sind die Basiskomponenten eines Message-Queuing-Systems.

### Was ist eine Nachricht?

Eine *Nachricht* ist eine Bytefolge, die Informationen für die Anwendungen bereitstellt, die diese Nachricht verwendet. Mithilfe von Nachrichten werden Informationen von einem Anwendungsprogramm an ein anderes oder zwischen den verschiedenen Teilen ein und derselben Anwendung übertragen. Die Anwendungen können dabei auf derselben oder auf verschiedenen Plattformen aktiv sein.

IBM WebSphere MQ-Nachrichten setzen sich aus zwei Teilen zusammen:

- *Den Anwendungsdaten.* Inhalt und Struktur der Anwendungsdaten werden vom Anwendungsprogramm vorgegeben, das diese Daten nutzt.
- *Einem Nachrichtendeskriptor.* Der Nachrichtendeskriptor dient zur Kennzeichnung der Nachricht und enthält zusätzliche Steuerinformationen, z. B. den Nachrichtentyp und die Priorität, die der Nachricht von der sendenden Anwendung zugewiesen wurde.

Das Format des Nachrichtendeskriptors wird von IBM WebSphere MQ definiert. Eine umfassende Beschreibung des Nachrichtendeskriptors finden Sie im Abschnitt [MQMD – Nachrichtendeskriptor](#).

### Nachrichtenlänge

Die maximale Nachrichtenlänge ist standardmäßig 4 MB; diesen Wert können Sie auf maximal 100 MB (1 MB = 1,048,576 Byte) erhöhen. Tatsächlich kann die Nachrichtenlänge durch folgende Faktoren begrenzt werden:

- Die für die Empfangswarteschlange definierte maximale Nachrichtenlänge
- Die für den Warteschlangenmanager definierte maximale Nachrichtenlänge
- Die von der Warteschlange definierte maximale Nachrichtenlänge
- Die von der sendenden oder empfangenden Anwendung definierte maximale Nachrichtenlänge
- Dem für die Nachricht verfügbaren Speicherplatz

In manchen Fällen sind mehrere Nachrichten erforderlich, um alle notwendigen Informationen an eine Anwendung zu übertragen.

## Wie werden Nachrichten von Anwendungen gesendet und empfangen?

Anwendungsprogramme senden und empfangen Nachrichten über **MQI-Aufrufe**.

Um beispielsweise eine Nachricht in eine Warteschlange einzureihen, sind folgende Schritte erforderlich:

1. Die Anwendung öffnet die betreffende Warteschlange, indem sie den MQ-Aufruf MQOPEN ausgibt.
2. Anschließend gibt die Anwendung den MQI-Aufruf MQPUT aus, um die Nachricht in die Warteschlange einzureihen.

Diese Nachricht kann von einer anderen Anwendung mit dem MQI-Aufruf MQGET aus der Warteschlange abgerufen werden.

Weitere Informationen zu MQI-Aufrufen finden Sie im Abschnitt [MQI-Aufrufe](#).

## Was ist eine Warteschlange?

Eine *Warteschlange* ist eine Datenstruktur, in der Nachrichten gespeichert werden.

Jede Warteschlange ist einem *Warteschlangenmanager* zugeordnet, der damit Eigner der Warteschlange ist. Der Warteschlangenmanager ist zuständig für die Verwaltung aller ihm zugeordneten Warteschlangen; er muss außerdem alle Nachrichten, die er empfängt, in den entsprechenden Warteschlangen speichern. Die Nachrichten können entweder von den Anwendungsprogrammen oder vom Warteschlangenmanager (als Teil des normalen Warteschlangenmanagerbetriebs) in die Warteschlangen eingereiht werden.

## Vordefinierte und dynamische Warteschlangen

Warteschlangen werden anhand der Art und Weise ihrer Erstellung unterschieden:

- **Vordefinierte Warteschlangen** werden vom Administrator mit den entsprechenden MQSC- oder PCF-Befehlen erstellt. Vordefinierte Warteschlangen sind permanent; sie sind unabhängig von den Anwendungen vorhanden, von denen sie verwendet werden, und sind auch nach einem Neustart von IBM WebSphere MQ noch vorhanden.
- **Dynamische Warteschlangen** werden erstellt, wenn eine Anwendung eine MQOPEN-Anforderung unter Angabe des Namens einer *Modellwarteschlange* ausgibt. Die Warteschlange wird auf Basis einer *Definition für eine Warteschlangenschablone*, einer sogenannten Modellwarteschlange, erstellt. Modellwarteschlangen können mit dem WebSphere MQ-Scriptbefehl DEFINE QMODEL erstellt werden. Die Attribute einer Modellwarteschlange (beispielsweise die Anzahl der Nachrichten, die maximal in der Warteschlange gespeichert werden können) werden von allen dynamischen Warteschlangen übernommen, die auf Basis dieser Modellwarteschlange erstellt werden.

Modellwarteschlangen haben Attribute, die angeben, ob die dynamische Warteschlange permanent oder temporär sein soll. Permanente Warteschlangen sind auch nach dem Neustart einer Anwendung oder des Warteschlangenmanagers noch vorhanden, temporäre Warteschlangen hingegen nicht.

## Nachrichten aus Warteschlangen abrufen

Anwendungen mit der entsprechenden Berechtigung können Nachrichten anhand der folgenden Abrufalgorithmen aus einer Warteschlange abrufen:

- FIFO (First In/First Out).
- Nachrichtenpriorität (ist im Nachrichtendeskriptor festgelegt). Nachrichten mit derselben Priorität werden nach dem FIFO-Prinzip (First In/First Out) abgerufen.
- Programmanforderung für eine bestimmte Nachricht.

Der Algorithmus ist in der MQGET-Anforderung der Anwendung festgelegt.

## Konzepte der übergreifenden Kommunikation

In WebSphere MQ bezeichnet übergreifende Kommunikation das Senden von Nachrichten von einem Warteschlangenmanager an einen anderen. Der empfangende Warteschlangenmanager kann sich auf

demselben System oder auf einem anderen System ganz in der Nähe oder am anderen Ende der Welt befinden. Er kann auf derselben Plattform wie der lokale Warteschlangenmanager oder jeder anderen von WebSphere MQ unterstützten Plattform aktiv sein. Dies wird als *verteilte* Umgebung bezeichnet. WebSphere MQ wickelt die Kommunikation in einer verteilten Umgebung wie dieser mithilfe des verteilten Warteschlangenmanagements ab.

Der lokale Warteschlangenmanager wird manchmal auch *Quellen-Warteschlangenmanager* und der ferne Warteschlangenmanager manchmal auch *Ziel-Warteschlangenmanager* oder *Partner-Warteschlangenmanager* genannt.

## Funktionsweise der verteilten Steuerung von Warteschlangen

Mithilfe der verteilten Steuerung von Warteschlangen können Nachrichten von einem Warteschlangenmanager an einen anderen gesendet werden. Der empfangende Warteschlangenmanager kann sich auf demselben System oder einem fernen System befinden. Warteschlangenmanager, Warteschlangen, Kanäle und zugehörige Definitionen werden in Verbindung mit dem Clustering (ein Netz aus logisch zugeordneten Warteschlangenmanagern) beschrieben.

Abbildung 5 auf Seite 54 zeigt eine Übersicht über die Komponenten der verteilten Steuerung von Warteschlangen.

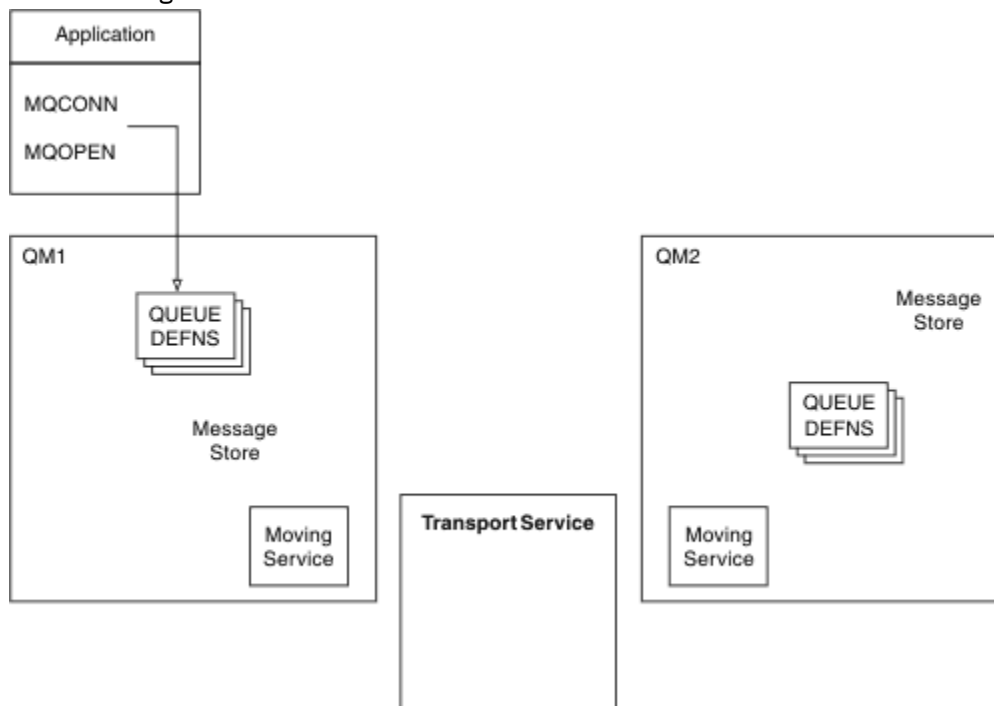


Abbildung 5. Übersicht über die Komponenten der verteilten Steuerung von Warteschlangen

1. Eine Anwendung stellt mit dem Aufruf MQCONN eine Verbindung mit einem Warteschlangenmanager her.
2. Danach öffnet die Anwendung mit dem Aufruf MQOPEN eine Warteschlange, sodass sie Nachrichten darin einreihen kann.
3. Ein Warteschlangenmanager besitzt für jede seiner Warteschlangen eine Definition mit Informationen wie die maximal zulässige Anzahl von Nachrichten in der Warteschlange. Er kann auch über Definitionen von Warteschlangen verfügen, die sich auf fernen Warteschlangenmanagern befinden.
4. Wenn die Nachrichten für eine Warteschlange auf einem fernen System bestimmt sind, legt der lokale Warteschlangenmanager sie in einem Nachrichtenspeicher ab, bis er bereit ist, sie an den fernen Warteschlangenmanager weiterzuleiten. Dies hat keine Auswirkung auf die Anwendung.
5. Jeder Warteschlangenmanager enthält eine Datenübertragungssoftware, die sogenannte *Übertragungsservice*-Komponente, über die der Warteschlangenmanager mit anderen Warteschlangenmanagern kommunizieren kann.

6. Der *Transportservice* ist vom Warteschlangenmanager unabhängig, sodass es sich um jeden der folgenden Services (je nach Plattform) handeln kann:

- Systems Network Architecture Advanced Program-to Program Communication (SNA APPC)
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- Network Basic Input/Output System (NetBIOS)
- Sequenced Packet Exchange (SPX)

## Komponenten der verteilten Steuerung von Warteschlangen

WebSphere MQ-Anwendungen können Nachrichten in eine lokale Warteschlange stellen, d. h. in eine Warteschlange des Warteschlangenmanagers, mit dem die Anwendung verbunden ist.

Ein Warteschlangenmanager besitzt für jede seiner Warteschlangen eine Definition. Er kann auch über Definitionen für Warteschlangen verfügen, die Eigentum anderer Warteschlangenmanager sind. Diese werden als *Definitionen ferner Warteschlangen* bezeichnet. WebSphere MQ-Anwendungen können auch Nachrichten einreihen, die für diese fernen Warteschlangen bestimmt sind.

Wenn die Nachrichten für einen fernen Warteschlangenmanager bestimmt sind, speichert der lokale Warteschlangenmanager sie in einer *Übertragungswarteschlange*, bis er bereit ist, sie an den fernen Warteschlangenmanager zu senden. Eine Übertragungswarteschlange ist ein besonderer Typ der lokalen Warteschlange, in der Nachrichten gespeichert werden, bis sie erfolgreich übertragen und beim fernen Warteschlangenmanager gespeichert werden können.

Die Software, die das Senden und Empfangen von Nachrichten ausführt, wird als *Nachrichtenkanalagent* bezeichnet.

Nachrichten werden in einem *Kanal* zwischen Warteschlangenmanagern übertragen. Ein Kanal ist eine einseitige Übertragungsverbindung zwischen zwei Warteschlangenmanagern. In ihm können Nachrichten, die für beliebig viele Warteschlangen des fernen Warteschlangenmanagers bestimmt sind, übertragen werden.

## Zum Senden einer Nachricht benötigte Komponenten

>Um eine Nachricht an einen fernen Warteschlangenmanager senden zu können, benötigt der lokale Warteschlangenmanager Definitionen für eine Übertragungswarteschlange und einen Kanal.

Es gibt für jedes Ende eines Kanals eine separate Definition, in der es zum Beispiel als Sendeseite oder Empfangsseite definiert ist. Ein einfacher Kanal besteht aus einer *Senderkanaldefinition* beim lokalen Warteschlangenmanager und einer *Empfängerkanaldefinition* beim fernen Warteschlangenmanager. Diese beiden Definitionen müssen den gleichen Namen haben und gemeinsam einen einzigen Kanal bilden.

Außerdem gibt es an jedem Ende eines Kanals einen *Nachrichtenkanalagenten*.

Jeder Warteschlangenmanager muss eine *Warteschlange für nicht zustellbare Nachrichten* (auch *Warteschlange für nicht zugestellte Nachrichten* genannt) besitzen. In diese Warteschlange werden Nachrichten gestellt, die nicht an ihre Zieladresse übermittelt werden können.

Abbildung 6 auf Seite 56 zeigt die Beziehung zwischen Warteschlangenmanagern, Übertragungswarteschlangen, Kanäle und Nachrichtenkanalagenten.

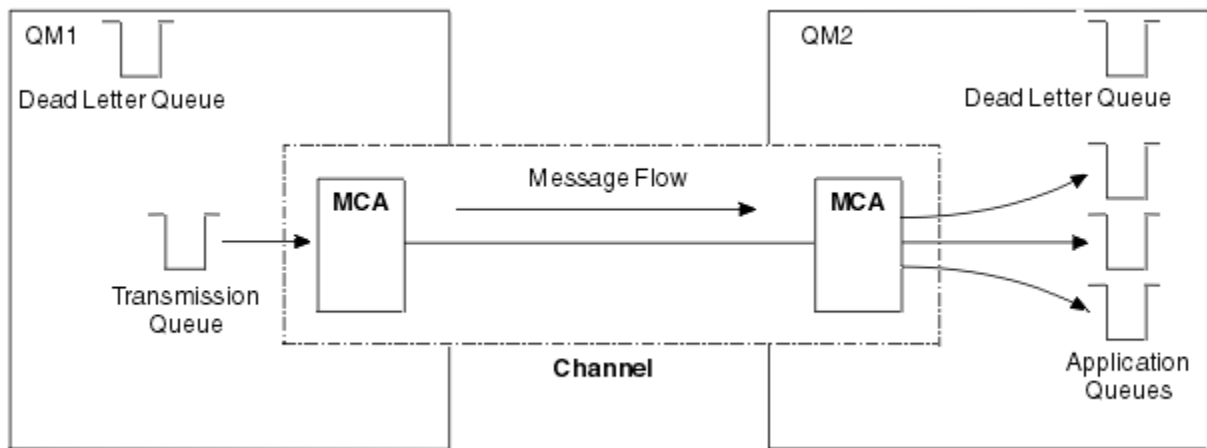


Abbildung 6. Nachrichten senden

### Zur Rückgabe einer Nachricht benötigte Komponenten

Wenn Ihre Anwendung erfordert, dass Nachrichten vom fernen Warteschlangenmanager zurückgegeben werden, müssen Sie einen weiteren Kanal definieren, der in entgegengesetzter Richtung zwischen den Warteschlangenmanagern verläuft (siehe [Abbildung 7](#) auf Seite 56).

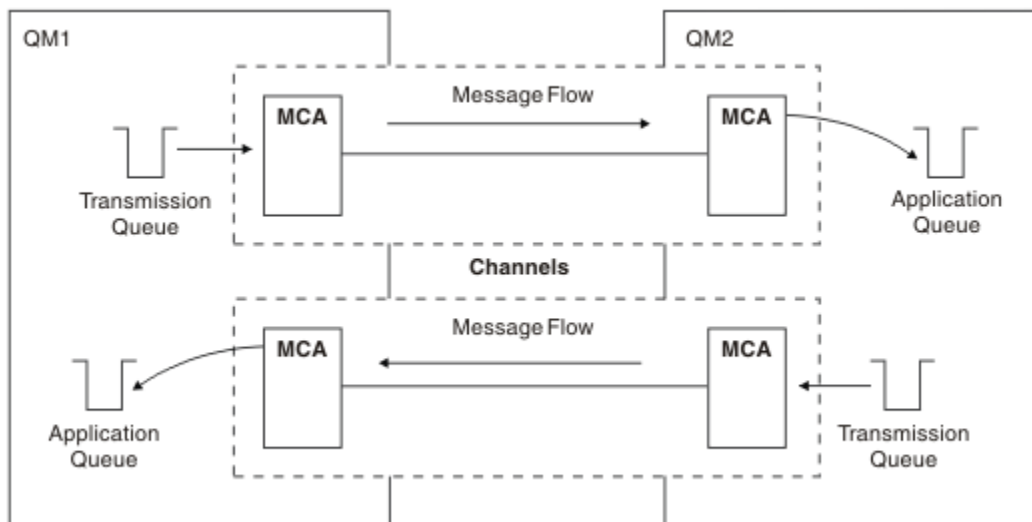


Abbildung 7. Nachrichten in beiden Richtungen senden

Weitere Informationen zum verteilten Warteschlangenmanagement finden Sie im Abschnitt [Einführung in verteiltes Warteschlangenmanagement](#).

### Clusterkomponenten

Eine Alternative zum traditionellen WebSphere MQ-Netz, das durch die manuelle Definition von Kanälen aufgebaut wird, ist die Verwendung von Clustern.

Ein Cluster ist ein Netz aus Warteschlangenmanagern, die logisch miteinander verbunden sind. Sie können Warteschlangenmanager zu einem Cluster gruppieren, sodass Warteschlangenmanager die von ihnen verwalteten Warteschlangen jedem anderen Warteschlangenmanager im Cluster zur Verfügung stellen können. Unter der Voraussetzung, dass die notwendige Netzinfrastruktur vorhanden ist, kann jeder Warteschlangenmanager eine Nachricht an jeden anderen Warteschlangenmanager in demselben Cluster senden, ohne dass für jedes Ziel explizite Kanaldefinitionen, Definitionen ferner Warteschlangen oder Übertragungswarteschlangen vorhanden sein müssen. Jeder Warteschlangenmanager in einem Cluster



besitzt eine einzelne Übertragungswarteschlange, um Nachrichten an einen anderen Warteschlangenmanager im Cluster zu übertragen. Jeder Warteschlangenmanager muss nur einen Clusterempfängerkanal und nur einen Clustersenderkanal definieren; alle zusätzlichen Kanäle werden automatisch vom Cluster verwaltet.

Abbildung 8 auf Seite 57 zeigt die Komponenten eines Clusters mit dem Namen CLUSTER:

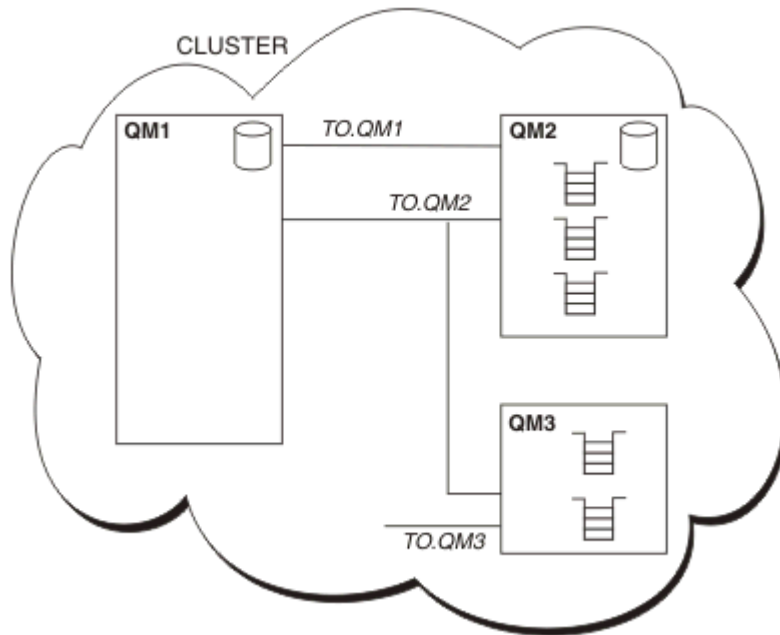


Abbildung 8. Warteschlangenmanagercluster

- CLUSTER enthält die drei Warteschlangenmanager QM1, QM2 und QM3.
- QM1 und QM2 verwalten vollständige Repositorys mit Informationen über die Warteschlangenmanager und Warteschlangen im Cluster.
- QM2 und QM3 verwalten einige Clusterwarteschlangen, d. h. Warteschlangen, auf die jeder andere Warteschlangenmanager im Cluster zugreifen kann.
- Jeder Warteschlangenmanager besitzt einen Clusterempfängerkanal mit dem Namen TO.qmgr, über den er Nachrichten empfangen kann.
- Außerdem besitzt jeder Warteschlangenmanager einen Clustersenderkanal, über den er Informationen an einen der Repository-Warteschlangenmanager senden kann.
- QM1 und QM3 senden Informationen an das Repository von QM2 und QM2 sendet Informationen an das Repository von QM1.

Wie bei der verteilten Steuerung von Warteschlangen wird der Aufruf MQPUT verwendet, um eine Nachricht in eine Warteschlange eines anderen Warteschlangenmanagers zu stellen. Mit dem Aufruf MQGET können Nachrichten aus einer lokalen Warteschlange abgerufen werden.

Weitere Informationen zu Clustern finden Sie im Abschnitt [„Cluster aus Warteschlangenmanagern“](#) auf Seite 40.

### Zugehörige Konzepte

[„Komponenten der verteilten Steuerung von Warteschlangen“](#) auf Seite 58

Die hier beschriebenen Objekte werden für eine übergreifende Kommunikation benötigt.

[„Warteschlangen für nicht zustellbare Nachrichten“](#) auf Seite 61

Die Warteschlange für nicht zustellbare Nachrichten (oder Warteschlange für nicht zugestellte Nachrichten) ist die Warteschlange, an die Nachrichten gesendet werden, wenn sie nicht an ihr eigentliches Ziel weitergeleitet werden können.

[„Definitionen ferner Warteschlangen“](#) auf Seite 61

Definitionen ferner Warteschlangen sind Definitionen für Warteschlangen, die Eigentum eines anderen Warteschlangenmanagers sind.

#### „Ferne Warteschlangenmanager abrufen“ auf Seite 62

Sie verfügen möglicherweise nicht immer über einen separaten Kanal zwischen jedem Quellen- und Ziel-Warteschlangenmanager. Es gibt verschiedene Möglichkeiten, eine Verbindung zwischen den beiden herzustellen, einschließlich Multihopping, gemeinsame Nutzung von Kanälen, Nutzung verschiedener Kanäle und Clustering.

#### „Adressinformation“ auf Seite 64

Wenn eine Anwendung Nachrichten einreicht, die für einen fernen Warteschlangenmanager bestimmt sind, fügt der lokale Warteschlangenmanager einen Übertragungsheader hinzu, bevor er die Nachrichten in die Übertragungswarteschlange stellt. Dieser Header enthält die Namen der Zielwarteschlange und des Warteschlangenmanagers, also die *Adressinformationen*.

#### „Was sind Aliasnamen?“ auf Seite 64

Aliasnamen dienen dazu, eine Servicequalität für Nachrichten bereitzustellen. Mithilfe des Warteschlangenmanager-Aliasnamens kann ein Systemadministrator den Namen eines Ziel-Warteschlangenmanagers ändern, ohne dass Sie daraufhin Ihre Anwendungen ändern müssen. Außerdem erhält der Systemadministrator die Möglichkeit, die Route zu einem Zielwarteschlangenmanager zu ändern oder eine Route einzurichten, die durch mehrere andere Warteschlangenmanager führt (Multihopping). Der Aliasname für Empfangswarteschlangen für Antworten bietet eine Servicequalität für Antworten.

#### „WS-Manager-Aliasdefinitionen“ auf Seite 65

Warteschlangenmanager-Aliasnamensdefinitionen werden wirksam, wenn eine Anwendung, die eine Warteschlange zum Einreihen einer Nachricht öffnet, den Warteschlangennamen **und** den Warteschlangenmanager-Namen angibt.

#### „Aliasnamensdefinitionen für Antwortwarteschlange“ auf Seite 66

In einer Aliasnamensdefinition für eine Empfangswarteschlange für Antworten werden alternative Namen für die Antwortinformationen im Nachrichtendeskriptor angegeben. Dies bietet den Vorteil, dass Sie den Namen einer Warteschlange oder eines Warteschlangenmanagers ändern können, ohne Ihre Anwendungen ändern zu müssen.

## **Komponenten der verteilten Steuerung von Warteschlangen**

Die hier beschriebenen Objekte werden für eine übergreifende Kommunikation benötigt.

Die Komponenten der verteilten Steuerung von Warteschlangen sind:

- Nachrichtenkanäle
- Nachrichtenkanalagenten
- Übertragungswarteschlangen
- Kanalinitiatoren und Empfangsprogramme
- Kanalexitprogramme

Nachrichtenkanäle sind die Kanäle, über die Nachrichten von einem Warteschlangenmanager an einen anderen übertragen werden. Verwechseln Sie Nachrichtenkanäle nicht mit MQI-Kanälen. Es gibt zwei Typen von MQI-Kanälen, Serververbindung (SVRCONN) und Clientverbindung (CLNTCONN). Weitere Informationen finden Sie im Abschnitt MQI-Kanäle.

Bei der Definition der beiden Enden eines Nachrichtenkanals kann es sich um einen der folgenden Typen handeln:

- Sender (SDR)
- Empfänger (RCVR)
- Server (SVR)
- Requester (RQSTR)
- Clustersender (CLUSSDR)

- Clusterempfänger (CLUSRCVR)

Ein Nachrichtenkanal wird definiert, indem einer dieser Typen an einem Ende und ein kompatibler Typ am anderen Ende definiert werden. Folgende Kombinationen sind möglich:

- Sender-Empfänger
- Requester-Server
- Requester-Sender (Callback)
- Server-Empfänger
- Clustersender-Clusterempfänger

Detaillierte Anweisungen zum Erstellen eines Sender-Empfänger-Kanals finden Sie im Abschnitt [Kanäle definieren](#) (gilt nicht für z/OS). Beispiele für die erforderlichen Parameter zur Konfiguration von Sender-Empfänger-Kanälen finden Sie im betreffenden Abschnitt [Beispielkonfigurationsinformationen](#) für Ihre Plattform. Informationen zu den Parametern, die zur Definition eines Kanals eines beliebigen Typs benötigt werden, finden Sie in der Beschreibung von [DEFINE CHANNEL](#).

## Sender-Empfänger-Kanäle

Ein Sender in einem System startet den Kanal, damit er Nachrichten an das andere System senden kann. Der Sender fordert den Empfänger am anderen Ende des Kanals zum Starten auf. Der Sender sendet Nachrichten aus seiner Übertragungswarteschlange an den Empfänger. Der Empfänger stellt die Nachrichten in die Zielwarteschlange. [Abbildung 9 auf Seite 59](#) zeigt dies.

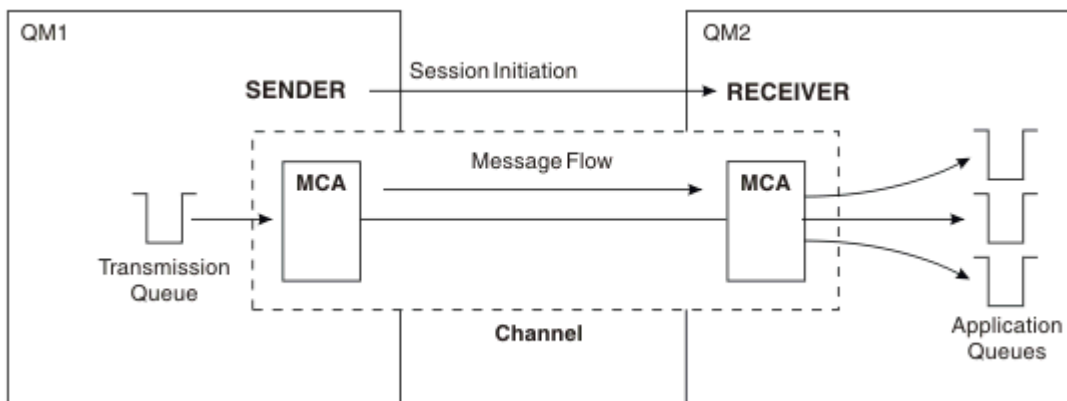


Abbildung 9. Sender-Empfänger-Kanal

## Anforderer-Server-Kanäle

Ein Requester in einem System startet den Kanal, damit er Nachrichten vom anderen System empfangen kann. Der Requester fordert den Server am anderen Ende des Kanals zum Starten auf. Der Server sendet Nachrichten aus der Übertragungswarteschlange, die in seiner Kanaldefinition angegeben ist, an den Requester.

Ein Serverkanal kann auch selbst die Kommunikation einleiten und Nachrichten an einen Requester senden. Dies gilt nur für *vollständig qualifizierte* Server, d. h. für Serverkanäle, in deren Kanaldefinition der Verbindungsname des Partners angegeben ist. Ein vollständig qualifizierter Server kann entweder von einem Requester gestartet werden oder selbst eine Kommunikation mit einem Requester einleiten.

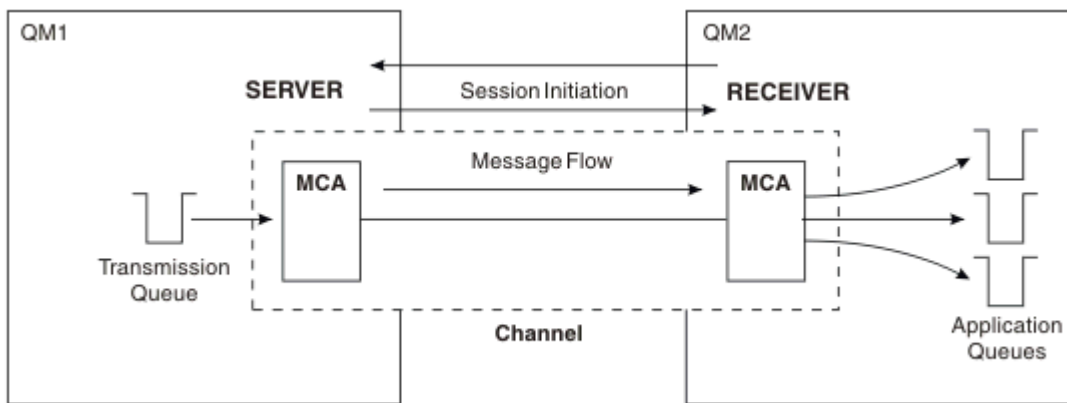


Abbildung 10. Requester-Server-Kanal

### Anforderer-Sender-Kanäle

Der Requester startet den Kanal und der Sender beendet den Aufruf. Der Sender startet die Kommunikation anschließend erneut anhand der Informationen in seiner Kanaldefinition (bekannt als *Callback*). Er sendet Nachrichten aus der Übertragungswarteschlange an den Requester.

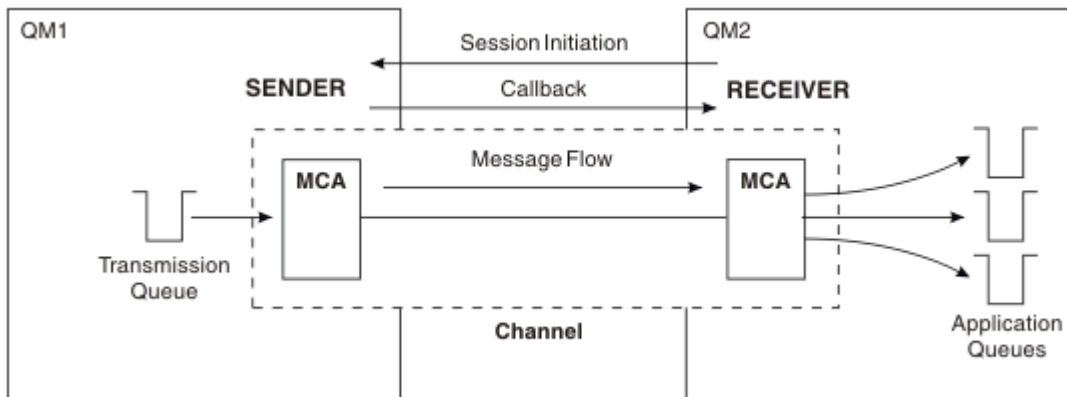


Abbildung 11. Requester-Sender-Kanal

### Server-Empfänger-Kanäle

Dies entspricht einem Sender-Empfänger-Kanal, gilt aber nur für *vollständig qualifizierte* Server, d. h. für Serverkanäle, in deren Kanaldefinition der Verbindungsname des Partners angegeben ist. Der Start des Kanals muss auf der Serverseite der Verbindung eingeleitet werden. Dieser Vorgang wird in [Abbildung 9](#) auf Seite 59 dargestellt.

### Clustersenderkanäle

In einem Cluster besitzt jeder Warteschlangenmanager einen Clustersenderkanal, über den er Clusterinformationen an eines der vollständigen Warteschlangenmanager-Repositorys senden kann. Warteschlangenmanager können über Clustersenderkanäle auch Nachrichten an andere Warteschlangenmanager senden.

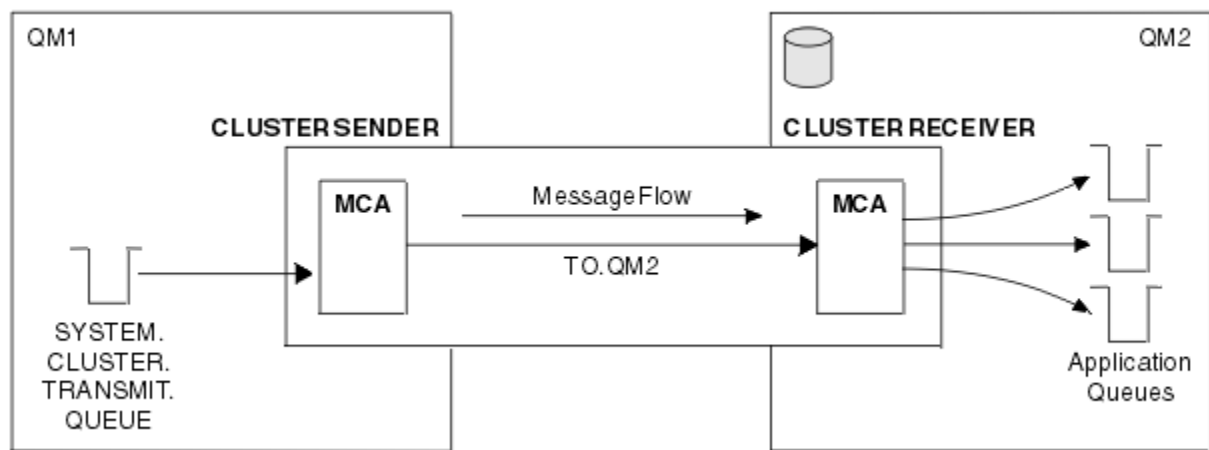


Abbildung 12. Ein Clustersenderkanal

## Clusterempfängerkanäle

In einem Cluster besitzt jeder Warteschlangenmanager einen Clusterempfängerkanal, über den er Nachrichten sowie Informationen über den Cluster empfangen kann. Dieser Vorgang wird in [Abbildung 12](#) auf Seite 61 dargestellt.

## Warteschlangen für nicht zustellbare Nachrichten

Die Warteschlange für nicht zustellbare Nachrichten (oder Warteschlange für nicht zugestellte Nachrichten) ist die Warteschlange, an die Nachrichten gesendet werden, wenn sie nicht an ihr eigentliches Ziel weitergeleitet werden können.

In diese Warteschlange werden Nachrichten eingereiht, wenn sie nicht in die Zielwarteschlange gestellt werden können. Dies kann beispielsweise dann der Fall sein, wenn die Zielwarteschlange nicht existiert oder voll ist. Warteschlangen für nicht zustellbare Nachrichten werden auch auf der Sendeseite eines Kanals bei Datenkonvertierungsfehlern verwendet.

Es wird empfohlen, für jeden Warteschlangenmanager eine Warteschlange für nicht zustellbare Nachrichten zu definieren. Wenn keine vorhanden ist und der Nachrichtenkanalagent eine Nachricht nicht einreihen kann, bleibt sie in der Übertragungswarteschlange stehen und der Kanal wird gestoppt.

Auch werden schnelle, nicht persistente Nachrichten (siehe [Schnelle, nicht persistente Nachrichten](#)), die nicht zugestellt werden können, verworfen, wenn auf dem Zielsystem keine Warteschlange für nicht zustellbare Nachrichten vorhanden ist.

Allerdings kann sich die Verwendung von Warteschlangen für nicht zustellbare Nachrichten auf die Reihenfolge auswirken, in der Nachrichten zugestellt werden, sodass Sie möglicherweise darauf verzichten möchten.

Mithilfe des Kanalattributs USEDLO können Sie festlegen, ob die Warteschlange für nicht zustellbare Nachrichten verwendet werden soll, wenn Nachrichten nicht übermittelt 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 USEDLO für verschiedene MQSC-Befehle finden Sie in [DEFINE CHANNEL](#), [DISPLAY CHANNEL](#), [ALTER CHANNEL](#) und [DISPLAY CLUSQMGR](#).

## Definitionen ferner Warteschlangen

Definitionen ferner Warteschlangen sind Definitionen für Warteschlangen, die Eigentum eines anderen Warteschlangenmanagers sind.

Abfragen können Anwendungen Nachrichten zwar nur aus lokalen Warteschlangen, aber Einreihen können Sie Nachrichten sowohl in lokale als auch in ferne Warteschlangen. Deshalb kann ein Warteschlangenma-

nager neben einer Definition für jede seiner lokalen Warteschlangen auch *Definitionen ferner Warteschlangen* besitzen. Der Vorteil von Definitionen ferner Warteschlangen besteht darin, dass sie es einer Anwendung ermöglichen, eine Nachricht in eine ferne Warteschlange zu stellen, ohne den Namen der fernen Warteschlange oder des fernen Warteschlangenmanagers bzw. den Namen der Übertragungswarteschlange kennen zu müssen. Definitionen ferner Warteschlangen bieten Standortunabhängigkeit.

Es gibt weitere Verwendungsmöglichkeiten für Definitionen ferner Warteschlangen, die später beschrieben werden.

## Ferne Warteschlangenmanager abrufen

Sie verfügen möglicherweise nicht immer über einen separaten Kanal zwischen jedem Quellen- und Ziel-Warteschlangenmanager. Es gibt verschiedene Möglichkeiten, eine Verbindung zwischen den beiden herzustellen, einschließlich Multihopping, gemeinsame Nutzung von Kanälen, Nutzung verschiedener Kanäle und Clustering.

## Multihopping

Wenn keine direkte Kommunikationsverbindung zwischen dem Quellen-Warteschlangenmanager und dem Ziel-Warteschlangenmanager besteht, ist es möglich, einen oder mehrere *zwischen geschaltete Warteschlangenmanager* auf dem Weg zum Ziel-Warteschlangenmanager zu durchlaufen. Dies ist als *Multihopping* bekannt.

Sie müssen Kanäle zwischen allen Warteschlangenmanagern und Übertragungswarteschlangen auf den zwischen geschalteten Warteschlangenmanagern definieren. Dies wird in [Abbildung 13](#) auf Seite 62 gezeigt.

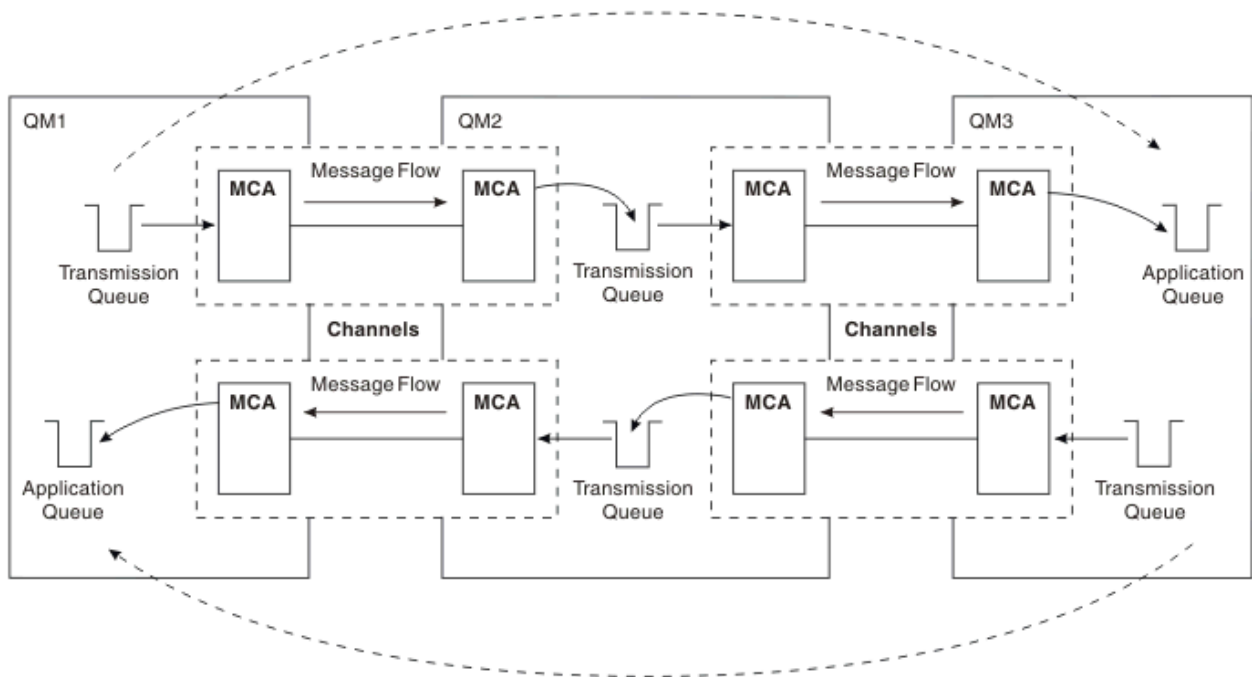


Abbildung 13. Übertragung über zwischen geschaltete Warteschlangenmanager

## Gemeinsame Nutzung von Kanälen

Als Anwendungsentwickler können Sie wählen, ob Sie Ihre Anwendungen zwingen, neben dem Namen der Warteschlange auch den Namen des fernen Warteschlangenmanagers anzugeben, oder ob Sie für jede ferne Warteschlange eine *Definition einer fernen Warteschlange* erstellen. Diese Definition enthält den Namen des fernen Warteschlangenmanagers, den Namen der Warteschlange und den Namen der Übertragungswarteschlange. In jedem Fall werden alle Nachrichten von allen Anwendungen, die War-

teschlangen an denselben fernen Standort adressieren, über dieselbe Übertragungswarteschlange gesendet. Dies wird in [Abbildung 14 auf Seite 63](#) gezeigt.

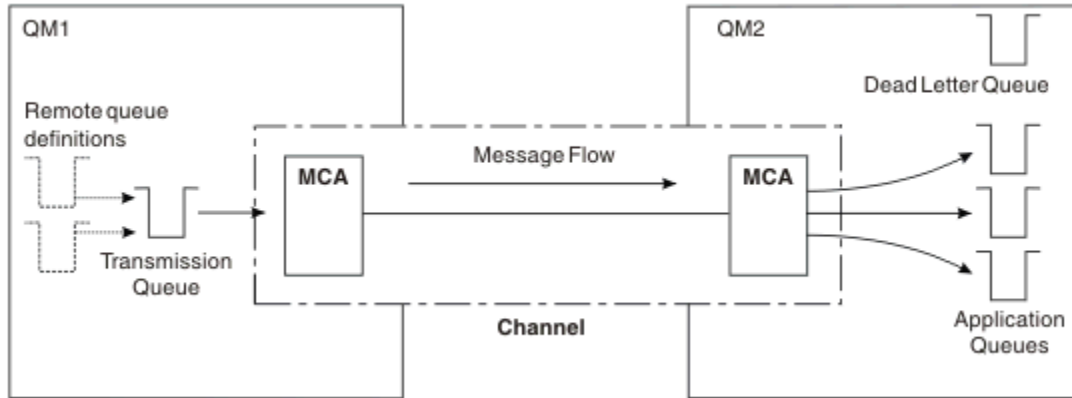


Abbildung 14. Gemeinsame Nutzung einer Übertragungswarteschlange

Abbildung 14 auf Seite 63 zeigt, dass für Nachrichten von mehreren Anwendungen an mehrere ferne Warteschlangen derselbe Kanal verwendet werden kann.

### Nutzung verschiedener Kanäle

Wenn Nachrichten unterschiedlichen Typs zwischen zwei Warteschlangenmanagern ausgetauscht werden müssen, können Sie mehrere Kanäle zwischen den beiden definieren. Es kann Situationen geben, in denen Sie alternative Kanäle benötigen, vielleicht aus Sicherheitsgründen oder um Zustellungsgeschwindigkeit durch reine Massenübertragung von Nachrichten auszugleichen.

Um einen zweiten Kanal zu konfigurieren, müssen Sie einen weiteren Kanal und eine weitere Übertragungswarteschlange definieren sowie eine Definition einer fernen Warteschlange erstellen, in der Sie den Standort und den Namen der Übertragungswarteschlange angeben. Ihre Anwendung kann anschließend jeden der beiden Kanäle verwenden, aber die Nachrichten werden weiterhin an dieselbe Zielwarteschlange zugestellt. Dies wird in [Abbildung 15 auf Seite 63](#) gezeigt.

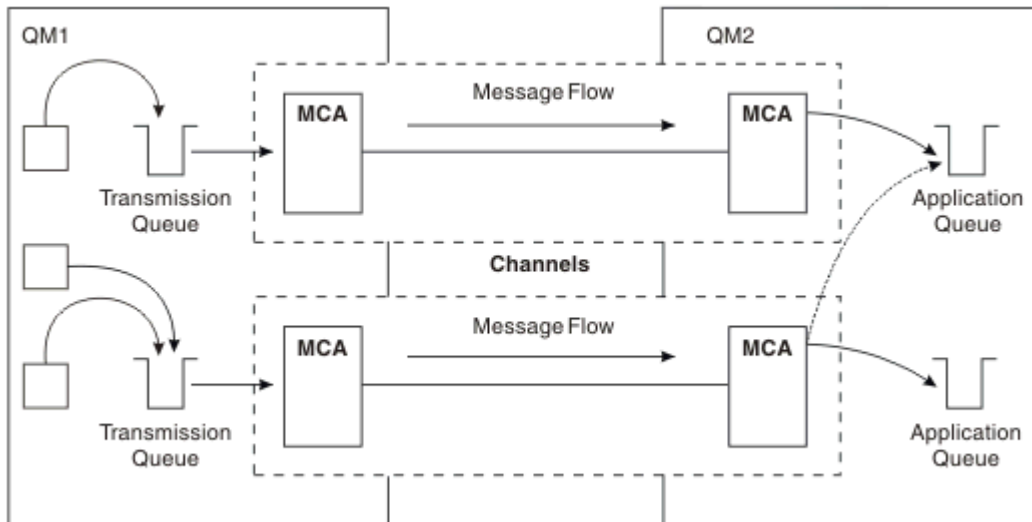


Abbildung 15. Nutzung mehrerer Kanäle

Wenn Sie eine Übertragungswarteschlange in Definitionen ferner Warteschlangen angeben, darf der Standort (d. h. der Ziel-Warteschlangenmanager) **nicht** in den Anwendungen selbst angegeben werden. Denn wenn er dort angegeben ist, verwendet der Warteschlangenmanager nicht die Definitionen ferner Warteschlangen. Definitionen ferner Warteschlangen bieten Standortunabhängigkeit. Anwendungen können Nachrichten in eine *logische* Warteschlange stellen, ohne zu wissen, wo sich die Warteschlange befindet, und Sie können die *physische* Warteschlange wechseln, ohne Ihre Anwendung ändern zu müssen.

## Clustering verwenden

Jeder Warteschlangenmanager in einem Cluster definiert einen Clusterempfängerkanal. Wenn ein Warteschlangenmanager eine Nachricht an einen anderen Warteschlangenmanager senden möchte, definiert er automatisch den zugehörigen Clustersenderkanal. Wenn es beispielsweise mehrere Instanzen einer Warteschlange in einem Cluster gibt, kann der Clustersenderkanal für jeden der Warteschlangenmanager, die die Warteschlange betreiben, definiert werden. WebSphere MQ verwendet einen Workload-Management-Algorithmus mit einer Umlaufroutine, um einen verfügbaren Warteschlangenmanager zur Weiterleitung einer Nachricht auszuwählen. Weitere Informationen finden Sie im Abschnitt „Cluster“ auf Seite 136.

## Adressinformation

Wenn eine Anwendung Nachrichten einreicht, die für einen fernen Warteschlangenmanager bestimmt sind, fügt der lokale Warteschlangenmanager einen Übertragungsheader hinzu, bevor er die Nachrichten in die Übertragungswarteschlange stellt. Dieser Header enthält die Namen der Zielwarteschlange und des Warteschlangenmanagers, also die *Adressinformationen*.

In einer Umgebung mit nur einem Warteschlangenmanager wird die Adresse einer Zielwarteschlange erstellt, wenn eine Anwendung eine Warteschlange zum Einreihen von Nachrichten öffnet. Da sich die Zielwarteschlange auf demselben Warteschlangenmanager befindet, werden keine Adressinformationen benötigt.

In einer verteilten Umgebung muss der Warteschlangenmanager nicht nur den Namen der Zielwarteschlange kennen, sondern auch deren Standort (d. h. den Namen des Warteschlangenmanagers) und die Route zu diesem Standort (d. h. die Übertragungswarteschlange). Diese Adressinformationen sind im Übertragungsheader enthalten. Der empfangende Kanal entfernt den Übertragungsheader und lokalisiert anhand der darin enthaltenen Informationen die Zielwarteschlange.

Sie können vermeiden, dass in Anwendungen der Name des Zielwarteschlangenmanagers angegeben werden muss, indem Sie eine Definition einer fernen Warteschlange verwenden. In dieser Definition werden der Name der fernen Warteschlange, der Name des fernen Warteschlangenmanagers, für den Nachrichten bestimmt sind, und der Name der Übertragungswarteschlange für den Transport der Nachrichten angegeben.

## Was sind Aliasnamen?

Aliasnamen dienen dazu, eine Servicequalität für Nachrichten bereitzustellen. Mithilfe des Warteschlangenmanager-Aliasnamens kann ein Systemadministrator den Namen eines Ziel-Warteschlangenmanagers ändern, ohne dass Sie daraufhin Ihre Anwendungen ändern müssen. Außerdem erhält der Systemadministrator die Möglichkeit, die Route zu einem Zielwarteschlangenmanager zu ändern oder eine Route einzurichten, die durch mehrere andere Warteschlangenmanager führt (Multihopping). Der Aliasname für Empfangswarteschlangen für Antworten bietet eine Servicequalität für Antworten.

Warteschlangenmanager-Aliasnamen und Aliasnamen für Empfangswarteschlangen für Antworten werden mithilfe einer Definition einer fernen Warteschlange, die ein leeres RNAME-Feld enthält, erstellt. Solche Definitionen definieren keine realen Warteschlangen; sie werden vom Warteschlangenmanager verwendet, um Namen von physischen Warteschlangen, Warteschlangenmanager-Namen und Namen von Übertragungswarteschlangen aufzulösen.

Aliasnamensdefinitionen sind durch ein leeres RNAME-Feld gekennzeichnet.

## Auflösung des Warteschlangennamens

Eine Auflösung von Warteschlangennamen findet auf jedem Warteschlangenmanager bei jedem Öffnen einer Warteschlange statt. Sie hat den Zweck, die Zielwarteschlange, den Ziel-Warteschlangenmanager (kann ein lokaler sein) und die Route zu diesem Warteschlangenmanager (kann null sein) zu ermitteln. Der aufgelöste Name besteht aus drei Teilen: den Namen des Warteschlangenmanagers, der Warteschlange und, falls es sich um einen fernen Warteschlangenmanager handelt, der Übertragungswarteschlange.

Wenn eine Definition einer fernen Warteschlange vorhanden ist, wird nicht auf Aliasnamensdefinitionen verwiesen. Der in der Anwendung angegebene Warteschlangename wird in die Namen der Zielwarte-



schlange, des fernen Warteschlangenmanagers und der Übertragungswarteschlange, die in der Definition einer fernen Warteschlange angegeben sind, aufgelöst. Ausführlichere Informationen zur Auflösung von Warteschlangennamen finden Sie im Abschnitt [Auflösung von Warteschlangennamen](#).

Wenn keine Definition einer fernen Warteschlange vorhanden und ein Warteschlangenmanager-Name angegeben ist (oder vom Namensservice aufgelöst wurde), überprüft der Warteschlangenmanager, ob für den übergebenen Warteschlangenmanager-Namen eine zugehörige Warteschlangenmanager-Aliasnamensdefinition vorhanden ist. Wenn ja, wird der Warteschlangenmanager-Name anhand der darin enthaltenen Informationen in den Namen des Zielwarteschlangenmanagers aufgelöst. Mithilfe der Warteschlangenmanager-Aliasnamensdefinition kann auch die Übertragungswarteschlange für den Zielwarteschlangenmanager ermittelt werden.

Wenn der aufgelöste Warteschlangennamen nicht der Name einer lokalen Warteschlange ist, werden sowohl der Warteschlangenmanager-Name als auch der Warteschlangennamen in den Übertragungsheader jeder Nachricht aufgenommen, die von der Anwendung in die Übertragungswarteschlange gestellt wird.

Der Name der verwendeten Übertragungswarteschlange entspricht normalerweise dem aufgelösten Warteschlangenmanager-Namen, sofern er nicht durch eine Definition einer fernen Warteschlange oder eine Warteschlangenmanager-Aliasnamensdefinition geändert wird. Wenn Sie keine Übertragungswarteschlange, aber eine Standardübertragungswarteschlange definiert haben, wird diese verwendet.

Namen von Warteschlangenmanagern, die unter z/OS ausgeführt werden, sind auf eine Länge von vier Zeichen begrenzt.

## WS-Manager-Aliasdefinitionen

Warteschlangenmanager-Aliasnamensdefinitionen werden wirksam, wenn eine Anwendung, die eine Warteschlange zum Einreihen einer Nachricht öffnet, den Warteschlangennamen **und** den Warteschlangenmanager-Namen angibt.

Warteschlangenmanager-Aliasnamensdefinitionen werden in drei Situationen verwendet:

- Beim Senden von Nachrichten, um den Warteschlangenmanager-Namen neu zuzuordnen
- Beim Senden von Nachrichten, um die Übertragungswarteschlange zu ändern oder anzugeben
- Beim Empfangen von Nachrichten, um zu ermitteln, ob der lokale Warteschlangenmanager das gewünschte Ziel für die Nachrichten ist

## Abgehende Nachrichten - Neuordnung des Warteschlangenmanager-Namens

Mithilfe von Warteschlangenmanager-Aliasnamensdefinitionen kann der in einem MQOPEN-Aufruf angegebene Warteschlangenmanager-Name neu zugeordnet werden. Ein Beispiel: In einem MQOPEN-Aufruf wird der Warteschlangennamen THISQ und der Warteschlangenmanager-Name YOURQM angegeben. Der lokale Warteschlangenmanager verfügt über folgende Warteschlangenmanager-Aliasnamensdefinition:

```
DEFINE QREMOTE (YOURQM) RQMNAME(REALQM)
```

Daraus geht hervor, dass tatsächlich der Warteschlangenmanager REALQM verwendet werden soll, wenn eine Anwendung Nachrichten an den Warteschlangenmanager YOURQM übergibt. Handelt es sich bei REALQM um den lokalen Warteschlangenmanager, stellt dieser die Nachrichten in die Warteschlange THISQ, eine lokale Warteschlange. Trägt der lokale Warteschlangenmanager nicht die Bezeichnung REALQM, wird die Nachricht an eine Übertragungswarteschlange namens REALQM weitergeleitet. Der Warteschlangenmanager ändert den Übertragungsheader von YOURQM in REALQM.

## Abgehende Nachrichten - Änderung oder Angabe der Übertragungswarteschlange

[Abbildung 16 auf Seite 66](#) zeigt ein Szenario, in dem bei Warteschlangenmanager QM1 Nachrichten ankommen, in deren Übertragungsheadern Namen von Warteschlangen auf Warteschlangenmanager QM3 angegeben sind. In diesem Szenario ist QM3 durch Multihopping über QM2 erreichbar.

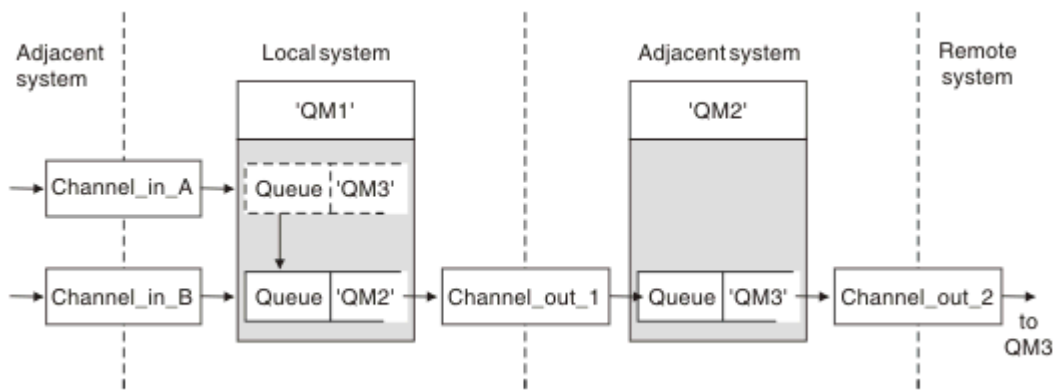


Abbildung 16. Aliasname des Warteschlangenmanagers

Alle Nachrichten für QM3 werden auf QM1 mit einem Warteschlangenmanager-Aliasnamen erfasst. Der Warteschlangenmanager-Aliasname lautet QM3 und enthält die Definition QM3 über die Übertragungswarteschlange QM2. Das folgende Beispiel zeigt die entsprechende Definition:

```
DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM2)
```

Der Warteschlangenmanager stellt die Nachrichten in Übertragungswarteschlange QM2, ändert jedoch den Header der Übertragungswarteschlange nicht, da sich der Name des Zielwarteschlangenmanagers QM3 nicht ändert.

Alle an QM1 ankommenden Nachrichten, in deren Übertragungsheader der Name einer Warteschlange von QM2 angegeben ist, werden auch in die Übertragungswarteschlange QM2 eingereiht. Auf diese Weise werden Nachrichten mit unterschiedlichen Zielen in einer gemeinsamen Übertragungswarteschlange auf einem geeigneten Nachbarsystem gesammelt, um sie dann an ihre jeweiligen Ziele zu übertragen.

## Eingehende Nachrichten - Ermittlung des Ziels

Ein empfangender Nachrichtenkanalagent öffnet die Warteschlange, die im Übertragungsheader angegeben ist. Wenn eine Warteschlangenmanager-Aliasnamensdefinition mit demselben Namen wie dem des angegebenen Warteschlangenmanagers vorhanden ist, wird der im Übertragungsheader empfangene Name des Warteschlangenmanagers aus der Definition ersetzt.

Für diesen Prozess gibt es zwei Anwendungsfälle:

- Weiterleitung von Nachrichten an einen anderen Warteschlangenmanager
- Änderung des Warteschlangenmanager-Namens in den Namen des lokalen Warteschlangenmanagers

## Aliasnamensdefinitionen für Antwortwarteschlange

In einer Aliasnamensdefinition für eine Empfangswarteschlange für Antworten werden alternative Namen für die Antwortinformationen im Nachrichtendeskriptor angegeben. Dies bietet den Vorteil, dass Sie den Namen einer Warteschlange oder eines Warteschlangenmanagers ändern können, ohne Ihre Anwendungen ändern zu müssen.

## Auflösung des Warteschlangennamens

Wenn eine Anwendung auf eine Nachricht antwortet, ermittelt sie anhand der Daten im *Nachrichtendeskriptor* der empfangenen Nachricht die Warteschlange, an die die Antwort gehen soll. Die sendende Anwendung gibt an, wohin Antworten gesendet werden sollen, indem sie entsprechende Informationen an ihre Nachrichten anhängt. Dieses Konzept muss im Rahmen des Anwendungsentwurfs koordiniert werden.

Die Auflösung des Warteschlangennamens finden auf der Sendeseite Ihrer Anwendung statt, bevor die Nachricht in eine Warteschlange gestellt wird. Dieser Fall ist eine unübliche Verwendung der Warte-

schlangennamensauflösung. Es ist die einzige Situation, in der die Namensauflösung zu einem Zeitpunkt erfolgt, an dem eine Warteschlange nicht geöffnet ist. Die Auflösung des Warteschlangennamens finden daher vor der Interaktion mit der fernen Anwendung statt, an die die Nachricht gesendet wird.

## Auflösung des Warteschlangennamens mithilfe eines Warteschlangenmanager-Aliasnamens

Normalerweise gibt eine Anwendung eine Empfangswarteschlange für Antworten an und lässt das Feld für den Namen des Managers der Empfangswarteschlange für Antworten leer. Der Warteschlangenmanager trägt seinen Namen zum Zeitpunkt des Einreihens der Nachricht ein. Dieses Verfahren funktioniert gut, es sei denn, für Antworten soll ein anderer Kanal verwendet werden, also beispielsweise ein Kanal, der die Übertragungswarteschlange QM1\_relief verwendet, und nicht der Standardrückgabekanal, der die Übertragungswarteschlange QM1 verwendet. In dieser Situation handelt es sich bei den Warteschlangenmanager-Namen, die in den Übertragungswarteschlangenheadern angegeben sind, nicht um "echte" Warteschlangenmanager-Namen, sondern um Namen, die mithilfe von Warteschlangenmanager-Aliasnamensdefinitionen neu festgelegt werden. Um Antworten über alternative Routen zurückgeben zu können, müssen auch Daten der Empfangswarteschlange für Antworten mithilfe von Aliasnamensdefinitionen für Empfangswarteschlangen für Antworten zugeordnet werden.

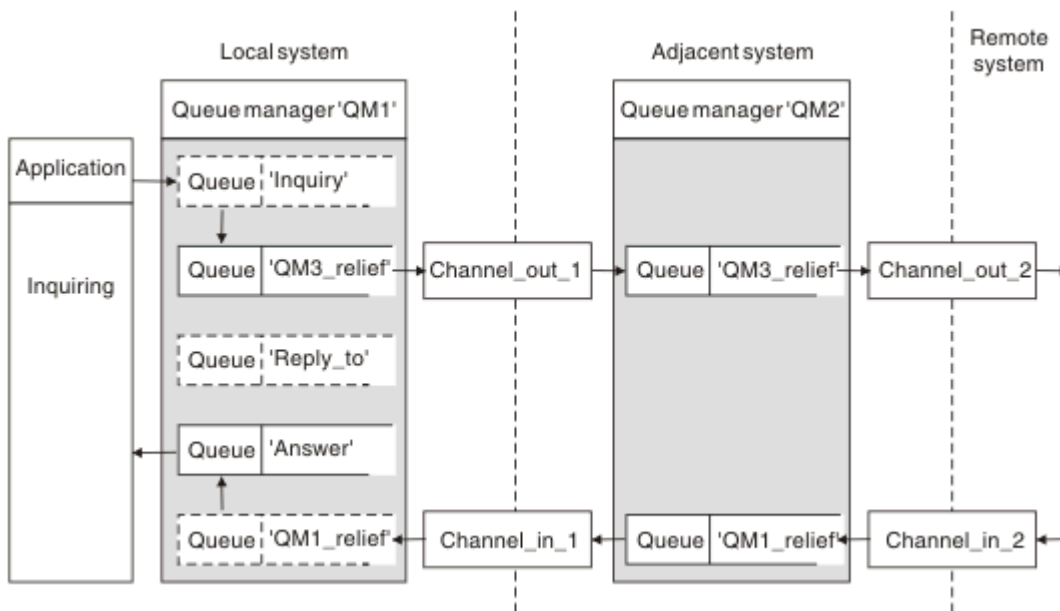


Abbildung 17. Aliasnamen für Empfangswarteschlange für Antworten zum Ändern der Antwortposition verwenden

Das Beispiel in [Abbildung 17](#) auf Seite 67 zeigt Folgendes:

1. Die Anwendung reiht mit dem Aufruf MQPUT eine Nachricht ein und gibt folgende Informationen im Nachrichtendeskriptor an:

```
ReplyToQ='Reply_to'
ReplyToQMgr=''
```

Die Angabe für ReplyToQMgr muss ein Leerzeichen sein, damit der Aliasname der Empfangswarteschlange für Antworten verwendet wird.

2. Sie erstellen eine Aliasdefinition Reply\_to für die Empfangswarteschlange für Antworten, in welcher der Name Answer und der Name des Warteschlangenmanagers QM1\_relief enthalten sind.

```
DEFINE QREMOTE ('Reply_to') RNAME ('Answer')
RQMNAME ('QM1_relief')
```

3. Die Nachrichten werden mit einem Nachrichtendeskriptor gesendet, in dem ReplyToQ= 'Answer' und ReplyToQMGr='QM1\_relief' angegeben ist.
4. Die Anwendungsspezifikation muss die Information enthalten, dass Antworten in der Warteschlange Answer und nicht in Reply\_to zu finden sind.

Um Vorbereitungen für den Empfang der Antworten zu treffen, müssen Sie den parallelen Rückgabekanal erstellen, indem Sie Folgendes definieren:

- Auf QM2 die Übertragungswarteschlange QM1\_relief

```
DEFINE QLOCAL ('QM1_relief') USAGE(XMITQ)
```

- Auf QM1 den Warteschlangenmanager-Aliasnamen QM1\_relief

```
DEFINE QREMOTE ('QM1_relief') RNAME() RQMNAME(QM1)
```

Dieser Warteschlangenmanager-Aliasname beendet die Kette aus parallelen Rückgabekanal und erfasst die Nachrichten für QM1.

Wenn Sie glauben, dass Sie dieses Verfahren irgendwann in der Zukunft einsetzen werden, stellen Sie sicher, dass der Aliasname von Anfang an in Anwendungen verwendet wird. Im Moment handelt es sich um einen normalen Aliasnamen für die Empfangswarteschlange für Antworten, der aber später in einen Warteschlangenmanager-Aliasnamen geändert werden kann.

## Name der Empfangswarteschlange für Antworten

Bei der Benennung von Empfangswarteschlangen für Antworten ist sorgfältig vorzugehen. Dass eine Anwendung den Namen einer Empfangswarteschlange für Antworten in der Nachricht angibt, hat den Grund, dass sie auf diese Weise die Warteschlange angeben kann, an die die Antworten gesendet werden sollen. Wenn Sie eine Aliasnamensdefinition für eine Empfangswarteschlange für Antworten mit diesem Namen erstellen, darf die eigentliche Empfangswarteschlange für Antworten (d. h. die Definition einer lokalen Warteschlange) nicht denselben Namen haben. Deshalb muss die Aliasnamensdefinition der Empfangswarteschlange für Antworten einen neuen Warteschlangennamen und den Warteschlangenmanager-Namen enthalten und die Anwendungsspezifikation muss die Information enthalten, dass die Antworten in dieser anderen Warteschlange eingehen werden.

Die Anwendungen müssen die Nachrichten jetzt aus einer anderen Warteschlange abrufen, als sie beim Einreihen der ursprünglichen Nachricht als Empfangswarteschlange für Antworten angegeben haben.

## Wie funktionieren Cluster?

An dieser Stelle werden das Konzept der Cluster und ihre Funktionsweise erläutert.

Ein Cluster ist ein Netz aus Warteschlangenmanagern, die logisch miteinander verbunden sind. Physisch können die Warteschlangenmanager eines Clusters getrennt sein. Sie können zum Beispiel die Filialen einer international vertretenen Geschäftskette darstellen und sich in verschiedenen Ländern befinden. Jeder Cluster innerhalb eines Unternehmens muss einen eindeutigen Namen haben.

In der Regel enthält ein Cluster Warteschlangenmanager, die auf irgendeine Weise logisch in Verbindung stehen und bestimmte Daten oder Anwendungen gemeinsam verwenden. Denkbar wäre zum Beispiel ein Warteschlangenmanager pro Unternehmensabteilung, der die für die jeweilige Abteilung typischen Daten und Anwendungen verwaltet. Diese Warteschlangenmanager könnten zu einem Cluster zusammengefasst werden, damit deren Daten der Buchhaltungsanwendung zur Verfügung stehen. Ebenso wäre ein Warteschlangenmanager pro Filiale einer Geschäftskette möglich, der die Warenbestände und andere Informationen der jeweiligen Filiale verwaltet. Wenn diese Warteschlangenmanager zu einem Cluster zusammengefasst werden, können sie alle auf die gleichen Vertriebs- und Einkaufsanwendungen zugreifen. Diese werden möglicherweise zentral bereitgestellt, zum Beispiel durch den Warteschlangenmanager der Hauptniederlassung.

Nach der Einrichtung eines Clusters können die Warteschlangenmanager ohne zusätzliche Kanaldefinitionen oder Definitionen für ferne Warteschlangen miteinander kommunizieren.

Einen Cluster können Sie aus einem bestehenden Netz aus Warteschlangenmanagern konvertieren oder bei der Einrichtung eines neuen Netzes erstellen.

Die Verbindung eines IBM WebSphere MQ-Clients mit einem Warteschlangenmanager eines Clusters erfolgt auf die gleiche Weise wie bei jedem anderen Warteschlangenmanager.

Cluster können auch für den Lastausgleich verwendet werden. Weitere Informationen finden Sie im Abschnitt [Cluster für Workload-Management verwenden](#).

## Weiterleitung von Nachrichten in einem Cluster

Wenn Ihnen IBM WebSphere MQ und die verteilte Steuerung von Warteschlangen vertraut sind, können Sie sich einen Cluster wie ein Netz aus Warteschlangen vorstellen, das von einem gewissenhaften Systemadministrator verwaltet wird. Sobald Sie eine Clusterwarteschlange definieren, erstellt der Systemadministrator automatisch auf den anderen Warteschlangenmanagern die benötigten Definitionen für die ferne Warteschlange.

Definitionen für Übertragungswarteschlangen müssen nicht erstellt werden, da IBM WebSphere MQ auf jedem Warteschlangenmanager des Clusters eine Übertragungswarteschlange bereitstellt. Über diese Übertragungswarteschlange können Nachrichten an jeden anderen Warteschlangenmanager des Clusters übertragen werden. Sie müssen sich nicht auf die Verwendung nur einer einzelnen Übertragungswarteschlange beschränken. Ein Warteschlangenmanager kann mehrere Übertragungswarteschlangen verwenden, um die Nachrichten, die an die einzelnen Warteschlangenmanager in einem Cluster gesendet werden, voneinander zu trennen. In der Regel verwendet ein Warteschlangenmanager eine einzelne Clusterübertragungswarteschlange. Durch die Änderung des Warteschlangenmanager-Attributs DEFCLXQ können Sie festlegen, dass ein Warteschlangenmanager für jeden Warteschlangenmanager in einem Cluster eine andere Clusterübertragungswarteschlange verwenden soll. Cluster-Übertragungs-WS können auch manuell definiert werden.

Alle Warteschlangenmanager, die einem Cluster beitreten, sind sich über diese Art der Zusammenarbeit einig. Sie geben Informationen über sich selbst und die von ihnen bereitgestellten Warteschlangen nach außen und erhalten ihrerseits Informationen über die anderen Mitglieder des Clusters.

Diese Informationen werden in Repositories gespeichert. Die meisten Warteschlangenmanager behalten nur die Informationen, die sie tatsächlich benötigen, also Informationen über Warteschlangen und Warteschlangenmanager, mit denen sie kommunizieren. Jeder Warteschlangenmanager speichert diese Informationen in einem Teilrepository. Einige designierte Warteschlangenmanager speichern ein vollständiges Repository mit sämtlichen Informationen zu allen Warteschlangenmanagern des Clusters.

Für den Anschluss an einen Cluster benötigt ein Warteschlangenmanager zwei Kanäle: einen Clustersenderkanal und einen Clusterempfängerkanal.

Ein Clustersenderkanal ist ein Senderkanal entsprechender Kommunikationskanal. Den Clustersenderkanal müssen Sie auf jedem Warteschlangenmanager manuell erstellen, um den Warteschlangenmanager mit einem vollständigen Repository des Clusters zu verbinden.

Ein Clusterempfängerkanal ist ein Empfängerkanal entsprechender Kommunikationskanal. Sie müssen manuell einen Clusterempfängerkanal erstellen. Dieser Kanal wird als Mechanismus für den Empfang der Clusterkommunikation durch den Warteschlangenmanager genutzt.

Alle anderen Kanäle, die zusätzlich für die Kommunikation zwischen diesem Warteschlangenmanager und den anderen Mitgliedern des Clusters erforderlich sind, werden automatisch erstellt.

Warteschlangenmanager auf Plattformen, die Cluster unterstützen, müssen jedoch kein Teil eines Clusters sein. Neben oder anstatt Clustern können Sie nach wie vor Techniken für die verteilte Steuerung von Warteschlangen verwenden.

## Beispiel für einen Cluster

[Abbildung 18 auf Seite 70](#) zeigt die Komponenten eines Clusters namens CLSTR1.

- In diesem Cluster gibt es drei Warteschlangenmanager: QM1, QM2 und QM3.
- QM1 und QM2 Host-Repositorys mit Informationen zu allen Warteschlangenmanagern und clusterbezogenen Objekten im Cluster. Sie werden als *vollständige WS-Manager-Repository-Warteschlangenmanager* bezeichnet. Die Repositorys werden in dem Diagramm durch die schattierten Zylinder dargestellt.
- QM2 und QM3 enthalten einige Warteschlangen, auf die alle anderen Warteschlangenmanager im Cluster zugreifen können. Warteschlangen, die für alle anderen WS-Manager im Cluster zugänglich sind, werden als *Clusterwarteschlangen* bezeichnet. Die Clusterwarteschlangen werden in dem Diagramm durch die schraffierten Warteschlangen dargestellt. Auf Clusterwarteschlangen kann von einer beliebigen Position im Cluster aus zugegriffen werden. Mit dem IBM WebSphere MQ-Clustering-Code wird sichergestellt, dass für Clusterwarteschlangen auf jedem Warteschlangenmanager, der auf sie verweist, entsprechende Definitionen für ferne Warteschlangen erstellt werden.

Wie bei der verteilten Steuerung von Warteschlangen verwendet eine Anwendung den Aufruf MQPUT , um eine Nachricht in eine Clusterwarteschlange auf einem beliebigen Warteschlangenmanager im Cluster einzureihen. Eine Anwendung verwendet den Aufruf MQGET , um Nachrichten aus einer Clusterwarteschlange nur auf dem Warteschlangenmanager abzurufen, auf dem sich die Warteschlange befindet.

- Jeder Warteschlangenmanager verfügt über eine manuell erstellte Definition für das Empfangsende eines Kanals mit dem Namen *cluster-name.queue-manager*, auf dem Nachrichten empfangen werden können. Auf dem Empfangswarteschlangenmanager ist *cluster-name.queue-manager* ein Cluster-Empfängerkanal. Ein Clusterempfängerkanal ist wie ein Empfängerkanal, der in der verteilten Warteschlangensteuerung verwendet wird; er empfängt Nachrichten für den Warteschlangenmanager. Darüber hinaus erhält er auch Informationen über den Cluster.

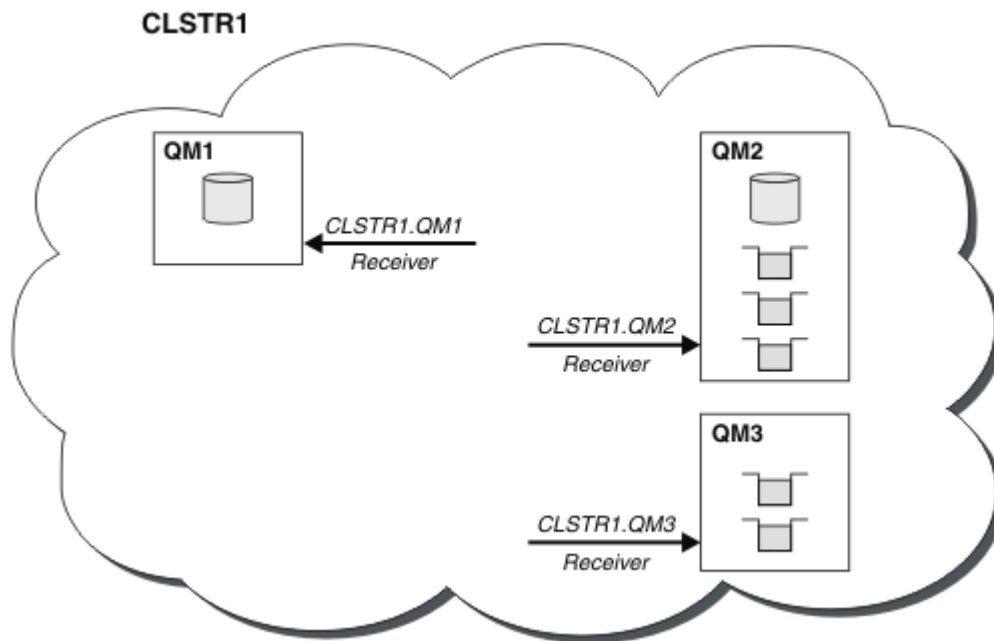


Abbildung 18. Warteschlangenmanagercluster

- In [Abbildung 19](#) auf [Seite 71](#) verfügt jeder Warteschlangenmanager auch über eine Definition für die Sendeseite des Kanals. Es stellt eine Verbindung zum Clusterempfängerkanal eines der vollständigen WS-Manager-Repositorys her. Auf dem sendenden Warteschlangenmanager ist *cluster-name.queue-manager* ein Clustersenderkanal. QM1 und QM3 verfügen über Clustersenderkanäle, die eine Verbindung zu CLSTR1.QM2 herstellen, siehe gepunktete Linie "2".

QM2 verfügt über einen Clustersenderkanal, der mit CLSTR1.QM1 verbunden ist (siehe gepunktete Linie "3"). Ein Clustersenderkanal ist vergleichbar mit einem Senderkanal, wie er bei der verteilten Steuerung von Warteschlangen verwendet wird. Über ihn werden Nachrichten an den empfangenden Warteschlangenmanager gesendet. Darüber hinaus sendet er auch Informationen zum Cluster.

Wenn sowohl das Clusterempfangende als auch das Clustersendernde eines Kanals definiert sind, wird der Kanal automatisch gestartet.

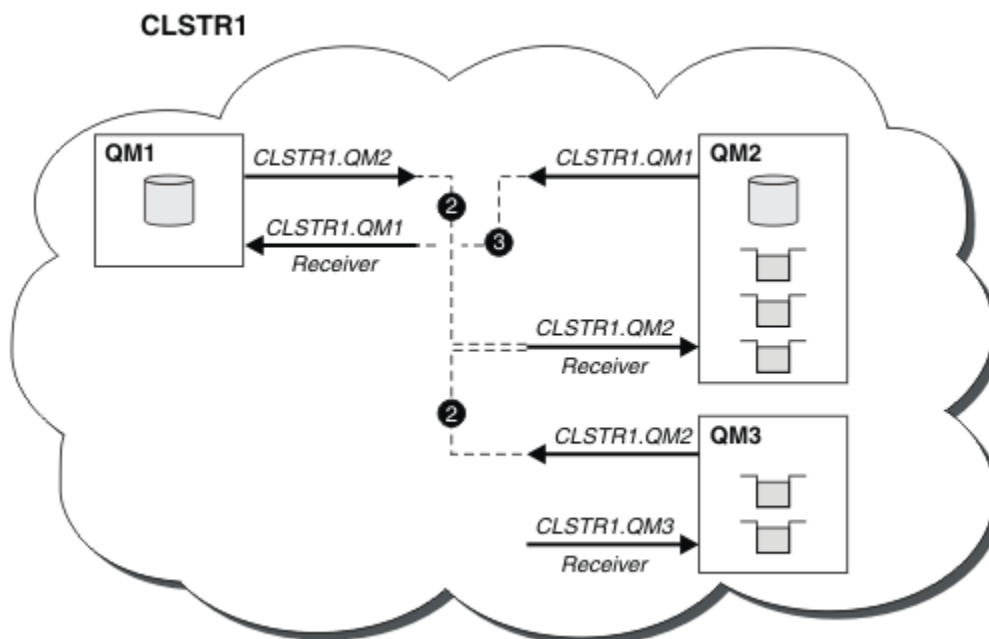


Abbildung 19. Ein Cluster von WS-Managern mit Senderkanälen

## Wie funktioniert Clustering?

Wenn Sie einen Clustersenderkanal auf dem lokalen Warteschlangenmanager definieren, wird dieser Warteschlangenmanager zu einem der vollständigen WS-Manager-Repositorys eingeführt. Der vollständige Repository-WS-Manager aktualisiert die Informationen in seinem vollständigen Repository entsprechend. Anschließend erstellt es automatisch einen Clustersenderkanal zurück zum ursprünglichen Warteschlangenmanager und sendet diese WS-Manager-Informationen über den Cluster. Daher lernt ein WS-Manager über einen Cluster und ein Cluster lernt einen Warteschlangenmanager.

Sehen Sie sich [Abbildung 18](#) auf Seite 70 noch einmal an. Angenommen, eine Anwendung, die mit dem Warteschlangenmanager QM3 verbunden ist, möchte einige Nachrichten an die Warteschlangen von QM2 senden. Wenn QM3 zum ersten Mal auf diese Warteschlangen zugreifen muss, erkennt es sie anhand eines vollständigen Repositorys. Das vollständige Repository ist in diesem Fall QM2, auf das über den Senderkanal CLSTR1.QM2 zugegriffen wird. Mit den Informationen aus dem Repository kann es automatisch ferne Definitionen für diese Warteschlangen erstellen. Wenn sich die Warteschlangen unter QM1 befinden, funktioniert dieser Mechanismus weiterhin, da QM2 ein vollständiges Repository ist. Ein vollständiges Repository verfügt über einen vollständigen Datensatz aller Objekte im Cluster. In diesem Fall würde QM3 auch automatisch einen Clustersenderkanal erstellen, der dem Clusterempfangerkanal unter QM1 entspricht, wodurch eine direkte Kommunikation zwischen den beiden Kanälen ermöglicht wird.

[Abbildung 20](#) auf Seite 72 zeigt den gleichen Cluster mit den beiden automatisch erstellten Clustersenderkanälen. Die Clustersenderkanäle werden durch die beiden gestrichelten Linien dargestellt, die mit dem Clusterempfangerkanal CLSTR1.QM3 verbunden sind. Außerdem wird die Clusterübertragungswarteschlange, SYSTEM.CLUSTER.TRANSMIT.QUEUE, angezeigt, die von QM1 zum Senden der Nachrichten verwendet wird. Alle WS-Manager im Cluster verfügen über eine Clusterübertragungswarteschlange, von der aus sie Nachrichten an einen beliebigen anderen Warteschlangenmanager in demselben Cluster senden können.

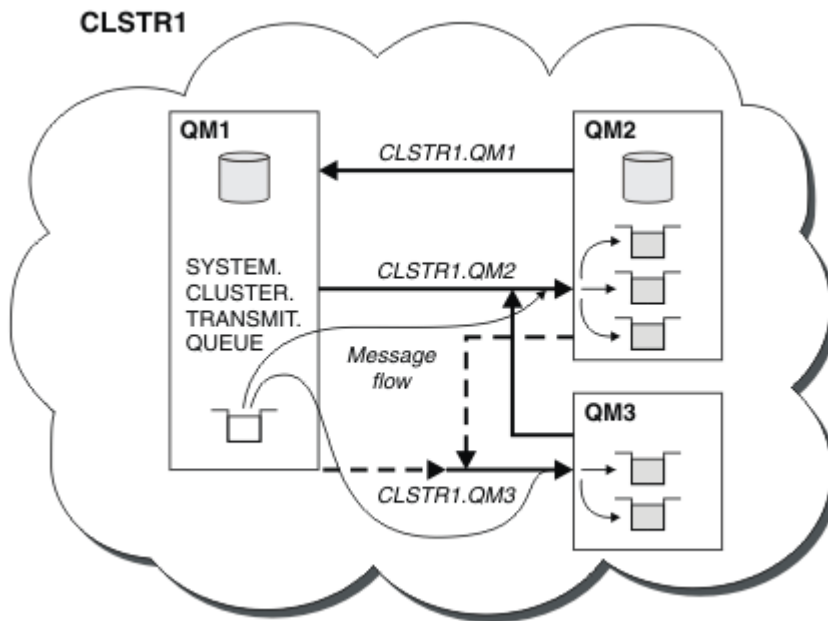


Abbildung 20. Ein Cluster von WS-Managern mit automatisch definierten Kanälen

**Anmerkung:** Andere Diagramme zeigen nur die Empfangsenden von Kanälen an, für die Sie manuelle Definitionen vornehmen. Die sendenden Enden werden weggelassen, da sie meist automatisch bei Bedarf definiert werden. Die automatische Definition der meisten Clustersenderkanäle ist entscheidend für die Funktion und die Effizienz von Clustern.

### Zugehörige Konzepte

#### Cluster

Sie können Warteschlangenmanager in einem Cluster anordnen. Jeder Warteschlangenmanager in einem Cluster kann die ihm zugeordneten Warteschlangen den anderen Warteschlangenmanagern im Cluster zugänglich machen. Ebenso kann jeder Warteschlangenmanager Nachrichten an die anderen Warteschlangenmanager im Cluster senden, ohne dass hierfür viele der Objektdefinitionen erforderlich sind, die standardmäßig bei einer verteilten Steuerung von Warteschlangen benötigt werden.

#### Vergleich von Clustering und verteilter Steuerung von Warteschlangen

#### Komponenten eines Clusters

### Zugehörige Tasks

#### WS-Manager-Cluster konfigurieren

#### Neuen Cluster einrichten

#### WebSphere MQ-Cluster verwalten

## IBM WebSphere MQ Telemetry

IBM WebSphere MQ Telemetry umfasst einen Telemetrieservice (MQXR), der Bestandteil eines Warteschlangenmanagers ist, und Telemetrieclients, die Sie selbst schreiben können. Sie können auch einen der bereitgestellten Clients verwenden. Außerdem werden Befehlszeilenschnittstellen und Explorer-Verwaltungsschnittstellen zur Verfügung gestellt. Die Telemetrie bezeichnet die Erfassung von Daten aus zahlreichen fernen Einheiten und deren Verwaltung. Mit IBM WebSphere MQ Telemetry können Sie die Datensammlung und die Einheitensteuerung in Webanwendungen integrieren.

Die MQTT-Unterstützung war in der Vergangenheit entweder im Rahmen von WebSphere Message Broker oder WebSphere MQ Version 7.0.1 verfügbar, wobei WebSphere MQ Telemetry ein eigenes Feature war. Da WebSphere MQ Telemetry eine Komponente von WebSphere MQ Version 7.1 und höher ist, besteht das Upgrade im Wesentlichen aus der Deinstallation von WebSphere MQ Telemetry Version 7.0.1 und



der Installation von WebSphere MQ Version 7.1. WebSphere MQ Telemetry kann entweder mit dem Hauptprodukt oder nach der Installation von Version 7.1 oder höher installiert werden. Informationen zur Migration finden Sie unter [Migration IBM WebSphere MQ Telemetry von Version 7.0.1 in Version 7.5](#) oder [Migration von Telemetrieanwendungen von WebSphere Message Broker Version 6 auf IBM WebSphere MQ Telemetry und WebSphere Message Broker Version 7.0](#).

IBM WebSphere MQ Telemetry enthält folgende Komponenten:

### **Telemetriedkanäle**

Mithilfe von Telemetriedkanälen können Sie die Verbindung zwischen MQTT-Clients und IBM WebSphere MQ verwalten. Telemetriedkanäle verwenden neue IBM WebSphere MQ -Objekte, z. B. `SYSTEM.MQTT.TRANSMIT.QUEUE`, für die Interaktion mit IBM WebSphere MQ.

### **Telemetrieservice (MQXR)**

MQTT-Clients verwenden den Telemetrieservice `SYSTEM.MQXR.SERVICE` für die Verbindung mit Telemetriedkanälen.

### **IBM WebSphere MQ Explorer-Unterstützung für IBM WebSphere MQ Telemetry**

IBM WebSphere MQ Telemetry kann mit IBM WebSphere MQ Explorer verwaltet werden.

### **Software-Development-Kit (SDK) für Clients**

Das Client-SDK besteht aus vier Teilen:

1. MQTT v3-Clientbibliotheken für Java SE und Java ME. Verwenden Sie die Java-Bibliotheken, um Java-Clients für Geräte zu schreiben, die Java SE oder Java ME unterstützen.
2. MQTT v3 -Bibliotheken für C. Verwenden Sie die C-Bibliotheken, um C-Clients für eine Reihe von Plattformen zu schreiben.
3. IBM WebSphere MQ Telemetry daemon for devices ist ein erweiterter Client, der in der Programmiersprache C geschrieben ist und auf einer Reihe von Plattformen ausgeführt werden kann.
4. MQTT v3-Protokoll. Das MQTT v3-Protokoll wird zur Wiederverwendung veröffentlicht und lizenziert. Verwenden Sie das Protokoll und referenzieren Sie MQTT-Clientimplementierungen, um MQTT-Clients für verschiedene Plattformen und Sprachen zu schreiben.

### **Documentation**

Die Dokumentation zu IBM WebSphere MQ Telemetry ist in der Standardprodukt dokumentation zu IBM WebSphere MQ ab Version 7.1 enthalten. Die SDK-Dokumentation für Java- und C-Clients ist in der Produktdokumentation sowie als Javadoc und im HTML-Format verfügbar.

## **Telemetriedkonzepte**

Sie sammeln Informationen aus Ihrer Umgebung und entscheiden dann über Ihre Vorgehensweise. Als Verbraucher überprüfen Sie Ihre Vorräte und entscheiden dann, welche Lebensmittel Sie kaufen. Vor der Buchung einer Anschlussverbindung möchten Sie wissen, wie lange die Reise dauert, wenn Sie jetzt aufbrechen. Sie überprüfen Ihre Symptome und entscheiden dann, ob Sie einen Arzt konsultieren. Sie prüfen, wann ein Bus ankommen wird, und entscheiden dann, ob Sie warten. Die Informationen für diese Entscheidungen stammen direkt aus Zählern und Geräten, aus schriftlichen Unterlagen oder einem Bildschirm oder von Ihnen selbst. Sie sammeln Informationen, kombinieren diese, analysieren sie und handeln auf deren Grundlage - immer und überall.

Wenn die Informationsquellen weit verstreut oder nicht zugänglich sind, wird die Erfassung präziser Informationen schwierig und aufwändig. Wenn Sie viele Änderungen vornehmen möchten oder es schwierig ist, diese Änderungen vorzunehmen, erfolgen keine Änderungen oder sie erfolgen an Stellen, an denen sie nicht so effektiv sind.

Was wäre, wenn sich der Aufwand der Erfassung von Informationen von den weit verstreuten Geräten und deren Steuerung erheblich verringern ließe, indem die Geräte mithilfe einer digitalen Technologie mit dem Internet verbunden würden? Die Informationen können mithilfe der Ressourcen des Internets und Unternehmens analysiert werden. Sie haben bessere Möglichkeiten für fundierte Entscheidungen und die Ergreifung der entsprechenden Maßnahmen.

Diese Änderungen werden durch technologische Trends und durch den umweltpolitischen und wirtschaftlichen Druck vorangetrieben:

1. Der Aufwand und die Kosten für die Verbindung und Steuerung von Sensoren und Aktuatoren können aufgrund der Standardisierung und Verbindung mit kostengünstigen digitalen Prozessoren reduziert werden.
2. Zur Verbindung von Geräten werden immer häufiger das Internet und Internettechnologien genutzt. In manchen Ländern werden für die Verbindung mit Internetanwendungen mehr Mobiltelefone als Personal Computer verwendet. Mit Sicherheit werden andere Geräte folgen.
3. Das Internet und die Internettechnologien erleichtern einer Anwendung den Abruf von Daten erheblich. Der schnelle Zugriff auf Daten fördert den Einsatz der Datenanalyse, damit Daten von Sensoren in Informationen umgewandelt werden können, die in zahlreichen anderen Lösungen hilfreich sind.
4. Der intelligente Einsatz von Ressourcen ist häufig der schnellere und kostengünstigere Weg zur Reduzierung von CO<sub>2</sub>-Emissionen und Kosten. Die Alternativen: Die Suche neuer Ressourcen oder die Entwicklung neuer Technologien zur Nutzung bestehender Ressourcen stellt vermutlich die langfristige Lösung dar. Im kurzfristigen Einsatz ist die Entwicklung neuer Technologien oder die Suche nach neuen Ressourcen oft riskanter, langsamer und teurer als die Verbesserung vorhandener Lösungen.

### Beispiel

Anhand eines Beispiels soll veranschaulicht werden, wie diese Trends neue Gelegenheiten für eine intelligente Interaktion mit der Umwelt bieten.

Das internationale Übereinkommen zum Schutz des menschlichen Lebens auf See (International Convention for the Safety of Life at Sea (SOLAS)) verlangt auf vielen Schiffen den Einsatz eines automatischen Identifikationssystems (AIS). Dieses System ist auf Handelsschiffen mit über 300 Tonnen und auf Passagierschiffen erforderlich. AIS ist in erster Linie ein System zur Kollisionsvermeidung in der Küstenschifffahrt. Es wird von Schifffahrtsbehörden zur Überwachung und Kontrolle von Küstengewässern eingesetzt.

Enthusiasten auf der ganzen Welt implementieren kostengünstige AIS-Überwachungsstationen und stellen Informationen der Küstenschifffahrt in das Internet. Andere begeisterte Anhänger schreiben Anwendungen, die die Informationen aus dem AIS mit anderen Informationen aus dem Internet kombinieren. Die Ergebnisse werden auf Websites gestellt und mithilfe von Twitter und SMS veröffentlicht.

In einer Anwendung werden beispielsweise Informationen der AIS-Stationen in der Nähe von Southampton mit Informationen zum Schiffseigner und geografischen Daten kombiniert. Die Anwendung speist die Echtzeitdaten zu Ankunfts- und Abfahrtszeiten von Fähren in Twitter ein. Regelmäßige Pendler, die die Fähren zwischen Southampton und der Isle of Wight benutzen, abonnieren den Newsfeed mittels Twitter oder SMS. Wenn der Feed eine Verspätung der Fähre anzeigt, können sich Pendler entsprechend später auf den Weg machen und die Fähre pünktlich erreichen, obwohl diese nicht zur planmäßigen Ankunftszeit anlegt.

Weitere Beispiele finden Sie im Abschnitt [„Telemetriekonzepte und -szenarios im Bereich der Überwachung und Steuerung“](#) auf Seite 74.

### Zugehörige Tasks

[WebSphere MQ Telemetry installieren](#)

[WebSphere MQ Telemetry verwalten](#)

[Migration von Telemetrieapplikationen von einer Verwendung von WebSphere Message Broker Version 6 hin zur Verwendung von WebSphere MQ Telemetry und WebSphere Message Broker Version 7.0](#)

[WebSphere MQ Telemetry von Version 7.0.1 auf Version 7.5 migrieren](#)

[Anwendungen für WebSphere MQ Telemetry entwickeln](#)

[Fehlerbehebung für WebSphere MQ Telemetry](#)

### Zugehörige Verweise

[WebSphere MQ Telemetry-Referenz](#)

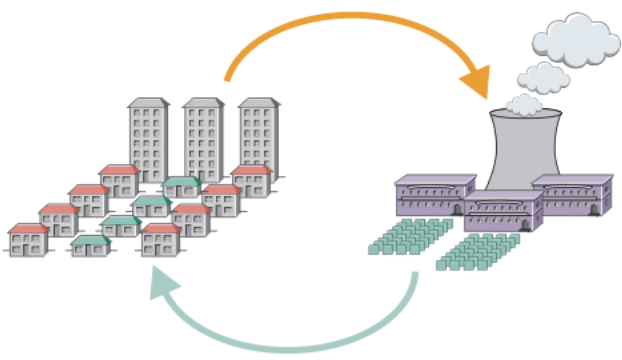
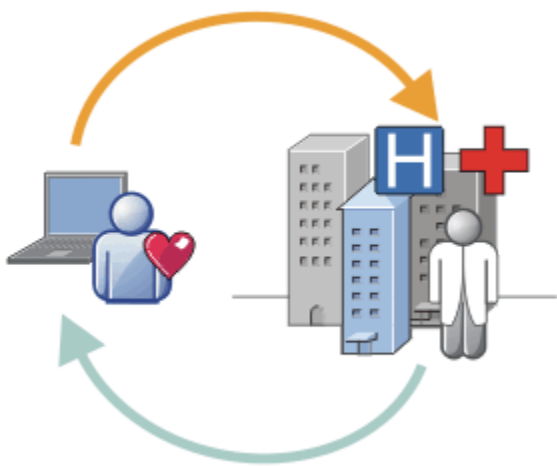
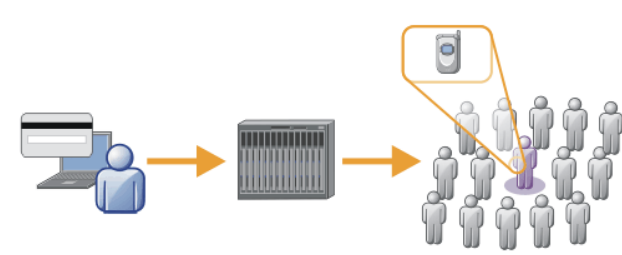
## Telemetriekonzepte und -szenarios im Bereich der Überwachung und Steuerung

Telemetrie ist die automatisierte Erkennung und Messung von Daten ferner Geräte sowie die Steuerung dieser fernen Geräte. Der Schwerpunkt liegt in der Datenübertragung von fernen Geräten zu einem

zentralen Kontrollpunkt. Die Telemetrie beinhaltet auch das Senden von Konfigurations- und Steuerinformationen an Einheiten.

IBM WebSphere MQ Telemetry verbindet kompakte Endgeräte mithilfe des MQTT-Protokolls. Die Geräte werden unter Verwendung von IBM WebSphere MQ mit anderen Anwendungen verbunden. IBM WebSphere MQ Telemetry schließt eine Lücke zwischen Geräten und dem Internet und erleichtert so die Entwicklung "intelligenter Lösungen". Smarte Lösungen ermöglichen Anwendungen, die Geräte überwachen und steuern, den Zugriff auf die Fülle von Informationen im Internet und in Unternehmensanwendungen.

Die folgenden Diagramme zeigen Beispiele zur typischen Verwendung von IBM WebSphere MQ Telemetry:

<b>Telemetry: Smart Electricity</b>	
	<ul style="list-style-type: none"> <li>• MQTT-Nachrichten mit den Energieverbrauchsdaten, die an den Service-Provider gesendet werden</li> <li>• IBM WebSphere MQ Telemetry sendet Steuerbefehle auf Grundlage der analysierten Energieverbrauchsdaten.</li> <li>• Weitere Informationen erhalten Sie in den folgenden Szenarios: „<a href="#">Telemetrieszenario: Überwachung und Steuerung des Energieverbrauchs in Privathaushalten</a>“ auf Seite 78</li> </ul>
<b>Telemetry: Smart Health Services</b>	
<ul style="list-style-type: none"> <li>• IBM WebSphere MQ Telemetry sendet Gesundheitsdaten an Ihr Krankenhaus &amp; Ihren Arzt.</li> <li>• MQTT-Rückmeldungen mit Warnungen oder Feedback können auf Basis der analysierten medizinischen Daten gesendet werden.</li> <li>• Weitere Informationen erhalten Sie in den folgenden Szenarios: „<a href="#">Telemetrieszenario: Überwachung nicht-stationärer Patienten</a>“ auf Seite 76</li> </ul>	
<b>Telemetry: One in a Crowd</b>	
	<ul style="list-style-type: none"> <li>• Eine einfache Kartentransaktion wird an den Server einer Bank gesendet.</li> <li>• IBM WebSphere MQ Telemetry ermittelt die betreffende Kontaktperson und warnt den Kunden, dass seine Karte verwendet wurde.</li> <li>• IBM WebSphere MQ Telemetry kann mithilfe einfachster Eingabeinformationen die betreffende Kontaktperson ausfindig machen.</li> </ul>

Die nachfolgenden Szenarios, die auf praktischen Beispielen beruhen, veranschaulichen einige Einsatzmöglichkeiten der Telemetrie und zeigen einige der häufig auftretenden Probleme auf, die von der Telemetrietechnologie gelöst werden müssen.

### **Zugehörige Konzepte**

„Telemetrieszenario: Überwachung nicht-stationärer Patienten“ auf Seite 76

Im Rahmen der Zusammenarbeit zwischen IBM und einem Anbieter des Gesundheitswesens bei einem Betreuungssystem für Herzpatienten kommuniziert ein implantierter Kardioverter-Defibrillator mit einem Krankenhaus. Die Daten des Patienten und implantierten Geräts werden mithilfe der RF-Telemetrie (RF = Radiofrequenz) an das MQTT-Gerät in der Wohnung des Patienten übertragen.

„Telemetrieszenario: Überwachung und Steuerung des Energieverbrauchs in Privathaushalten“ auf Seite 78

„Telemetrieszenarios: Radiofrequenzidentifikation (RFID)“ auf Seite 79

„Telemetrieszenarios: Erfassung von Umweltdaten“ auf Seite 80

Bei der Erfassung von Umweltdaten werden mithilfe der Telemetrie Informationen zum Wasserstand und zur Wasserqualität von Flüssen, zu Luftschadstoffen und zu sonstigen Umweltdaten gesammelt.

„Telemetrieszenarios: Mobile Anwendungen“ auf Seite 81

Mobile Anwendungen sind Anwendungen, die auf mobilen Endgeräten ausgeführt werden. Bei den Geräten handelt es sich entweder um generische Anwendungsplattformen oder kundenspezifische Geräte.

### ***Telemetrieszenario: Überwachung nicht-stationärer Patienten***

Im Rahmen der Zusammenarbeit zwischen IBM und einem Anbieter des Gesundheitswesens bei einem Betreuungssystem für Herzpatienten kommuniziert ein implantierter Kardioverter-Defibrillator mit einem Krankenhaus. Die Daten des Patienten und implantierten Geräts werden mithilfe der RF-Telemetrie (RF = Radiofrequenz) an das MQTT-Gerät in der Wohnung des Patienten übertragen.

Für gewöhnlich erfolgt die Übertragung nachts an einen Sender, der sich neben dem Bett befindet. Der Sender überträgt die Daten auf sichere Weise über die Telefonanlage an das Krankenhaus, wo die Daten analysiert werden.

Durch dieses System verringert sich die Anzahl der Besuche des Patienten beim Arzt. Es erkennt, wenn Probleme beim Patienten oder Gerät auftreten, und benachrichtigt bei einem Notfall den diensthabenden Arzt.

Die Zusammenarbeit zwischen IBM und dem Anbieter des Gesundheitswesens weist Merkmale auf, die in einer ganzen Reihe von Telemetrieszenarios typisch sind:

#### **Unsichtbarkeit**

Für die Nutzung des Geräts muss der Anwender lediglich Strom und eine Telefonverbindung bereitstellen und sich zu bestimmten Tageszeiten in der Nähe des Geräts befinden. Ansonsten ist keinerlei Benutzereingriff erforderlich. Der Betrieb ist zuverlässig und die Bedienung einfach.

Damit der Patient das Gerät nicht einrichten muss, wird es vom Gerätelieferanten bereits vorkonfiguriert. Der Patient muss das Gerät lediglich einstecken. Dadurch, dass die Konfiguration nicht durch den Patienten erfolgt, wird der Gerätebetrieb vereinfacht und die Gefahr einer falschen Konfiguration verringert sich.

Der MQTT-Client ist als Bestandteil des Geräts integriert. Der Geräteentwickler integriert die MQTT-Clientimplementierung im Gerät und der Entwickler bzw. Anbieter konfiguriert den MQTT-Client im Rahmen der Vorkonfiguration.

Der MQTT-Client wird als Java SE and Java ME-JAR-Datei geliefert, die der Entwickler in seine Java-Anwendung einschließt. In Umgebungen ohne Java (wie in diesem Fall) kann der Geräteentwickler einen Client unter Verwendung der veröffentlichten MQTT-Formate und des MQTT-Protokolls in einer anderen Sprache implementieren. Alternativ kann der Entwickler einen der C-Clients verwenden, die als gemeinsam genutzte Bibliotheken für Windows-, Linux- und ARM-Plattformen bereitgestellt werden.

## Ungleichmäßige Konnektivität

Die Kommunikation zwischen dem Defibrillator und dem Krankenhaus ist mit den Herausforderungen einer ungleichmäßigen Konnektivität konfrontiert. Zwei verschiedene Netze werden zur Lösung der unterschiedlichen Problemstellung der Datenerfassung beim Patienten und der Datenübertragung an das Krankenhaus genutzt. Zwischen dem Patienten und dem MQTT-Gerät wird ein RF-Nahbereichsnetz mit geringer Leistung verwendet. Der Sender verbindet sich unter Verwendung einer TCP/IP-Verbindung im virtuellen privaten Netz über eine Telefonleitung mit geringer Bandbreite mit dem Krankenhaus.

Die direkte Verbindung jedes Geräts mit einem IP-Netz gestaltet sich häufig als aufwändig. Die Verwendung von zwei Netzen, die über einen Hub miteinander verbunden sind, ist eine bewährte Lösung für dieses Problem. Das MQTT-Gerät ist ein einfacher Hub, der Informationen des Patienten speichert und diese an das Krankenhaus weiterleitet.

## Sicherheit

Der Arzt muss sich auf die Authentizität der Patientendaten verlassen können, während der Patient die Vertraulichkeit seiner Daten wünscht.

Bei einigen Szenarios reicht es aus, die Verbindung über ein virtuelles privates Netz (VPN) oder Secure Sockets Layer (SSL) zu verschlüsseln. Bei anderen Szenarios müssen die Daten auch nach ihrer Abspeicherung sicher aufbewahrt werden.

Gelegentlich lässt die Sicherheit des Telemetriegeräts zu wünschen übrig. Es könnte sich beispielsweise in einer Gemeinschaftsunterkunft befinden. Der Benutzer des Geräts muss authentifiziert werden, damit die Daten auf jeden Fall vom richtigen Patienten stammen. Das Gerät selbst kann über SSL beim Server authentifiziert werden. Die Authentifizierung in umgekehrter Richtung erfolgt ebenso.

Der Telemetriekanal zwischen dem Gerät und dem Warteschlangenmanager unterstützt JAAS (Java Authentication and Authorization Service) für die Benutzerauthentifizierung und Secure Sockets Layer (SSL) für die Verschlüsselung der Kommunikation sowie die Geräteauthentifizierung. Der Zugriff auf eine Veröffentlichung wird durch den Objektberechtigungsmanager in WebSphere MQ gesteuert.

Die Kennung, die für die Authentifizierung des Benutzers verwendet wird, kann einer anderen Kennung zugeordnet werden - beispielsweise einer allgemeinen Patientenidentität. Eine allgemeine Kennung vereinfacht die Konfiguration der Autorisierung für Veröffentlichungsthemen in WebSphere MQ.

## Konnektivität

Der Verbindungsaufbau zwischen dem MQTT-Gerät und dem Krankenhaus erfolgt über eine Anwahl, wobei eine so niedrige Bandbreite wie 300 Baud genutzt werden kann.

Damit auch bei 300 Baud ein effektiver Betrieb möglich ist, fügt das MQTT-Protokoll neben den TCP/IP-Headern lediglich eine geringe Anzahl an zusätzlichen Bytes zu einer Nachricht hinzu.

Das MQTT-Protokoll bietet eine Nachrichtenübermittlung auf Basis des "Fire-and-Forget"-Prinzips mit einer Einzelübertragung, was zu geringen Latenzzeiten führt. Es kann aber auch mehrere Übertragungen verwenden, um die Zustellung des Typs "mindestens einmal" und "genau einmal" zu ermöglichen. Dies empfiehlt sich in Fällen, bei denen eine garantierte Übermittlung wichtiger als die Reaktionszeit ist. Zur Gewährleistung der Übermittlung werden die Nachrichten im Gerät gespeichert, bis sie erfolgreich zugestellt wurden. Wenn ein Gerät drahtlos verbunden ist, ist die garantierte Übermittlung besonders sinnvoll.

## Skalierbarkeit

Telemetriegeräte werden für gewöhnlich in einer hohen Zahl implementiert - dies reicht von Zehntausenden bis hin zu Millionen.

Die Verbindung vieler Geräte mit einem System stellt hohe Anforderungen an die Lösung. Dies sind wirtschaftliche Anforderungen wie die Kosten für Geräte und die zugehörige Software und administrative Anforderungen wie die Verwaltung der Lizenzen, Geräte und Benutzer. Technische Anforderungen umfassen die Netz- und Serverauslastung.

Das Öffnen von Verbindungen belegt mehr Serverressourcen, als die Verbindungen offen zu halten. In einem Szenario wie dem vorliegenden, bei dem Telefonleitungen genutzt werden, bringen die Aus-

gaben für die Telefonverbindungen mit sich, dass die Verbindungen nur so lange offen bleiben, wie sie benötigt werden. Die Datenübertragungen erfolgen vorwiegend im Stapelbetrieb. Die Verbindungen können für die gesamte Nacht terminiert werden, um eine plötzliche Verbindungsspitze vor dem Schlafengehen zu vermeiden.

Auf dem Client wird die Skalierbarkeit von Clients durch den minimalen Aufwand für die Clientkonfiguration gefördert. Der MQTT-Client ist in das Gerät integriert. Es müssen keine Konfigurationsschritte oder Schritte zum Akzeptieren der MQTT-Clientlizenz zur Bereitstellung der Geräte für Patienten eingebunden werden.

Auf dem Server hat WebSphere MQ Telemetry ein Anfangsziel von 50.000 offenen Verbindungen je Warteschlangenmanager.

Die Verbindungen werden mithilfe von WebSphere MQ Explorer verwaltet. Der Explorer filtert die anzuzeigenden Verbindungen auf eine Anzahl, deren Verwaltung bewältigt werden kann. Durch die geeignete Wahl eines Schemas, bei dem den Clients Kennungen zugeordnet werden, können Sie Verbindungen auf Basis der geografischen Lage oder alphabetisch nach Patientennamen filtern.

## ***Telemetrieszenario: Überwachung und Steuerung des Energieverbrauchs in Privathaushalten***

Intelligente Zähler erfassen mehr Einzeldaten zum Energieverbrauch als herkömmliche Messgeräte.

Intelligente Zähler sind häufig mit einem lokalen Telemetrienetz verbunden, um einzelne Geräte in einem Privathaushalt zu überwachen und zu steuern. Einige davon sind auch remote zur Fernüberwachung und -steuerung verbunden.

Die Fernverbindung kann von einer Einzelperson, einem Elektrizitätswerk oder einer zentralen Kontrollstelle eingerichtet werden. Die ferne Kontrollstelle kann den Leistungsverbrauch ablesen und Nutzungsdaten bereitstellen. Sie kann Daten bereitstellen, die die Nutzung beeinflussen, beispielsweise fortlaufende Preis- und Wetterdaten. Durch eine Lastbegrenzung kann die gesamte Stromerzeugungseffizienz gesteigert werden.

Die Implementierung intelligenter Zähler wird immer beliebter. Die britische Regierung steht beispielsweise in Verhandlungen über die Implementierung intelligenter Zähler in allen britischen Haushalten bis zum Jahr 2020.

Die Szenarios der Messung in Privathaushalten weisen eine Reihe allgemeiner Merkmale auf:

### **Unsichtbarkeit**

Das Messgerät erfordert keinen Benutzereingriff, es sei denn, der Benutzer möchte aktiv in die Energieersparnis eingreifen, die durch die Verwendung des Messgeräts ermöglicht wird. Es darf die Zuverlässigkeit der Energieversorgung einzelner Systeme nicht beeinträchtigen.

Ein MQTT-Client kann in die Software eingebettet werden, die mit dem Messgerät implementiert wird, und erfordert keine separate Installation oder Konfiguration.

### **Ungleichmäßige Konnektivität**

Die Datenübertragung zwischen Geräten und dem intelligenten Zähler erfordert andere Konnektivitätsstandards als die Datenübertragung zwischen dem Zähler und dem Fernverbindungspunkt.

Die Verbindung vom intelligenten Zähler zu den Geräten muss hoch verfügbar sein und den Netzstandards für ein Heimnetz entsprechen.

Das ferne Netz nutzt wahrscheinlich verschiedene physische Verbindungen. Einige davon, beispielsweise Mobiltelefone, weisen hohe Übertragungskosten auf und sind möglicherweise nicht unterbrechungsfrei. Die MQTT v3-Spezifikation wurde für Fernverbindungen und Verbindungen zwischen lokalen Adaptionen und dem intelligenten Zähler konzipiert.

Für die Verbindung zwischen den Netzsteckdosen und Geräten und dem Messgerät wird ein Heimnetz wie beispielsweise Zigbee verwendet. MQTT für Sensorennetze (MQTT-S) wurde für die Arbeit mit Zigbee und anderen Netzprotokollen mit geringer Bandbreite entwickelt. MQTT-S wird von WebSphere

MQ Telemetry nicht direkt unterstützt. Es wird ein Gateway für die Verbindung von MQTT-S mit MQTT v3 benötigt.

Wie bei der Überwachung nicht stationärer Patienten erfordern auch die Lösungen für die Überwachung und Steuerung des Energieverbrauchs in Privathaushalten mehrere Netze, die unter Verwendung des intelligenten Zählers als Hub miteinander verbunden sind.

### **Sicherheit**

Im Zusammenhang mit intelligenten Zählern bestehen einige Sicherheitsprobleme. Dazu gehören der fälschungssichere Herkunftsnachweis von Transaktionen, die Autorisierung eingeleiteter Steuerungsvorgänge und der Datenschutz der Energieverbrauchswerte.

Zur Gewährleistung des Datenschutzes können die Daten, die von MQTT zwischen dem Messgerät und der fernen Kontrollstelle übertragen werden, mithilfe von Secure Sockets Layer (SSL) verschlüsselt werden. Zur Sicherstellung, dass Steuerungsvorgänge tatsächlich autorisiert sind, kann die MQTT-Verbindung zwischen dem Messgerät und der fernen Kontrollstelle gegenseitig unter Verwendung von SSL authentifiziert werden.

### **Konnektivität**

Die physische Spezifik des fernen Netzes kann beträchtlich variieren. Sie kann beispielsweise eine bestehende Breitbandverbindung oder ein Mobilnetz mit hohen Verbindungskosten und einer nicht unterbrechungsfreien Verfügbarkeit nutzen. Für hohe Kosten und intermittierende Verbindungen ist MQTT ein effizientes und zuverlässiges Protokoll (siehe „Telemetrieszenario: Überwachung nicht-stationärer Patienten“ auf Seite 76).

### **Skalierbarkeit**

Energieversorgungsunternehmen oder zentrale Kontrollstellen planen die Implementierung von letztendlich Zig-Millionen intelligenter Zähler. Anfänglich bewegt sich die Anzahl der Messgeräte je Implementierung zwischen Zehn- und Hunderttausenden. Diese Zahl ist vergleichbar mit dem anfänglichen MQTT-Ziel von 50.000 offenen Clientverbindungen je Warteschlangenmanager.

Ein kritischer Aspekt in der Architektur bei der Überwachung und Steuerung des Energieverbrauchs in Privathaushalten ist die Verwendung des intelligenten Zählers als Netzkonzentrator. Jeder Geräteadapter ist ein separater Sensor. Durch deren Verbindung mit einem lokalen Hub unter Verwendung von MQTT kann der Hub die Datenflüsse in einer einzelnen TCP/IP-Sitzung mit der zentralen Kontrollstelle konzentrieren und außerdem die Nachrichten für einen kurzen Zeitraum speichern, um Sitzungsausfallzeiten entgegen zu wirken.

Fernverbindungen müssen aus zwei Gründen in den Szenarios der Energieverbrauchssteuerung in Privathaushalten offen gehalten werden. Der erste Grund ist, dass das Öffnen von Verbindungen im Vergleich mit dem Senden von Anforderungen viel Zeit in Anspruch nimmt. Der zeitliche Aufwand für das Öffnen vieler Verbindungen zum Senden von Anforderungen zur "Lastbegrenzung" in einem kurzen Zeitraum ist zu hoch. Der zweite Grund ist, dass die Verbindung für den Empfang von Anforderungen zur Lastbegrenzung vom Energieversorgungsunternehmen zuerst vom Client geöffnet werden muss. Bei MQTT werden Verbindungen immer durch den Client aufgebaut, und für den Empfang von Anforderungen zur Lastbegrenzung vom Energieversorgungsunternehmen muss die Verbindung offen gehalten werden.

Wenn die Geschwindigkeit des Öffnens von Verbindungen kritisch ist oder der Server zeitkritische Anforderungen initialisiert, unterhält die Lösung für gewöhnlich viele offene Verbindungen.

### ***Telemetrieszenarios: Radiofrequenzidentifikation (RFID)***

RFID ist die Verwendung eines eingebetteten RFID-Tags zur Identifizierung und drahtlosen Überwachung eines Objekts. RFID-Tags können auf eine Entfernung von bis zu mehreren Metern gelesen werden und können sich außerhalb der Sichtweite der RFID-Leseinheit befinden. Passive Tags werden von einer RFID-Leseinheit aktiviert. Aktive Tags übertragen Daten ohne externe Aktivierung. Aktive Tags müssen über einen Versorgungsstromkreis verfügen. Passive Tags können einen Versorgungsstromkreis nutzen, um ihre Reichweite zu erhöhen.

RFID wird in vielen Anwendungen verwendet und die Arten der Einsatzszenarios sind sehr vielfältig. RFID-Szenarios und die Szenarios der Überwachung nicht stationärer Patienten sowie der Überwachung und Steuerung des Energieverbrauchs in Privathaushalten weisen einige Gemeinsamkeiten und Unterschiede auf.

### **Unsichtbarkeit**

In vielen Szenarios werden die RFID-Leseeinheiten in hoher Zahl implementiert und müssen ohne Benutzereingriff funktionieren. Die Leseeinheit enthält einen integrierten MQTT-Client für die Kommunikation mit einem zentralen Kontrollpunkt.

In einem Vertriebslager verwendet die Leseeinheit beispielsweise einen Bewegungssensor zur Ortung einer Palette. Sie aktiviert die RFID-Tags der Artikel auf der Palette und sendet Daten und Anforderungen an zentrale Anwendungen. Mithilfe der Daten wird die Position des Warenbestands aktualisiert. Die Anforderungen steuern, was mit der Palette im nächsten Schritt passieren soll; sie könnte beispielsweise an eine bestimmte Position verschoben werden. Fluglinien und Gepäcksysteme in Flughäfen nutzen RFID beispielsweise auf diese Weise.

In einigen RFID-Szenarios verfügt der Leser über eine Standarddatenverarbeitungsumgebung, wie z. B. Java ME. In diesen Fällen kann der MQTT-Client nach der Herstellung in einem eigenen Konfigurationsschritt implementiert werden.

### **Ungleichmäßige Konnektivität**

Die RFID-Leseeinheiten können getrennt vom lokalen Steuergerät sein, das einen MQTT-Client enthält, oder jede Leseeinheit kann über einen integrierten MQTT-Client verfügen. Für gewöhnlich wird die gewählte Topologie von geografischen oder kommunikationstechnischen Faktoren bestimmt.

### **Sicherheit**

Bei der Anlage von RFID-Tags stellen der Datenschutz und die Authentizität ein Sicherheitsproblem dar. RFID-Tags sind unauffällig und können heimlich überwacht, verfälscht oder manipuliert werden.

Die Lösung der RFID-Sicherheitsprobleme schafft Raum für die Implementierung neuer RFID-Lösungen. Auch wenn das Sicherheitsrisiko im RFID-Tag und in der lokalen Leseeinheit liegt, kann die Verarbeitung zentraler Daten Möglichkeiten für die Reaktion auf verschiedene Sicherheitsbedrohungen bieten. Eine Tag-Manipulation kann beispielsweise erkannt werden, indem der Warenbestand dynamisch mit Lieferungen und dem Versand korreliert wird.

### **Konnektivität**

In RFID-Anwendungen waren für gewöhnlich sowohl stapelorientierte Store-and-forward-Verfahren für erfasste Informationen, die aus den RFID-Leseeinheiten stammen, als auch sofortige Abfragen involviert. Im Szenario des Vertriebslagers ist die RFID-Leseeinheit ständig verbunden. Wenn ein Tag gelesen wird, wird er gemeinsam mit Informationen zur Leseeinheit veröffentlicht. Die Warehousing-Anwendung veröffentlicht die Antwort wiederum bei der Leseeinheit.

In der Warehousing-Anwendung ist das Netz normalerweise zuverlässig und die sofortigen Anforderungen verwenden unter Umständen zur Verkürzung der Latenzzeit Nachrichten des Typs "Fire and forget". Die Daten des stapelorientierten Store-and-forward-Verfahrens nutzen möglicherweise eine Nachrichtenübertragung nach dem Prinzip "genau einmal", um den Verwaltungsaufwand zu minimieren, der mit einem Datenverlust verbunden ist.

### **Skalierbarkeit**

Wenn die RFID-Anwendung sofortige Antworten benötigt, also innerhalb von ein oder zwei Sekunden, müssen die RFID-Leseeinheiten verbunden bleiben.

### ***Telemetrieszenarios: Erfassung von Umweltdaten***

Bei der Erfassung von Umweltdaten werden mithilfe der Telemetrie Informationen zum Wasserstand und zur Wasserqualität von Flüssen, zu Luftschadstoffen und zu sonstigen Umweltdaten gesammelt.

Die Sensoren befinden sich häufig an fernen Orten ohne Zugriff auf eine festnetzgebundene Kommunikation. Die drahtlose Bandbreite ist kostenintensiv und die Zuverlässigkeit kann gering sein. Für gewöhnlich sind mehrere Umgebungssensoren in einem kleinen geografischen Bereich mit einem lokalen Überwa-



chungsgerät an einem sicheren Ort verbunden. Die lokalen Verbindungen können festnetzgebunden oder drahtlos sein.

### **Unsichtbarkeit**

Die Sensoreinheiten sind in der Regel weniger zugänglich, haben eine niedrigere Leistung und werden in größerer Zahl implementiert als das zentrale Überwachungsgerät. Die Sensoren selbst sind gelegentlich "unintelligent" und das lokale Überwachungsgerät enthält Adapter für die Umwandlung und Speicherung der Sensordaten. Die Überwachungseinheit enthält wahrscheinlich einen Universalcomputer, der Java SE oder ME unterstützt. Bei der Konfiguration des MQTT-Clients spielt die Unsichtbarkeit für gewöhnlich eine untergeordnete Rolle.

### **Ungleichmäßige Konnektivität**

Das Leistungsspektrum von Sensoren sowie die Kosten und Bandbreite der Fernverbindung führen für gewöhnlich dazu, dass ein lokaler Überwachungshub mit einem zentralen Server verbunden wird.

### **Sicherheit**

Die Sicherheit spielt eine untergeordnete Rolle, es sei denn, die Lösung wird in einem militärischen Umfeld oder Verteidigungsszenario verwendet.

### **Konnektivität**

Viele Einsatzgebiete erfordern keine kontinuierliche Überwachung oder sofortige Datenverfügbarkeit. Ausnahmedaten wie beispielsweise eine Warnung vor hohem Wasserstand müssen hingegen sofort weitergeleitet werden. Die Sensordaten werden beim lokalen Überwachungsprogramm gesammelt, um Verbindungs- und Kommunikationskosten zu reduzieren, und dann in einem festgelegten Zeitrahmen über Verbindungen übertragen. Ausnahmedaten werden direkt nach ihrer Ermittlung beim Überwachungsprogramm weitergeleitet.

### **Skalierbarkeit**

Sensoren konzentrieren sich um lokale Hubs und die Sensordaten werden in Paketen gesammelt, die nach einem festgelegten Zeitplan übertragen werden. Diese Faktoren reduzieren die Arbeitslast auf dem zentralen Server, die bei der Verwendung von direkt verbundenen Sensoren bestehen würde.

## ***Telemetrieszenarios: Mobile Anwendungen***

Mobile Anwendungen sind Anwendungen, die auf mobilen Endgeräten ausgeführt werden. Bei den Geräten handelt es sich entweder um generische Anwendungsplattformen oder kundenspezifische Geräte.

Allgemeine Plattformen umfassen mobile Endgeräte wie Telefone und elektronische Organizer sowie mobile Geräte wie Notebook-Computer. Kundenspezifische Geräte verwenden spezielle Hardware, die auf bestimmte Anwendungen zugeschnitten ist. Ein Gerät für die Erfassung einer Paketzustellung "mit Unterschrift" ist ein Beispiel für ein solches kundenspezifisches mobiles Gerät. Anwendungen auf kundenspezifischen mobilen Geräten basieren häufig auf einer generischen Softwareplattform.

### **Unsichtbarkeit**

Die Implementierung kundenspezifischer mobiler Anwendungen wird verwaltet und kann die Konfiguration der MQTT-Clientanwendung beinhalten. Bei der Konfiguration des MQTT-Clients spielt die Unsichtbarkeit für gewöhnlich eine untergeordnete Rolle.

### **Ungleichmäßige Konnektivität**

Im Gegensatz zu der lokalen Hub-Topologie in den vorherigen Szenarios verbinden sich mobile Clients über Fernzugriff. Die Clientanwendungsschicht verbindet sich direkt mit einer Anwendung am zentralen Hub.

### **Sicherheit**

Da nur wenig physische Sicherheit besteht, müssen das mobile Gerät und der mobile Benutzer authentifiziert werden. Zur Überprüfung der Identität des Geräts wird SSL verwendet, während die Benutzerauthentifizierung über JAAS erfolgt.

## Konnektivität

Wenn die mobile Anwendung von einer drahtlosen Versorgung abhängt, muss sie in der Lage sein, offline zu agieren und auf effiziente Weise mit einer unterbrochenen Verbindung umgehen zu können. In dieser Umgebung besteht das Ziel darin, verbunden zu bleiben, die Anwendung muss jedoch Nachrichten im Store-and-forward-Verfahren verarbeiten können. Bei den Nachrichten handelt es sich häufig um Bestellungen oder Zustellnachweise, die von wichtigem geschäftlichen Nutzen sind. Sie müssen zuverlässig im Store-and-forward-Verfahren gespeichert und weitergeleitet werden.

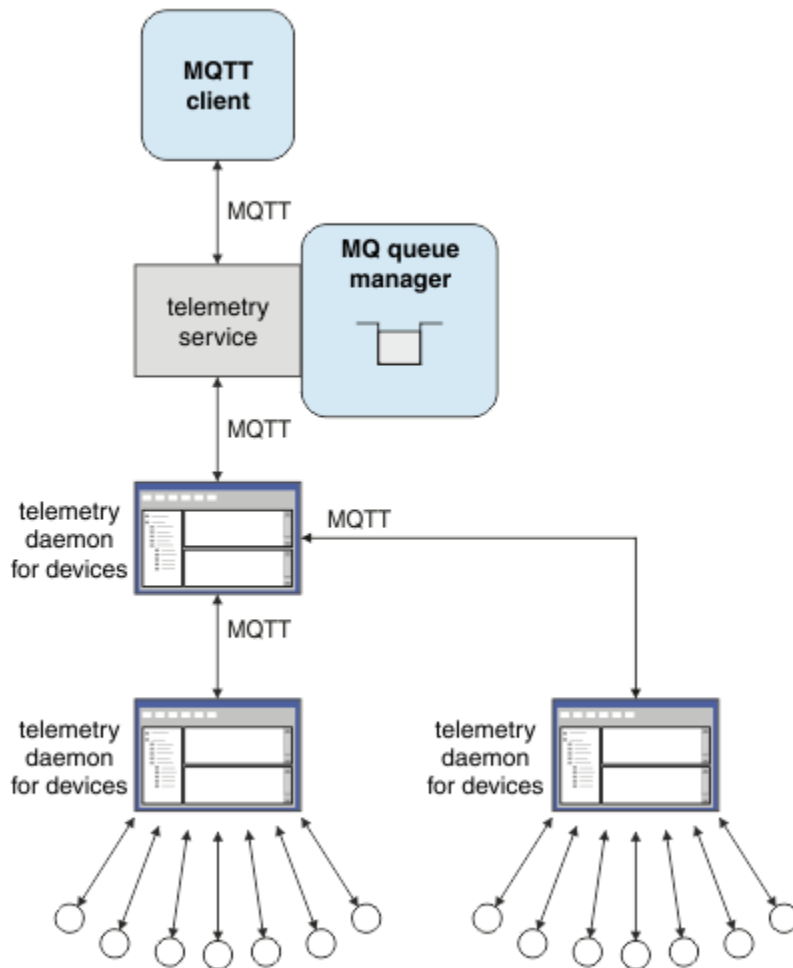
## Skalierbarkeit

Die Skalierbarkeit spielt in diesem Fall eine untergeordnete Rolle. In angepassten Szenarios mobiler Anwendungen ist es unwahrscheinlich, dass die Anzahl der Anwendungsclients Tausende oder Zehntausende von Clients überschreitet.

## Telemetriegeräte mit einem Warteschlangenmanager verbinden

Telemetriegeräte verbinden sich unter Verwendung eines MQTT v3-Clients mit einem Warteschlangenmanager. Der MQTT v3-Client nutzt TCP/IP, um eine Verbindung zu einem TCP/IP-Empfangsprogramm, dem sogenannten Telemetrieservice (MQXR), herzustellen.

Anstelle der direkten Verbindung der Telemetriegeräte mit dem Telemetrieservice können Sie die Geräte auch mit dem WebSphere MQ Telemetry-Dämon für Geräte verbinden. Der Dämon ist selbst ein MQTT v3-Client. Er bündelt die Geräteverbindungen und stellt eine einzelne Verbindung zum Telemetrieservice (MQXR) her. Sie können Dämonen in einer Hierarchie verbinden und somit die Anzahl der Geräte erhöhen, die in vielen Größenordnungen indirekt mit IBM WebSphere MQ verbunden werden können.



Der MQTT-Client baut unter Verwendung der Methode `MqttClient.connect` eine TCP/IP-Verbindung auf. Wie IBM WebSphere MQ-Clients muss auch ein MQTT-Client mit dem Warteschlangenmanager verbunden sein, damit Nachrichten gesendet und empfangen werden können. Die Verbindung wird auf dem Server unter Verwendung eines TCP/IP-Empfangsprogramms hergestellt, das im Rahmen von IBM WebSphere MQ Telemetry installiert und als Telemetrieservice (MQXR) bezeichnet wird. Auf jedem Warteschlangenmanager wird höchstens ein Telemetrieservice (MQXR) ausgeführt.

Der Telemetrieservice (MQXR) nutzt zur Zuordnung der Verbindung zu einem Telemetriekanal die Adresse des fernen Sockets, die durch jeden Client in der Methode `MqttClient.connect` festgelegt wird. Eine Socketadresse ist eine Kombination aus TCP/IP-Hostnamen und Portnummer. Mehrere Clients, die dieselbe Adresse eines fernen Sockets verwenden, werden vom Telemetrieservice (MQXR) mit demselben Telemetriekanal verbunden.

Falls sich auf einem Server mehrere Warteschlangenmanager befinden, teilen Sie die Telemetriekanäle zwischen den Warteschlangenmanagern auf. Ordnen Sie die Adressen ferner Sockets den verschiedenen Warteschlangenmanagern zu. Definieren Sie jeden Telemetriekanal mit einer eindeutigen Adresse eines fernen Sockets. Eine Socketadresse kann nicht von zwei Telemetriekanälen verwendet werden.

Wenn dieselbe Adresse eines fernen Sockets für Telemetriekanäle auf mehreren Warteschlangenmanagern konfiguriert ist, wird der erste Telemetriekanal für die Verbindung genutzt. Nachfolgende Kanäle, die sich unter derselben Adresse verbinden, schlagen fehl und führen zur Erstellung einer Datei zur Erfassung von Fehlerdaten beim ersten Auftreten (FDC-Datei).

Falls sich auf dem Server mehrere Netzadapter befinden, teilen Sie die Adressen ferner Sockets zwischen den Telemetriekanälen auf. Die Zuordnung der Socketadressen erfolgt völlig beliebig, sofern eine bestimmte Socketadresse in nur einem Telemetriekanal konfiguriert ist.

Konfigurieren Sie IBM WebSphere MQ für die Verbindung von MQTT-Clients unter Verwendung der Assistenten, die in der WebSphere MQ Telemetry-Ergänzung für IBM WebSphere MQ Explorer bereitgestellt werden. Alternativ können Sie die Telemetrie auch manuell konfigurieren. Folgen Sie dazu den Anweisungen in den Abschnitten [Warteschlangenmanager für Telemetrie unter Linux und AIX konfigurieren](#) und [Warteschlangenmanager für Telemetrie unter Windows konfigurieren](#).

## Zugehörige Verweise

[MQXR-Eigenschaften](#)

## Telemetry-Verbindungsprotokolle

WebSphere MQ Telemetry unterstützt TCP/IP IPv4 und IPv6 sowie SSL.

## Telemetrieservice (MQXR)

Beim Telemetrieservice (MQXR) handelt es sich um ein TCP/IP-Empfangsprogramm, das als IBM WebSphere MQ-Service verwaltet wird. Erstellen Sie den Service mithilfe eines IBM WebSphere MQ Explorer-Assistenten oder mit einem `runmqsc`-Befehl.

Der IBM WebSphere MQ Telemetrieservice (MQXR) wird als `SYSTEM.MQXR.SERVICE` bezeichnet.

Der Assistent **Telemetry sample configuration**, der in der IBM WebSphere MQ Telemetry-Ergänzung für IBM WebSphere MQ Explorer bereitgestellt wird, erstellt den Telemetrieservice und einen Beispieltlemetriekanal (siehe [Installation von IBM WebSphere MQ Telemetry mithilfe von IBM WebSphere MQ Explorer überprüfen](#)). Erstellen Sie die Beispielkonfiguration über die Befehlszeile (siehe [Installation von IBM WebSphere MQ Telemetry über die Befehlszeile überprüfen](#)).

Der Telemetrieservice (MQXR) wird gemeinsam mit dem Warteschlangenmanager automatisch gestartet und gestoppt. Steuern Sie den Service über den Ordner 'Services' in IBM WebSphere MQ Explorer. Damit Sie den Service sehen können, müssen Sie auf das Symbol klicken, damit SYSTEM-Objekte nicht mehr vom Explorer aus der Anzeige herausgefiltert werden.

`installMQXRService_unix.mqsc` enthält ein Beispiel der manuellen Erstellung des Service unter AIX und Linux. `installMQXRService_win.mqsc` enthält ein Beispiel der manuellen Erstellung des Service unter Windows.

## Telemetriedkanäle

Erstellen Sie Telemetriedkanäle, um Verbindungen mit unterschiedlichen Eigenschaften wie einer JAAS- oder SSL-Authentifizierung zu erstellen oder Clientgruppen zu verwalten.

Erstellen Sie Telemetriedkanäle mit dem Assistenten **New Telemetry Channel**, der in der Ergänzung IBM WebSphere MQ Telemetry für IBM WebSphere MQ Explorer bereitgestellt wird. Konfigurieren Sie einen Kanal unter Verwendung des Assistenten für die Annahme von Verbindungen aus MQTT-Clients an einem bestimmten TCP/IP-Port. Seit Version 7.1 können Sie IBM WebSphere MQ Telemetry mit dem Befehlszeilenprogramm **runmqsc** konfigurieren.

Erstellen Sie mehrere Telemetriedkanäle an verschiedenen Ports, um die Verwaltung sehr vieler Clienttransaktionen durch die Aufteilung von Clients in Gruppen zu vereinfachen. Jeder Telemetriedkanal hat einen anderen Namen.

Sie können Telemetriedkanäle mit unterschiedlichen Sicherheitsattributen konfigurieren und so verschiedene Verbindungsarten erstellen. Erstellen Sie mehrere Kanäle, damit Clientverbindungen an verschiedenen TCP/IP-Adressen akzeptiert werden können. Verschlüsseln Sie Nachrichten mithilfe von SSL und authentifizieren Sie den Telemetriedkanal und Client; siehe [SSL-Konfiguration von MQTT-Clients und Telemetriedkanälen](#). Geben Sie die Benutzer-ID zur einfacheren Berechtigung des Zugriffs auf WebSphere MQ-Objekte. Geben Sie eine JAAS-Konfiguration an, um den MQTT-Benutzer mit JAAS zu authentifizieren (siehe [MQTT-Clientidentifikation, Autorisierung und Authentifizierung](#)).

## MQTT-Protokoll

Das Protokoll 'MQ Telemetry Transport (MQTT) v3' wurde für den Nachrichtenaustausch zwischen kompakten Endgeräten mit geringer Bandbreite oder kostenintensiven Verbindungen und für eine zuverlässige Nachrichtenübertragung entwickelt. Es verwendet TCP/IP.

Das MQTT-Protokoll wird veröffentlicht; siehe [Format und Protokoll von MQ Telemetry Transport](#). Version 3 des Protokolls nutzt Publish/Subscribe und unterstützt drei Servicequalitäten: "Fire and Forget", "mindestens einmal" und "genau einmal".

Die geringe Größe der Protokollheader und der Bytefeldgruppe der Nachrichtennutzdaten sorgt für Nachrichten mit geringem Umfang. Die Header umfassen einen festgelegten Header mit 2 Bytes und bis zu 12 Bytes zusätzliche variable Header. Das Protokoll verwendet variable Header mit 12 Bytes für Subskriptionen und Verbindungen und variable Header mit nur 2 Bytes für die meisten Veröffentlichungen.

Mit drei Servicequalitäten haben Sie Spielraum für einen guten Ausgleich zwischen niedrigen Latenzzeiten und Zuverlässigkeit; siehe [Von einem MQTT-Client bereitgestellte Servicequalitäten](#). "Fire and Forget" verwendet keinen permanenten Gerätespeicher und nur eine Übertragung für das Senden oder Empfangen einer Veröffentlichung. "Mindestens einmal" und "genau einmal" erfordern einen permanenten Speicher im Gerät, um den Protokollstatus aufrechtzuerhalten und eine Nachricht bis zu ihrer Bestätigung zu speichern.

Das Protokoll gehört zu einer Familie von MQTT-Protokollen, die in anderen Produkten verwendet werden.

## MQTT-Clients

Eine MQTT-Client-App ist für die Erfassung von Daten aus dem Telemetriedgerät, die Verbindung zum Server und die Veröffentlichung der Informationen auf dem Server verantwortlich. Sie kann außerdem Themen abonnieren, Veröffentlichungen empfangen und das Telemetriedgerät steuern.

Im Gegensatz zu IBM WebSphere MQ-Clientanwendungen handelt es sich bei MQTT-Client-Apps um keine IBM WebSphere MQ-Anwendungen. Sie geben keinen Warteschlangenmanager für die Verbindungsherstellung an. Sie sind nicht auf die Verwendung bestimmter IBM WebSphere MQ-Programmierschnittstellen begrenzt. Stattdessen implementieren MQTT-Clients das MQTT version 3-Protokoll. Sie können Ihre eigene Clientbibliothek als Schnittstelle zum MQTT-Protokoll in der Programmiersprache und auf der Plattform Ihrer Wahl schreiben. Nähere Informationen hierzu finden Sie im Abschnitt [Format und Protokoll von MQ Telemetry Transport](#).

Die Entwicklung von MQTT-Client-Apps erleichtern Sie sich durch die C-, Java- und JavaScript-Clientbibliotheken, in denen das MQTT-Protokoll für eine Reihe von Plattformen enthalten ist. Links zur Client-API-Dokumentation für die MQTT-Clientbibliotheken finden Sie unter [MQTT client programming reference](#). Wenn Sie diese Bibliotheken in Ihre MQTT-Apps einbinden, können Sie auch einen MQTT-Client mit vollem Funktionsumfang schreiben, der nicht größer als 15 Codezeilen lang ist. Weitere Informationen hierzu finden Sie im Abschnitt [PubSync.java](#).

Es werden zwei Kopien der JAR-Datei `com.ibm.micro.client.mqttv3.jar` installiert. Bei einer Kopie ist die Versionsnummer im Dateinamen angegeben. Beispiel: `com.ibm.micro.client.mqttv3_3.0.2.0-20100723.jar`. Verwenden Sie in OSGi-Anwendungen die versionierte Kopie. Der Inhalt der JAR-Datei ist identisch.

Die MQTT-Client-App ist immer für den Aufbau einer Verbindung mit einem Telemetriekanal verantwortlich. Sobald die Verbindung hergestellt ist, kann entweder die MQTT-Client-App oder eine IBM WebSphere MQ-Anwendung den Nachrichtenaustausch starten.

MQTT-Client-Apps und IBM WebSphere MQ-Anwendungen veröffentlichen und abonnieren dieselbe Themengruppe. Eine IBM WebSphere MQ -Anwendung kann eine Nachricht auch direkt an eine MQTT -Client-App senden, ohne dass die Client-App zuerst eine Subskription erstellt. Nähere Informationen hierzu finden Sie im Abschnitt [Verteilte Steuerung von Warteschlangen für das Senden von Nachrichten an MQTT-Clients konfigurieren](#).

MQTT-Client-Apps werden unter Verwendung eines Telemetriekanals mit IBM WebSphere MQ verbunden. Der Telemetriekanal agiert als Brücke zwischen den unterschiedlichen Nachrichtentypen, die von MQTT und IBM WebSphere MQ verwendet werden. Er erstellt für die MQTT-Client-App Veröffentlichungen und Subskriptionen im Warteschlangenmanager. Der Telemetriekanal sendet Veröffentlichungen, die den Subskriptionen einer MQTT-Client-App entsprechen, vom Warteschlangenmanager an die MQTT-Client-App.

## **Nachricht an einen MQTT-Client senden**

WebSphere MQ-Anwendungen können Nachrichten an MQTT v3-Clients senden, indem Sie Informationen in Subskriptionen veröffentlichen, die von Clients erstellt wurden, oder indem Sie die Nachrichten direkt senden. MQTT-Clients können sich gegenseitig Nachrichten senden, indem sie Informationen in Themen veröffentlichen, die von anderen Clients abonniert wurden.

## **Ein MQTT-Client abonniert eine Veröffentlichung, die er von WebSphere MQ erhält**

Führen Sie die Task „[Nachricht über IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm veröffentlichen](#)“ auf Seite [87](#) aus, um eine Veröffentlichung aus WebSphere MQ an einen MQTT-Client zu senden.

Üblicherweise erhält ein MQTT v3-Client Nachrichten, indem er eine Subskription für ein Thema oder eine Themengruppe erstellt. In dem Beispielausschnitt [Abbildung 21](#) auf Seite [86](#) abonniert der MQTT-Client die Themenzeichenfolge "MQTT Examples". Eine in der Programmiersprache C geschriebene WebSphere MQ-Anwendung ([Abbildung 22](#) auf Seite [87](#)) veröffentlicht unter Verwendung der Themenzeichenfolge "MQTT Examples" Informationen in dem Thema. Im Codeausschnitt [Abbildung 23](#) auf Seite [87](#) empfängt der MQTT-Client die Veröffentlichung in der Callback-Methode `messageArrived`.

Weitere Informationen zur Konfiguration von WebSphere MQ zum Senden von Veröffentlichungen als Antwort auf Subskriptionen von MQTT-Clients finden Sie unter [Nachricht als Antwort auf eine MQTT-Client-Subskription veröffentlichen](#).

## **Eine WebSphere MQ-Anwendung sendet eine Nachricht direkt an einen MQTT-Client**

Führen Sie die Task „[Nachricht mit IBM WebSphere MQ Explorer an einen MQTT-Client senden](#)“ auf Seite [92](#) aus, um eine Nachricht direkt aus WebSphere MQ an einen MQTT-Client zu senden.

Eine Nachricht, die auf diese Weise an einen MQTT-Client gesendet wird, wird als "nicht erwartete Nachricht" bezeichnet. MQTT v3-Clients empfangen nicht erwartete Nachrichten als Veröffentlichungen mit einem festgelegten Themennamen. Der Telemetrieservice (MQXR) setzt den Themennamen auf den Namen der fernen Warteschlange.

Sie können nicht erwartete Nachrichten nicht an den WebSphere MQ-Dämon für Geräte senden: Der Dämon wird möglicherweise abgeschaltet, wenn er eine nicht erwartete Nachricht empfängt. Ein MQTT v3-Client kann eine nicht erwartete Nachricht weder an einen anderen MQTT v3-Client noch an eine WebSphere MQ-Warteschlange senden.

Der Abschnitt [Nachricht direkt an einen Client senden](#) enthält weitere Informationen darüber, wie Sie WebSphere MQ für das direkte Senden von Nachrichten an MQTT-Clients konfigurieren können.

## Ein MQTT-Client veröffentlicht eine Nachricht

Ein MQTT v3-Client kann eine Nachricht veröffentlichen, die von einem anderen MQTT v3-Client empfangen wird, er kann jedoch keine nicht erwartete Nachricht senden. Das Code-Snippet [Abbildung 24 auf Seite 87](#) zeigt, wie ein in Java geschriebener MQTT v3 -Client eine Nachricht veröffentlicht.

Das typische Beispiel für das Senden einer Nachricht an einen bestimmten MQTT v3-Client sieht so aus, dass jeder Client eine Subskription für seinen eigenen `ClientIdentifier`-Eintrag erstellt. Führen Sie die Task „[Nachricht in einem bestimmten MQTT v3-Client veröffentlichen](#)“ auf [Seite 94](#) aus, um eine Nachricht von einem MQTT-Client für einen anderen MQTT-Client unter Verwendung von `ClientIdentifier` als Topic-Zeichenfolge zu veröffentlichen.

## Beispielcodeausschnitte

Das Code-Snippet in [Abbildung 21 auf Seite 86](#) zeigt, wie ein in Java geschriebener MQTT-Client eine Subskription erstellt. Er benötigt außerdem eine Callback-Methode (`messageArrived`), um Veröffentlichungen für die Subskription zu empfangen. Das Code-Snippet wird aus der Task [Subskribenten für MQ Telemetry Transport mit Java erstellen](#) extrahiert.

```
String    clientId = String.format("%-23.23s",
                                System.getProperty("user.name") + "_" +
                                (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient("localhost", clientId);
String topicString = "MQTT Examples";
int      QoS = 1;
client.subscribe(topicString, QoS);
```

*Abbildung 21. MQTT v3-Client-Subskribent*

Der Codeausschnitt in [Abbildung 22 auf Seite 87](#) zeigt, wie eine WebSphere MQ-Anwendung, die in der Programmiersprache C geschrieben wurde, eine Veröffentlichung sendet. Der Codeausschnitt wird aus der Task [Veröffentlichungskomponente für ein variables Thema erstellen](#) extrahiert.

```

/* Define and set variables to defaults */
/* Omitted lines declaring variables */
char * topicName = ""
char * topicString = "MQTT Examples"
char * publication = "Hello world!";
do {
    MQCONN(qMgrName, &Hconn, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    td.ObjectType = MQOT_TOPIC; /* Object is a topic */
    td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
    strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
    td.ObjectString.VSPtr = topicString;
    td.ObjectString.VSLength = (MQLONG)strlen(topicString);
    MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    pmo.Options = MQPMO_FAIL_IF QUIESCING | MQPMO_RETAIN;
    MQPUT(Hconn, Hobj, &md, &pmo, (MQLONG)strlen(publication)+1, publication, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQCLOSE(Hconn, &Hobj, MQCO_NONE, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQDISC(&Hconn, &CompCode, &Reason);
} while (0);

```

Abbildung 22. WebSphere MQ-Veröffentlichungskomponente

Sobald die Veröffentlichung eintrifft, ruft der MQTT-Client die Methode `messageArrived` der Klasse `MqttCallback` des MQTT-Anwendungsclients auf. Das Code-Snippet wird aus der Task [Subskribenten für MQ Telemetry Transport mit Java erstellen](#) extrahiert.

```

public class Callback implements MqttCallback {
    public void messageArrived(MqttTopic topic, MqttMessage message) {
        try {
            System.out.println("Message arrived: \"" + message.toString()
                + "\" on topic \"" + topic.toString() + "\"");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
// ... Other callback methods
}

```

Abbildung 23. Methode `messageArrived`

Abbildung 24 auf Seite 87 zeigt einen MQTT v3-Client, der eine Nachricht in der Subskription veröffentlicht, die in [Abbildung 21](#) auf Seite 86 erstellt wurde. Das Code-Snippet wird aus der Task [Erste MQ Telemetry Transport-Publisher-Anwendung mit Java erstellen](#) extrahiert.

```

String address = "localhost";
String clientId = String.format("%-23.23s",
    System.getProperty("user.name") + "_" +
    (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient(address, clientId);
String topicString = "MQTT Examples";
MqttTopic topic = client.getTopic(Example.topicString);
String publication = "Hello world";
MqttMessage message = new MqttMessage(publication.getBytes());
MqttDeliveryToken token = topic.publish(message);

```

Abbildung 24. MQTT v3-Client-Veröffentlichungskomponente

## Nachricht über IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm veröffentlichen

Führen Sie die hier beschriebenen Schritte aus, um eine Nachricht mit IBM WebSphere MQ Explorer zu veröffentlichen und sie mit dem MQTT-Clientdienstprogramm zu abonnieren. In einer weiteren

Task wird veranschaulicht, wie die Standardübertragungswarteschlange nicht auf `SYSTEM.MQTT.TRANS-MIT.QUEUE` gesetzt, sondern ein Warteschlangenmanager-Aliasname konfiguriert wird.

## Vorbereitende Schritte

Bei der Task wird vorausgesetzt, dass Sie mit IBM WebSphere MQ und IBM WebSphere MQ Explorer vertraut sind und dass IBM WebSphere MQ und die Komponente IBM WebSphere MQ Telemetry installiert sind.

Der Benutzer, der die Warteschlangenmanagerressourcen für diese Task erstellt, muss über ausreichende Berechtigungen für diesen Vorgang verfügen. Zu Demonstrationszwecken wird davon ausgegangen, dass die IBM WebSphere MQ Explorer-Benutzer-ID der Gruppe `mqm` angehört.

## Informationen zu diesem Vorgang

In dieser Task erstellen Sie ein Thema in IBM WebSphere MQ und abonnieren das Thema unter Verwendung des MQTT-Clientdienstprogramms. Wenn Sie mit IBM WebSphere MQ Explorer Informationen zu dem Thema veröffentlichen, empfängt der MQTT-Client die Veröffentlichung.

## Vorgehensweise

Führen Sie eine der folgenden Tasks aus:

- Sie haben WebSphere MQ Telemetry, zwar installiert, jedoch noch nicht gestartet. Führen Sie diese Task aus: [„Task ohne definierten Telemetrieservice \(MQXR\) starten“](#) auf Seite 89.
- Sie haben WebSphere MQ Telemetry zuvor bereits ausgeführt, möchten für die Demonstration aber einen neuen Warteschlangenmanager verwenden. Führen Sie diese Task aus: [„Task ohne definierten Telemetrieservice \(MQXR\) starten“](#) auf Seite 89.
- Sie möchten die Task mit einem vorhandenen Warteschlangenmanager ausführen, auf dem keine Telemetrieressourcen definiert sind. Sie möchten den Assistenten **Beispielkonfiguration definieren** nicht ausführen.
  - a. Führen Sie eine der folgenden Tasks aus, um die Telemetrie einzurichten:
    - [Warteschlangenmanager unter Linux und AIX für die Telemetrie konfigurieren](#)
    - [Warteschlangenmanager unter Windows für die Telemetrie konfigurieren](#)
  - b. Führen Sie diese Task aus: [„Task mit einem aktiven Telemetrieservice \(MQXR\) starten“](#) auf Seite 90.
- Wenn Sie die Task mit einem vorhandenen Warteschlangenmanager ausführen möchten, auf dem bereits Telemetrieressourcen definiert sind, führen Sie diese Task aus: [„Task mit einem aktiven Telemetrieservice \(MQXR\) starten“](#) auf Seite 90.

## Nächste Schritte

Führen Sie die Task [„Nachricht mit IBM WebSphere MQ Explorer an einen MQTT-Client senden“](#) auf Seite 92 aus, um eine Nachricht direkt an das Clientdienstprogramm zu senden.

### Zugehörige Tasks

[Nachricht mit IBM WebSphere MQ Explorer an einen MQTT-Client senden](#)

Senden Sie eine Nachricht an das MQTT-Clientdienstprogramm, indem Sie eine Nachricht mithilfe von IBM WebSphere MQ Explorer in eine IBM WebSphere MQ-Warteschlange stellen. Diese Task veranschaulicht die Konfiguration einer Definition einer fernen Warteschlange zum direkten Senden einer Nachricht an einen MQTT-Client.

[Nachricht in einem bestimmten MQTT v3-Client veröffentlichen](#)

Veröffentlichen Sie eine Nachricht eines MQTT v3-Clients in einem anderen und verwenden Sie dabei `ClientIdentifier` als Themennamen und IBM WebSphere MQ als Publish/Subscribe-Broker. Wiederholen Sie die Task unter Verwendung des WebSphere MQ Telemetry-Dämons für Geräte als Publish/Subscribe-Broker.



*Task ohne definierten Telemetrieservice (MQXR) starten*

Erstellen Sie einen Warteschlangenmanager und führen Sie den Assistenten **Beispielkonfiguration definieren** aus, um Beispieltelemetriressourcen für den Warteschlangenmanager zu definieren. Veröffentlichen Sie eine Nachricht unter Verwendung von IBM WebSphere MQ Explorer und abonnieren Sie diese mit dem MQTT-Clientdienstprogramm.

## Informationen zu diesem Vorgang

Wenn Sie Beispieltelemetriressourcen mithilfe des Assistenten **Beispielkonfiguration definieren** einrichten, legt der Assistent die Berechtigungen für die Gastbenutzer-ID fest. Überlegen Sie genau, ob Sie eine derartige Berechtigung der Gastbenutzer-ID wünschen. `guest` unter Windows und `nobody` unter Linux erhalten die Berechtigung zum Veröffentlichen und Subskribieren im Stammverzeichnis der Themenstruktur und zum Einreihen von Nachrichten in `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Der Assistent setzt außerdem die Standardübertragungswarteschlange auf `SYSTEM.MQTT.TRANSMIT.QUEUE`, was zu Konflikten bei Anwendungen führen kann, die auf einem bestehenden Warteschlangenmanager ausgeführt werden. Es ist zwar aufwändig, jedoch möglich, die Telemetrie zu konfigurieren und nicht die Standardübertragungswarteschlange zu verwenden; führen Sie diese Folgetask aus: „Warteschlangenmanager-Aliasnamen verwenden“ auf Seite 91. In dieser Task erstellen Sie einen Warteschlangenmanager, um einen möglichen Konflikt mit vorhandenen Standardübertragungswarteschlangen zu vermeiden.

## Vorgehensweise

1. Erstellen Sie mit IBM WebSphere MQ Explorer einen neuen Warteschlangenmanager und starten Sie diesen.
  - a) Klicken Sie mit der rechten Maustaste auf `Queue Managers Ordner > Neu > Warteschlangenmanager ....` Geben Sie einen Warteschlangenmanagername ein und klicken Sie auf **Fertigstellen**.  
Denken Sie sich einen Warteschlangenmanagernamen aus; Beispiel: `MQTTQMGR`.
2. Erstellen und starten Sie den Telemetrieservice (MQXR) und erstellen Sie einen Beispieltelemetriekanal.
  - a) Öffnen Sie den Ordner `Queue Managers\QmgrName\Telemetry`.
  - b) Klicken Sie auf **Beispielkonfiguration definieren ... > Fertigstellen**.  
Lassen Sie das Kontrollkästchen **MQTT-Clientdienstprogramm starten** aktiviert.
3. Erstellen Sie unter Verwendung des MQTT-Clientdienstprogramms eine Subskription für `MQTT Example`.
  - a) Klicken Sie auf **Connect** (Verbinden).  
Das **Clientprotokoll** enthält jetzt das Ereignis `Connected` (Verbunden).
  - b) Geben Sie `MQTT Example` in das Feld **Subscription \ Topic** ein > **Subscribe**.  
Das **Clientprotokoll** enthält jetzt das Ereignis `Subscribed` (Verbunden).
4. Erstellen Sie `MQTTExampleTopic` in WebSphere MQ.
  - a) Klicken Sie mit der rechten Maustaste auf den Ordner `Queue Managers\QmgrName\Topics` im WebSphere MQ Explorer > **Neu > Thema**.
  - b) Geben Sie `MQTTExampleTopic` als **Name** ein und klicken Sie auf **Weiter**.
  - c) Geben Sie `MQTT Example` als **Themenzeichenfolge** > **Fertigstellenein**.
  - d) Klicken Sie auf **OK**, um das Bestätigungsfenster zu schließen.
5. Veröffentlichen Sie `Hello World!` mithilfe von IBM WebSphere MQ Explorer im Thema `MQTT Example`.
  - a) Klicken Sie im IBM WebSphere MQ Explorer auf den Ordner `Queue Managers\QmgrName\Topics`.
  - b) Klicken Sie mit der rechten Maustaste auf `MQTTExampleTopic` > **Testveröffentlichung**.

- c) Geben Sie Hello World! in das Feld **Nachrichtendaten** > **Nachricht veröffentlichen** > Wechseln Sie zum Fenster **MQTT-Clientdienstprogramm**.

Das **Clientprotokoll** enthält jetzt das Ereignis Received (Verbunden).

#### *Task mit einem aktiven Telemetrieservice (MQXR) starten*

Erstellen Sie einen Telemetriekanal und ein Thema. Berechtigen Sie den Benutzer für die Verwendung des Themas und der Übertragungswarteschlange der Telemetrie. Veröffentlichen Sie eine Nachricht unter Verwendung von IBM WebSphere MQ Explorer und abonnieren Sie diese mit dem MQTT-Clientdienstprogramm.

## Vorbereitende Schritte

In dieser Version der Task ist der Warteschlangenmanager *QmgrName* definiert und aktiv. Ein Telemetrieservice (MQXR) ist definiert und aktiv. Der Telemetrieservice (MQXR) kann manuell oder durch Ausführung des Assistenten **Beispielkonfiguration definieren** erstellt worden sein.

## Informationen zu diesem Vorgang

In dieser Task konfigurieren Sie einen vorhandenen Warteschlangenmanager für das Senden einer Veröffentlichung an das MQTT-Clientdienstprogramm.

In Schritt „1“ auf Seite 90 der Task wird außerdem die Standardübertragungswarteschlange auf SYSTEM.MQTT.TRANSMIT.QUEUE gesetzt, was zu Konflikten bei Anwendungen führen kann, die auf einem bestehenden Warteschlangenmanager ausgeführt werden. Es ist zwar aufwändig, jedoch möglich, die Telemetrie zu konfigurieren und nicht die Standardübertragungswarteschlange zu verwenden; führen Sie diese Folgetask aus: „Warteschlangenmanager-Aliasnamen verwenden“ auf Seite 91.

## Vorgehensweise

- Legen Sie SYSTEM.MQTT.TRANSMIT.QUEUE als Standardübertragungswarteschlange fest.
  - Klicken Sie mit der rechten Maustaste auf Queue Managers\*QmgrName* folder und klicken Sie auf **Eigenschaften ...**
  - Klicken Sie im Navigator auf **Kommunikation**.
  - Klicken Sie auf **Auswählen ...** > Auswählen SYSTEM.MQTT.TRANSMIT.QUEUE > **OK** > **OK**.
- Erstellen Sie den Telemetriekanal MQTTExampleChannel, um das MQTT-Clientdienstprogramm mit WebSphere MQ zu verbinden, und starten Sie das MQTT-Clientdienstprogramm.
  - Klicken Sie mit der rechten Maustaste auf den Ordner Queue Managers\*QmgrName*\Telemetry\Channels im WebSphere MQ Explorer > **Neu** > **Telemetriekanal ....**
  - Geben Sie MQTTExampleChannel in das Feld **Kanalname** ein und klicken Sie auf **Weiter** > **Weiter**.
  - Ändern Sie den Eintrag unter **Festgelegte Benutzer-ID** im Fenster für die Clientberechtigung in die Benutzer-ID, die für MQTTExample Veröffentlichungen und Subskriptionen vornehmen wird, und klicken Sie auf **Weiter**.
  - Lassen Sie **Clientdienstprogramm starten** ausgewählt > **Fertigstellen**.
- Erstellen Sie unter Verwendung des MQTT-Clientdienstprogramms eine Subskription für MQTT Example.
  - Klicken Sie auf **Connect** (Verbinden).

Das **Clientprotokoll** enthält jetzt das Ereignis Connected (Verbunden).
  - Geben Sie MQTT Example in das Feld **Subscription \ Topic** ein > **Subscribe**.

Das **Clientprotokoll** enthält jetzt das Ereignis Subscribed (Verbunden).
- Erstellen Sie MQTTExampleTopic in WebSphere MQ.
  - Klicken Sie mit der rechten Maustaste auf den Ordner Queue Managers\*QmgrName*\Topics im WebSphere MQ Explorer > **Neu** > **Thema**.

- b) Geben Sie MQTTExampleTopic als **Name** ein und klicken Sie auf **Weiter**.
  - c) Geben Sie MQTT Example als **Themenzeichenfolge** > **Fertigstellenein**.
  - d) Klicken Sie auf **OK**, um das Bestätigungsfenster zu schließen.
5. Wenn Sie möchten, dass ein Benutzer, der nicht zur Gruppe mqm gehört, Veröffentlichungen und Subskriptionen für das Thema MQTTExample vornimmt, gehen Sie wie folgt vor:
- a) Berechtigen Sie den Benutzer für die Veröffentlichung und Subskription im Zusammenhang mit dem Thema MQTTExampleTopic:

```
setmqaut -m qMgrName -t topic -n MQTTExampleTopic -p User ID -all +pub +sub
```

- b) Berechtigen Sie den Benutzer zur Einreihung einer Nachricht in die Warteschlange SYSTEM.MQTT.TRANSMIT.QUEUE:

```
setmqaut -m qMgrName -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p User ID -all +put
```

6. Veröffentlichen Sie Hello World! mithilfe von IBM WebSphere MQ Explorer im Thema MQTT Example.
- a) Klicken Sie im IBM WebSphere MQ Explorer auf den Ordner Queue Managers\qMgrName\Topics.
  - b) Klicken Sie mit der rechten Maustaste auf MQTTExampleTopic > **Testveröffentlichung**.
  - c) Geben Sie Hello World! in das Feld **Nachrichtendaten** > **Nachricht veröffentlichen** > Wechseln Sie zum Fenster **MQTT-Clientdienstprogramm**.

Das **Clientprotokoll** enthält jetzt das Ereignis Received (Verbunden).

#### *Warteschlangenmanager-Aliasnamen verwenden*

Veröffentlichen Sie eine Nachricht mit IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm, ohne die Standardübertragungswarteschlange auf SYSTEM.MQTT.TRANSMIT.QUEUE zu setzen.

Diese Task ist eine Fortsetzung der vorherigen Task; dabei wird ein Warteschlangenmanager-Aliasname verwendet, damit die Standardübertragungswarteschlange nicht auf SYSTEM.MQTT.TRANSMIT.QUEUE gesetzt werden muss.

## Vorbereitende Schritte

Die Task „[Task ohne definierten Telemetrieservice \(MQXR\) starten](#)“ auf Seite 89 oder „[Task mit einem aktiven Telemetrieservice \(MQXR\) starten](#)“ auf Seite 90 muss vollständig ausgeführt worden sein.

## Informationen zu diesem Vorgang

Wenn ein MQTT-Client eine Subskription erstellt, sendet IBM WebSphere MQ seine Antwort unter Verwendung von ClientIdentifier für den Namen des fernen Warteschlangenmanagers. In dieser Task wird MyClient als Client-ID (ClientIdentifier) verwendet.

Falls keine Übertragungswarteschlange oder kein Warteschlangenmanager-Aliasname namens MyClient vorhanden ist, wird die Antwort in die Standardübertragungswarteschlange gestellt. Durch die Festlegung der Standardübertragungswarteschlange auf SYSTEM.MQTT.TRANSMIT.QUEUE empfängt der MQTT-Client die Antwort.

Wenn Sie Warteschlangenmanager-Aliasnamen verwenden, müssen Sie die Standardübertragungswarteschlange nicht auf SYSTEM.MQTT.TRANSMIT.QUEUE setzen. Sie müssen für jede Instanz von ClientIdentifier einen Warteschlangenmanager-Aliasnamen definieren. Für gewöhnlich können Warteschlangenmanager-Aliasnamen aufgrund der Vielzahl an Clients nicht verwendet werden. Häufig ist der ClientIdentifier-Eintrag unvorhersehbar, was eine derartige Konfiguration der Telemetrie unmöglich macht.

In manchen Situationen kann es jedoch vorkommen, dass Sie die Standardübertragungswarteschlange mit einem anderen Wert als SYSTEM.MQTT.TRANSMIT.QUEUE konfigurieren müssen. Durch die unter [Vorgehensweise](#) beschriebenen Schritte wird die Standardübertragungswarteschlange nicht auf SYS-

TEM.MQTT.TRANSMIT.QUEUE gesetzt, sondern es wird ein Warteschlangenmanager-Aliasname konfiguriert.

## Vorgehensweise

1. Entfernen Sie SYSTEM.MQTT.TRANSMIT.QUEUE als Standardübertragungswarteschlange.
  - a) Klicken Sie mit der rechten Maustaste auf Queue Managers\QmgrName folder und klicken Sie auf **Eigenschaften ...**
  - b) Klicken Sie im Navigator auf **Kommunikation**.
  - c) Entfernen Sie SYSTEM.MQTT.TRANSMIT.QUEUE aus dem Feld **Standardübertragungswarteschlange** und klicken Sie auf **OK**.
2. Vergewissern Sie sich, dass mit dem MQTT-Clientdienstprogramm keine Erstellung einer Subskription mehr möglich ist:
  - a) Klicken Sie auf **Connect** (Verbinden).

Das **Clientprotokoll** enthält jetzt das Ereignis Connected (Verbunden).
  - b) Geben Sie MQTT Example in das Feld **Subscription \ Topic** ein > **Subscribe**.

Im **Clientprotokoll** werden ein Ereignis Subscribe failed und ein Ereignis Connection lost aufgezeichnet.
3. Erstellen Sie einen Warteschlangenmanager-Aliasnamen für den ClientIdentifier-Eintrag MyClient.
  - a) Klicken Sie mit der rechten Maustaste auf den Ordner Queue Managers\QmgrName\Queues > **Neu > Definition der fernen Warteschlange**.
  - b) Geben Sie der Definition den Namen MyClient > und klicken Sie auf **Weiter**.
  - c) Geben Sie MyClient in das Feld **Ferner Warteschlangenmanager** ein.
  - d) Geben Sie SYSTEM.MQTT.TRANSMIT.QUEUE im Feld **Übertragungswarteschlange** und klicken Sie auf **Fertigstellen**.
4. Stellen Sie wieder eine Verbindung zum MQTT-Clientdienstprogramm her.
  - a) Vergewissern Sie sich, dass die **Client-ID** auf MyClient gesetzt ist.
  - b) **Verbindung herstellen**

Das **Clientprotokoll** enthält jetzt das Ereignis Connected (Verbunden).
5. Erstellen Sie unter Verwendung des MQTT-Clientdienstprogramms eine Subskription für MQTT Example.
  - a) Klicken Sie auf **Connect** (Verbinden).

Das **Clientprotokoll** enthält jetzt das Ereignis Connected (Verbunden).
  - b) Geben Sie MQTT Example in das Feld **Subscription \ Topic** ein > **Subscribe**.

Das **Clientprotokoll** enthält jetzt das Ereignis Subscribed (Verbunden).
6. Veröffentlichen Sie Hello World! mithilfe von IBM WebSphere MQ Explorer im Thema MQTT Example.
  - a) Klicken Sie im IBM WebSphere MQ Explorer auf den Ordner Queue Managers\QmgrName\Topics.
  - b) Klicken Sie mit der rechten Maustaste auf MQTTExampleTopic > **Testveröffentlichung**.
  - c) Geben Sie Hello World! in das Feld **Nachrichtendaten > Nachricht veröffentlichen > Wechseln Sie zum Fenster MQTT-Clientdienstprogramm**.

Das **Clientprotokoll** enthält jetzt das Ereignis Received (Verbunden).

### **Nachricht mit IBM WebSphere MQ Explorer an einen MQTT-Client senden**

Senden Sie eine Nachricht an das MQTT-Clientdienstprogramm, indem Sie eine Nachricht mithilfe von IBM WebSphere MQ Explorer in eine IBM WebSphere MQ-Warteschlange stellen. Diese Task veranschau-

licht die Konfiguration einer Definition einer fernen Warteschlange zum direkten Senden einer Nachricht an einen MQTT-Client.

## Vorbereitende Schritte

Führen Sie den im Abschnitt „[Nachricht über IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm veröffentlichen](#)“ auf Seite 87 beschriebenen Schritt aus. Behalten Sie die Verbindung des MQTT-Clientdienstprogramms bei.

## Informationen zu diesem Vorgang

Die Task veranschaulicht das Senden einer Nachricht an einen MQTT-Client, indem sie nicht in einem Thema veröffentlicht, sondern in eine Warteschlange gestellt wird. Sie erstellen keine Subskription im Client. Schritt „[2](#)“ auf Seite 93 in der Task veranschaulicht, dass die vorherige Subskription gelöscht wurde.

## Vorgehensweise

1. Verwerfen Sie alle vorhandenen Subskriptionen, indem Sie die Verbindung des MQTT-Clientdienstprogramms trennen und wiederherstellen.

Sofern Sie die Standardeinstellungen nicht geändert haben, wird die Subskription verworfen, da das MQTT-Clientdienstprogramm mit einer bereinigten Sitzung eine Verbindung herstellt (siehe [Sitzungen bereinigen](#)).

Geben Sie zur Vereinfachung der Task Ihren eigenen ClientIdentifier-Eintrag ein, statt den generierten ClientIdentifier-Eintrag zu verwenden, der vom MQTT-Clientdienstprogramm erstellt wurde.

- a) Klicken Sie auf **Verbindung trennen**, um die Verbindung zwischen dem MQTT-Clientdienstprogramm und dem Telemetriekanal zu trennen.

Das **Clientprotokoll** enthält jetzt das Ereignis Disconnected (Verbindung getrennt).

- b) Ändern Sie die **Client-ID** in MyClient.
- c) Klicken Sie auf **Connect** (Verbinden).

Das **Clientprotokoll** enthält jetzt das Ereignis Connected (Verbindung getrennt).

2. Vergewissern Sie sich, dass das MQTT-Clientdienstprogramm keine Veröffentlichungen für das Thema MQTTExampleTopic mehr erhält.

- a) Klicken Sie im IBM WebSphere MQ Explorer auf den Ordner Queue Managers\*QmgrName*\Topics.
- b) Klicken Sie mit der rechten Maustaste auf MQTTExampleTopic > **Testveröffentlichung**.
- c) Geben Sie Hello World! in das Feld **Nachrichtendaten** > **Nachricht veröffentlichen** > Wechseln Sie zum Fenster **MQTT-Clientdienstprogramm**.

Im **Clientprotokoll** wird kein Ereignis erfasst.

3. Erstellen Sie eine Definition einer fernen Warteschlange für den Client.

Legen Sie den ClientIdentifier-Eintrag MyClient für den Namen des fernen Warteschlangenmanagers in der Definition einer fernen Warteschlange fest. Für den Namen der fernen Warteschlange kann ein beliebiger Name verwendet werden. Der Name der fernen Warteschlange wird als Themenna-  
me an einen MQTT-Client übergeben.

- a) Klicken Sie mit der rechten Maustaste auf den Ordner Queue Managers\*QmgrName*\Queues > **Neu** > **Definition der fernen Warteschlange**.
- b) Geben Sie der Definition den Namen MyClientRemoteQueue und klicken Sie auf **Weiter**.
- c) Geben Sie MQTTExampleQueue in das Feld **Ferne Warteschlange** ein.
- d) Geben Sie MyClient in das Feld **Ferner Warteschlangenmanager** ein.

- e) Geben Sie `SYSTEM.MQTT.TRANSMIT.QUEUE` im Feld **Übertragungswarteschlange** und klicken Sie auf **Fertigstellen**.
4. Reihen Sie eine Testnachricht in `MyClientRemoteQueue` ein.
- Klicken Sie mit der rechten Maustaste auf **MyClientRemoteQueue > Testnachricht einreihen...**
  - Geben Sie `Hello queue!` in das Feld für Nachrichtendaten ein > **Nachricht einreihen > Schließen**

Das **Clientprotokoll** enthält jetzt das Ereignis `Received` (Empfangen).

5. Entfernen Sie `SYSTEM.MQTT.TRANSMIT.QUEUE` als Standardübertragungswarteschlange.
- Klicken Sie mit der rechten Maustaste auf `Queue Managers\QmgrName` folder und klicken Sie auf **Eigenschaften ...**
  - Klicken Sie im Navigator auf **Kommunikation**.
  - Entfernen Sie `SYSTEM.MQTT.TRANSMIT.QUEUE` aus dem Feld **Standardübertragungswarteschlange** und klicken Sie auf **OK**.
6. Wiederholen Sie den Schritt „4“ auf Seite 94.

`MyClientRemoteQueue` ist eine Definition einer fernen Warteschlange, in der die Übertragungswarteschlange explizit genannt wird. Sie müssen keine Standardübertragungswarteschlange definieren, um eine Nachricht an `MyClient` zu senden.

## Nächste Schritte

Jetzt, da die Standardübertragungswarteschlange nicht mehr auf `SYSTEM.MQTT.TRANSMIT.QUEUE` gesetzt ist, kann das MQTT-Clientdienstprogramm keine neue Subskription erstellen, es sei denn, ein Warteschlangenmanager-Aliasname ist für den `ClientIdentifier`-Eintrag `MyClient` definiert. Setzen Sie die Standardübertragungswarteschlange wieder auf `SYSTEM.MQTT.TRANSMIT.QUEUE`.

### Zugehörige Tasks

Nachricht über IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm veröffentlichen

Führen Sie die hier beschriebenen Schritte aus, um eine Nachricht mit IBM WebSphere MQ Explorer zu veröffentlichen und sie mit dem MQTT-Clientdienstprogramm zu abonnieren. In einer weiteren Task wird veranschaulicht, wie die Standardübertragungswarteschlange nicht auf `SYSTEM.MQTT.TRANSMIT.QUEUE` gesetzt, sondern ein Warteschlangenmanager-Aliasname konfiguriert wird.

Nachricht in einem bestimmten MQTT v3-Client veröffentlichen

Veröffentlichen Sie eine Nachricht eines MQTT v3-Clients in einem anderen und verwenden Sie dabei `ClientIdentifier` als Themennamen und IBM WebSphere MQ als Publish/Subscribe-Broker. Wiederholen Sie die Task unter Verwendung des WebSphere MQ Telemetry-Dämons für Geräte als Publish/Subscribe-Broker.

### ***Nachricht in einem bestimmten MQTT v3-Client veröffentlichen***

Veröffentlichen Sie eine Nachricht eines MQTT v3-Clients in einem anderen und verwenden Sie dabei `ClientIdentifier` als Themennamen und IBM WebSphere MQ als Publish/Subscribe-Broker. Wiederholen Sie die Task unter Verwendung des WebSphere MQ Telemetry-Dämons für Geräte als Publish/Subscribe-Broker.

### Vorbereitende Schritte

Führen Sie den im Abschnitt „Nachricht über IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm veröffentlichen“ auf Seite 87 beschriebenen Schritt aus. Behalten Sie die Verbindung des MQTT-Clientdienstprogramms bei.

### Informationen zu diesem Vorgang

Die Task veranschaulicht zwei Punkte:

- Die Subskription eines Themas in dem einen MQTT-Client und dem Empfang einer Veröffentlichung von einem anderen MQTT-Client.

2. Die Einrichtung von "Punkt-zu-Punkt"-Subskriptionen unter Verwendung von `ClientIdentifier` als Themenzeichenfolge.

Die zusätzliche Task „WebSphere MQ Telemetry-Dämon für Geräte als Publish/Subscribe-Broker verwenden“ auf Seite 96 verwendet anstelle von WebSphere MQ den WebSphere MQ Telemetry-Dämon für Geräte als Publish/Subscribe-Broker.

## Vorgehensweise

1. Verwerfen Sie alle vorhandenen Subskriptionen, indem Sie die Verbindung des MQTT-Clientdienstprogramms trennen und wiederherstellen.

Sofern Sie die Standardeinstellungen nicht geändert haben, wird die Subskription verworfen, da das MQTT-Clientdienstprogramm mit einer bereinigten Sitzung eine Verbindung herstellt (siehe Sitzungen bereinigen).

Geben Sie zur Vereinfachung der Task Ihren eigenen `ClientIdentifier`-Eintrag ein, statt den generierten `ClientIdentifier`-Eintrag zu verwenden, der vom MQTT-Clientdienstprogramm erstellt wurde.

- a) Klicken Sie auf **Verbindung trennen**, um die Verbindung zwischen dem MQTT-Clientdienstprogramm und dem Telemetrikkanal zu trennen.

Das **Clientprotokoll** enthält jetzt das Ereignis `Disconnected` (Verbindung getrennt).

- b) Ändern Sie die **Client-ID** in `MyClient`.

- c) Klicken Sie auf **Connect** (Verbinden).

Das **Clientprotokoll** enthält jetzt das Ereignis `Connected` (Verbindung getrennt).

2. Erstellen Sie eine Subskription für das Thema `MyClient`.

`MyClient` ist der `ClientIdentifier`-Eintrag dieses Clients.

- a) Geben Sie `MyClient` in das Feld **Subscription \ Topic** ein > **Subscribe**.

Das **Clientprotokoll** enthält jetzt das Ereignis `Subscribed` (Verbunden).

3. Starten Sie ein anderes MQTT-Clientdienstprogramm.

- a) Öffnen Sie den Ordner `Queue Managers \ QmgrName \ Telemetry \ channels`.

- b) Klicken Sie mit der rechten Maustaste auf den Kanal **PlainText** > **MQTT-Clientdienstprogramm ausführen ...**

- c) Klicken Sie auf **Connect** (Verbinden).

Das **Clientprotokoll** enthält jetzt das Ereignis `Connected` (Verbindung getrennt).

4. Veröffentlichen Sie `Hello MyClient!` im Thema `MyClient`.

- a) Kopieren Sie das Subskriptionsthema `MyClient` aus dem MQTT-Clientdienstprogramm, das mit dem `ClientIdentifier`-Eintrag `MyClient` ausgeführt wird.

- b) Fügen Sie `MyClient` in das Feld **Publication \ Topic** aller MQTT-Clientdienstprogramminstanzen ein.

- c) Geben Sie `Hello MyClient!` in das Feld **Publication \ message** ein.

- d) Klicken Sie in beiden Instanzen auf **Veröffentlichen**.

## Ergebnisse

Das **Clientprotokoll** im MQTT-Clientdienstprogramm mit dem `ClientIdentifier`-Eintrag `MyClient` enthält zwei Ereignisse des Typs **Received** (Empfangen) und ein Ereignis des Typs **Published** (Veröffentlicht). Die andere Instanz des MQTT-Clientdienstprogramms dokumentiert ein Ereignis des Typs **Published** (Veröffentlicht).

Wenn nur ein Ereignis vom Typ **Empfangen** angezeigt wird, überprüfen Sie die folgenden möglichen Ursachen:

1. Ist die Standardübertragungswarteschlange für den Warteschlangenmanager auf SYSTEM.MQTT.TRANSMIT.QUEUE gesetzt?
2. Haben Sie bei der Durchführung der anderen Übungen Warteschlangenmanager-Aliasnamen oder Definitionen einer fernen Warteschlange erstellt, die auf MyClient verweisen? Falls ein Konfigurationsproblem vorliegt, löschen Sie alle Ressourcen, die auf MyClient verweisen (beispielsweise Warteschlangenmanager-Aliasnamen oder Übertragungswarteschlangen). Trennen Sie die Verbindung der Clientdienstprogramme, stoppen Sie den Telemetrieservice (MQXR) und starten Sie ihn erneut.

## Nächste Schritte

Führen Sie die nächste Task aus („WebSphere MQ Telemetry-Dämon für Geräte als Publish/Subscribe-Broker verwenden“ auf Seite 96). Das MQTT-Clientdienstprogramm verbindet sich nicht mit einem Telemetriekanal, sondern mit dem WebSphere MQ Telemetry-Dämon für Geräte.

### Zugehörige Tasks

Nachricht über IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm veröffentlichen

Führen Sie die hier beschriebenen Schritte aus, um eine Nachricht mit IBM WebSphere MQ Explorer zu veröffentlichen und sie mit dem MQTT-Clientdienstprogramm zu abonnieren. In einer weiteren Task wird veranschaulicht, wie die Standardübertragungswarteschlange nicht auf SYSTEM.MQTT.TRANSMIT.QUEUE gesetzt, sondern ein Warteschlangenmanager-Aliasname konfiguriert wird.

Nachricht mit IBM WebSphere MQ Explorer an einen MQTT-Client senden

Senden Sie eine Nachricht an das MQTT-Clientdienstprogramm, indem Sie eine Nachricht mithilfe von IBM WebSphere MQ Explorer in eine IBM WebSphere MQ-Warteschlange stellen. Diese Task veranschaulicht die Konfiguration einer Definition einer fernen Warteschlange zum direkten Senden einer Nachricht an einen MQTT-Client.

*WebSphere MQ Telemetry-Dämon für Geräte als Publish/Subscribe-Broker verwenden*

Verwenden Sie nicht WebSphere MQ, sondern den WebSphere MQ Telemetry-Dämon für Geräte als Publish/Subscribe-Broker. Veröffentlichen Sie eine Nachricht mit einer Instanz des MQTT-Clientdienstprogramms zum Senden an eine andere Instanz, indem Sie die Subskription mit dem zugehörigen ClientIdentifier-Eintrag als Themenzeichenfolge vornehmen.

## Vorbereitende Schritte

Installieren Sie den Dämon, falls noch nicht geschehen. In: .

Führen Sie die Überprüfung nicht aus; sie verwendet den Port 1883, der bereits vom Telemetriekanal PlainText verwendet wird.

## Informationen zu diesem Vorgang

In der Task verbinden Sie das MQTT-Clientdienstprogramm mit dem WebSphere MQ Telemetry-Dämon unter Verwendung eines TCP/IP-Ports, der vom Standard abweicht. Der eine Client verwendet in seiner Subskription ClientIdentifier als Themenzeichenfolge und der andere Client nimmt die Veröffentlichung für ClientIdentifier vor. Die Vorgehensweise ist mit derjenigen in der vorherigen Task identisch; siehe Vorgehensweise.

**Anmerkung:** In der Task wird die Ausführung des Dämons unter Windows beschrieben. Um den Dämon unter Linux auszuführen, ändern Sie den Pfad und die Berechtigungen für amqtd.

## Vorgehensweise

1. Öffnen Sie ein Befehlsfenster in dem Verzeichnis, das den WebSphere MQ Telemetry-Dämon für Geräte enthält.

Der Verzeichnispfad für Windows lautet *WebSphere MQ installation directory\mqxr\SDK\advanced\DeviceDaemon\windows\_ia32*.

2. Führen Sie den Dämon an einem anderen TCP/IP-Port aus.



- a) Erstellen Sie eine Datei namens `amqtdc.cfg` in dem Verzeichnis des Dämons.
- b) Fügen Sie der Datei eine Zeile hinzu, um einen anderen Standardport für den Dämon zu konfigurieren.

```
port 1884
```

- c) Speichern Sie die Datei.
3. Starten Sie den Dämon.

```
amqtdc
```

Der Dämon schreibt sein Konsolprotokoll in das Befehlsfenster:

```
20100712 123133.857 CWNAN9999I IBM WebSphere MQ Telemetry daemon for devices
20100712 123133.857 CWNAN9997I Licensed Materials - Property of IBM
20100712 123133.857 CWNAN9996I Copyright IBM Corp. 2007, 2024. All Rights Reserved
20100712 123133.857 CWNAN9995I US Government Users Restricted Rights ...
20100712 123133.857 CWNAN0049I Configuration file name is .\amqtdc.cfg
20100712 123133.873 CWNAN0054I Features included: bridge
20100712 123134.060 CWNAN0014I MQTT protocol starting, listening on port 1884
```

4. Starten Sie eine Instanz des MQTT-Clientdienstprogramms.

Starten Sie das MQTT-Clientdienstprogramm nur über einen Telemetrikkanal. Anschließend können Sie sich mit dem Dämon verbinden. Alternativ können Sie das IBM WebSphere MQ-SupportPac IA92 installieren. Das SupportPac ist unter [IA92: WBI Brokers-Java-Implementierung von WebSphere MQ Telemetry Transport](#).

- a) Öffnen Sie den Ordner `Queue Managers\QmgrName\Telemetry\channels`.
- b) Klicken Sie mit der rechten Maustaste auf den Kanal **PlainText > MQTT-Clientdienstprogramm ausführen ...**
- c) Ändern Sie den **Port** in 1884.
- d) Ändern Sie die **Client-ID** in MyClient.
- e) Klicken Sie auf **Verbinden**.

Das **Clientprotokoll** enthält jetzt das Ereignis Connected (Verbindung getrennt).

5. Erstellen Sie eine Subskription für das Thema MyClient.

MyClient ist der ClientIdentifier-Eintrag dieses Clients.

- a) Geben Sie MyClient in das Feld **Subscription \ Topic** ein > **Subscribe**.

Das **Clientprotokoll** enthält jetzt das Ereignis Subscribed (Verbunden).

6. Starten Sie ein anderes MQTT-Clientdienstprogramm.

- a) Öffnen Sie den Ordner `Queue Managers\QmgrName\Telemetry\channels`.
- b) Klicken Sie mit der rechten Maustaste auf den Kanal **PlainText > MQTT-Clientdienstprogramm ausführen ...**
- c) Ändern Sie den **Port** in 1884.
- d) Klicken Sie auf **Verbinden**.

Das **Clientprotokoll** enthält jetzt das Ereignis Connected (Verbindung getrennt).

7. Veröffentlichen Sie Hello MyClient! im Thema MyClient.

- a) Kopieren Sie das Subskriptionsthema MyClient aus dem MQTT-Clientdienstprogramm, das mit dem ClientIdentifier-Eintrag MyClient ausgeführt wird.
- b) Fügen Sie MyClient in das Feld **Publication\Topic** aller MQTT-Clientdienstprogramminstanzen ein.
- c) Geben Sie Hello MyClient! in das Feld **Publication\message** ein.
- d) Klicken Sie in beiden Instanzen auf **Veröffentlichen**.

## Ergebnisse

Das **Clientprotokoll** im MQTT-Clientdienstprogramm mit dem ClientIdentifier-Eintrag MyClient enthält zwei Ereignisse des Typs **Received** (Empfangen) und ein Ereignis des Typs **Published** (Veröffentlicht). Die andere Instanz des MQTT-Clientdienstprogramms dokumentiert ein Ereignis des Typs **Published** (Veröffentlicht).

Sie können die Ereignisse für den Verbindungsauf- und abbau im Zusammenhang mit dem WebSphere MQ Telemetry-Dämon für Geräte auch im Befehlsfenster überwachen.

## Senden Sie von einem MQTT-Client eine Nachricht an eine IBM WebSphere MQ-Anwendung

Eine IBM WebSphere MQ -Anwendung kann eine Nachricht von einem MQTT v3 -Client empfangen, indem sie ein Topic subskribiert. Der MQTT-Client stellt über einen Telemetrikkanal eine Verbindung zu IBM WebSphere MQ her und sendet eine Nachricht an die IBM WebSphere MQ -Anwendung, indem er im selben Thema veröffentlicht.

Führen Sie die Task „Nachricht von einem MQTT-Client an IBM WebSphere MQ Explorer veröffentlichen“ auf Seite 98 aus, um zu lernen, wie eine Veröffentlichung von einem MQTT-Client an eine in IBM WebSphere MQ definierte Subskription gesendet wird.

Falls das Thema in Gruppen zusammengefasst ist oder unter Verwendung der Publish/Subscribe-Hierarchie verteilt wird, kann sich die Subskription auf einem anderen Warteschlangenmanager befinden als dem Warteschlangenmanager, mit dem der MQTT-Client verbunden ist.

### ***Nachricht von einem MQTT-Client an IBM WebSphere MQ Explorer veröffentlichen***

Erstellen Sie eine Subskription für ein Thema unter Verwendung von WebSphere MQ Explorer und veröffentlichen Sie Informationen im Thema unter Verwendung des WebSphere MQTT-Clientdienstprogramms.

## Vorbereitende Schritte

Führen Sie den im Abschnitt „Nachricht über IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm veröffentlichen“ auf Seite 87 beschriebenen Schritt aus. Behalten Sie die Verbindung des MQTT-Clientdienstprogramms bei.

## Informationen zu diesem Vorgang

Die Task veranschaulicht die Veröffentlichung einer Nachricht mit einem MQTT-Client und den Empfang der Veröffentlichung über eine nicht verwaltete permanente Subskription, die unter Verwendung von WebSphere MQ Explorer erstellt wird.

## Vorgehensweise

1. Erstellen Sie eine permanente Subskription für die Themenzeichenfolge MQTT\_Example. Nutzen Sie eines der folgenden Verfahren:
  - Führen Sie das unter Ergebnisse beschriebene Befehlsscript aus.
  - Führen Sie die folgenden Schritte zur Erstellung der Warteschlange und Subskription unter Verwendung von WebSphere MQ Explorer aus.
    - a) Klicken Sie mit der rechten Maustaste auf den Ordner Queue Managers\*QmgrName*\Queues im WebSphere MQ Explorer > **Neu** > **Lokale Warteschlange ...**
    - b) Geben Sie MQTTExampleQueue als Warteschlangenname ein > **Fertigstellen**.
    - c) Klicken Sie mit der rechten Maustaste auf den Ordner Queue Managers\*QmgrName*\Subscriptions im WebSphere MQ Explorer > **Neu** > **Subskription ...**
    - d) Geben Sie MQTTExampleSubscription als Warteschlangenname ein und klicken Sie auf **Weiter**.
    - e) Klicken Sie auf **Auswählen...** > MQTTExampleTopic > **OK**.

Sie haben das Thema `MQTTExampleTopic` bereits in Schritt „4“ auf Seite 89 von „[Nachricht über IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm veröffentlichen](#)“ auf Seite 87 erstellt.

- f) Geben Sie `MQTTExampleQueue` als Bestimmungsname ein und klicken Sie auf **Fertigstellen**.
2. Sie können wahlweise die Warteschlange zur Verwendung eines anderen Benutzers ohne `mqm`-Berechtigung einrichten.

Wenn Sie die Konfiguration für Benutzer mit einer niedrigeren Berechtigung als `mqm` einrichten, müssen Sie für `MQTTExampleQueue` die Berechtigungen `put` und `get` erteilen. Der Zugriff auf das Thema und die Übertragungswarteschlange wurde in „[Nachricht über IBM WebSphere MQ Explorer im MQTT-Clientdienstprogramm veröffentlichen](#)“ auf Seite 87 konfiguriert.

- a) Berechtigen Sie einen Benutzer für die Einreihung in die Warteschlange `MQTTExampleQueue` sowie für den Abruf aus dieser Warteschlange:

```
setmqaut -m qMgrName -t queue -n MQTTExampleQueue -p User ID -all +put +get
```

3. Veröffentlichen Sie `Hello WebSphere MQ!` im Thema `MQTT Example` mithilfe des MQTT-Clientdienstprogramms.

Wenn Sie die Verbindung zum MQTT-Clientdienstprogramm nicht verlassen haben, klicken Sie mit der rechten Maustaste auf den Kanal **PlainText > MQTT-Clientdienstprogramm ausführen ... > Connect**.

- a) Geben Sie `MQTT Example` in das Feld **Publication\Topic** ein.
  - b) Geben Sie `Hello WebSphere MQ!` in das Feld **Publication \ Message > Publishein**.
4. Öffnen Sie den Ordner `Queue Managers\QmgrName\Queues` und suchen Sie den Eintrag `MQTTExampleQueue`.

Das Feld **Aktuelle Warteschlangenlänge** hat den Wert 1

5. Klicken Sie mit der rechten Maustaste auf `MQTTExampleQueue` > **Nachrichten durchsuchen ...** und sehen Sie sich die Veröffentlichung an.

## Nachrichten zwischen dem IBM WebSphere MQ Telemetry-Dämon für Geräte und IBM WebSphere MQ übertragen

Führen Sie diese Task aus, um sich mit dem Senden von Befehlen an den IBM WebSphere MQ Telemetry-Dämon für Geräte vertraut zu machen. Die von Ihnen geschriebenen Befehle bilden eine Brücke, die Nachrichten aus IBM WebSphere MQ an den Dämon und Nachrichten vom Dämon an IBM WebSphere MQ überträgt.

### Vorbereitende Schritte

Führen Sie die Tasks „[Nachricht in einem bestimmten MQTT v3-Client veröffentlichen](#)“ auf Seite 94 und „[WebSphere MQ Telemetry-Dämon für Geräte als Publish/Subscribe-Broker verwenden](#)“ auf Seite 96 aus, um sich mit der Verwendung des MQTT-Clientdienstprogramms vertraut zu machen. Wenn Sie die Tasks abgeschlossen haben, lassen Sie eine Instanz des MQTT-Clientdienstprogramms mit dem Telemetry-Dämon für Geräte verbunden. Lassen Sie eine andere Instanz mit dem Telemetrie Kanal verbunden.

Bei dieser Task wird vorausgesetzt, dass Sie einen Kanal für den Telemetrieservice definiert haben, der an Port 1883 unter der Adresse `127.0.0.1` empfangsbereit ist. Analog hierzu ist das standardmäßige Dämonempfangsprogramm so konfiguriert, dass es an Port 1884 unter der Adresse `127.0.0.1` empfangsbereit ist. Der Port des standardmäßigen Dämonempfangsprogramms ist in einer einzelnen Zeile in der Datei `amqtdd.cfg` konfiguriert. Diese Datei wird in demselben Verzeichnis wie der Dämon (`amqtdd`) gespeichert.

```
port 1884
```

### Informationen zu diesem Vorgang

In dieser Task aktualisieren Sie einen aktiven Dämon, um eine Verbindungsbrücke zum WebSphere MQ-Telemetrieservice (MQXR) zu erstellen und anschließend Nachrichten mit dem Dämon auszutauschen.

**Tipp:** Die Aktualisierungsdatei `amqtd.d.upd` wird nach ihrer Verwendung vom Dämon gelöscht. Wenn Sie die Befehle zur späteren Verwendung aufbewahren möchten, wird empfohlen, die Befehle in einer anderen Datei zu erstellen und diese dann an `amqtd.d.upd` zu übertragen.

## Vorgehensweise

1. Vergewissern Sie sich, dass zwei Instanzen des MQTT-Clientdienstprogramms ausgeführt werden. Die eine Instanz ist mit dem Dämon an Port 1884 verbunden, während die andere mit dem Telemetriekanal verbunden ist, der an Port 1883 ausgeführt wird.
2. Erstellen Sie die Datei `amqtd.d.upd` in dem Verzeichnis des Dämons (`amqtd.d`) und nehmen Sie die folgenden Befehle in der Datei auf.

```
connection daemon1
address 127.0.0.1:1883
topic # in import/ export/
topic # out export/ import/
try_private false
```

- Die Brücke wird als `daemon1` bezeichnet und stellt eine Verbindung zu dem Kanal her, der für den Telemetrieservice (MQXR) konfiguriert ist, der an der Socketadresse `127.0.0.1:1883` ausgeführt wird. Der Befehl `try_private` ist optional; `true` ist der Standardwert. Ohne diese Zeile versucht diese Brücke zunächst, unter Verwendung eines privaten Protokolls, das vom WebSphere MQ Telemetry-Dämon für Geräte verstanden wird, eine Verbindung herzustellen. Wenn Sie `try_private false` in die Befehle aufnehmen, wird dieser Schritt vermieden und die Dauer für die Herstellung einer erfolgreichen Verbindung wird verkürzt.
- Die Zeile `topic # in import/ export/` weist `daemon1` an, alle Themen zu abonnieren, die der Themenzeichenfolge `export/#` entsprechen, die im Warteschlangenmanager erstellt wird. Sie überträgt die entsprechenden Veröffentlichungen vom Warteschlangenmanager an den Dämon und ändert den Anfang der Themenzeichenfolge von `export/` in `import/`. Die Zeile `topic # out export/ import/` erstellt eine Subskription beim lokalen Dämon. Die Brücke abonniert alle Themen, die der Themenzeichenfolge `export/#` entsprechen, die im Dämon erstellt wird. Sie überträgt die Veröffentlichungen vom Dämon an den Warteschlangenmanager und ändert den Anfang der Themenzeichenfolge von `export/` in `import/`.

Abbildung 25 auf Seite 100 zeigt das Ergebnis im Konsolprotokoll.

```
CWNAN0124I Starting bridge connection daemon1
CWNAN0133I Bridge connection daemon1 to 127.0.0.1:1883 now established
```

Abbildung 25. Konsolprotokoll über den Start der Verbindungsbrücke

3. Geben Sie in jeder Instanz des MQTT-Clientdienstprogramms `import/#` im Eingabefeld **Subskription/Thema:** > **Subskribiererein.**
4. Geben Sie in jeder Instanz des MQTT-Clientdienstprogramms `export/#` in das Eingabefeld **Publication/Topic:** ein.
  - a) Geben Sie im MQTT-Clientdienstprogramm, das mit Port 1883 verbunden ist, im Telemetriekanal `From the queue manager` im Eingabefeld **Publication/Message:** > **Publishein.**
  - b) Geben Sie im MQTT-Clientdienstprogramm, das mit Port 1884 verbunden ist, den Telemetriedämon `From the daemon` in das Eingabefeld **Publication/Message:** > **Publishein.**

Das Clientprotokoll in jedem MQTT-Clientdienstprogramm enthält die Veröffentlichung, die von einem Broker an den anderen übertragen wurde.

## MQTT-Publish/Subscribe-Anwendungen

Verwenden Sie die themenbasierte Publish/Subscribe-Funktion zum Schreiben von MQTT-Anwendungen.

Wenn der MQTT-Client verbunden ist, fließen Veröffentlichungen zwischen dem Client und Server in beide Richtungen. Die Veröffentlichungen werden vom Client gesendet, sobald Informationen auf dem Client veröffentlicht werden. Veröffentlichungen werden beim Client empfangen, sobald eine Nachricht in einem Thema veröffentlicht wird, das einer vom Client erstellten Subskription entspricht.

Die von den MQTT-Clients erstellten Themen und Subskriptionen werden vom WebSphere MQ-Publish/Subscribe-Broker verwaltet. Die von MQTT-Clients erstellten Themen nutzen denselben Themenbereich wie Themen, die von WebSphere MQ-Anwendungen erstellt werden.

Veröffentlichungen, die der Themenzeichenfolge in einer MQTT-Clientsubskription entsprechen, werden in die Warteschlange `SYSTEM.MQTT.TRANSMIT.QUEUE` gestellt, wobei der Name des fernen Warteschlangenmanagers auf den `ClientIdentifier`-Eintrag des Clients gesetzt ist. Der Telemetrieservice (MQXR) leitet die Veröffentlichungen an den Client weiter, von dem die Subskription erstellt wurde. Er verwendet den `ClientIdentifier`-Eintrag, der als Name des fernen Warteschlangenmanagers festgelegt wurde, zur Identifizierung des Clients.

Für gewöhnlich muss `SYSTEM.MQTT.TRANSMIT.QUEUE` als Standardübertragungswarteschlange definiert werden. Es ist möglich, aber mühsam, MQTT so zu konfigurieren, dass es nicht die Standardübertragungswarteschlange verwendet; siehe [Konfigurieren der verteilten Steuerung von Warteschlangen zum Senden von Nachrichten an MQTT-Clients](#).

Ein MQTT-Client kann eine persistente Sitzung erstellen; siehe [„Statusunabhängige und statusabhängige MQTT-Sitzungen“](#) auf Seite 105. Subskriptionen, die in einer persistenten Sitzung erstellt werden, sind permanent. Veröffentlichungen, die für einen Client mit einer persistenten Sitzung eingehen, werden in der Warteschlange `SYSTEM.MQTT.TRANSMIT.QUEUE` gespeichert und an den Client weitergeleitet, wenn er die Verbindung wiederherstellt.

Ein MQTT-Client kann auch in ständigen Veröffentlichungen Informationen veröffentlichen und abonnieren; siehe [Ständige Veröffentlichungen und MQTT-Clients](#). Ein Subskribent eines Themas in einer ständigen Veröffentlichung empfängt die neueste Veröffentlichung des Themas. Der Subskribent empfängt die ständige Veröffentlichung, wenn er eine Subskription erstellt oder die Verbindung zu seiner früheren Sitzung wiederherstellt.

### **Zugehörige Tasks**

[Erste MQ Telemetry Transport-Publisher-Anwendung mit Java erstellen](#)

[Subskribenten für MQ Telemetry Transport mit Java erstellen](#)

## **Telemetrieanwendungen**

Sie können unter Verwendung von WebSphere MQ- oder WebSphere Message Broker-Nachrichtenflüssen Telemetrieanwendungen schreiben.

Verwenden Sie JMS, MQI oder sonstige WebSphere MQ-Programmierschnittstellen, um Telemetrieanwendungen in WebSphere MQ zu programmieren.

Der Telemetrieservice (MQXR) nimmt eine Konvertierung zwischen MQTT v3-Nachrichten und WebSphere MQ-Nachrichten vor. Er erstellt Subskriptionen und Veröffentlichungen für MQTT-Clients und leitet Veröffentlichungen an MQTT-Clients weiter. Eine Veröffentlichung sind die Nutzdaten einer MQTT v3-Nachricht. Die Nutzdaten umfassen Nachrichtenheader und eine Bytefeldgruppe im Format `jms-bytes`. Der Telemetrieserver ordnet die Header zwischen einer MQTT v3-Nachricht und einer WebSphere MQ-Nachricht zu; siehe [„Integration von WebSphere MQ Telemetry in Warteschlangenmanager“](#) auf Seite 102.

Verwenden Sie die `Publication`-, `MQInput`- und `JMSInput`-Knoten, um Publizierungen zwischen WebSphere Message Broker und MQTT-Clients zu senden und empfangen.

Mithilfe von Nachrichtenflüssen können Sie die Telemetrie in Websites mit HTTP integrieren sowie in andere Anwendungen, die WebSphere MQ und WebSphere-Adapter verwenden.

WebSphere MQ Telemetry ersetzt die SCADA-Knoten in WebSphere Message Broker Version 7. Informationen zur Migration von WebSphere Message Broker-Nachrichtenflüssen unter Verwendung der `SCADAInput`- und `SCADAOutput`-Knoten auf Version 7 finden Sie im Abschnitt [Migration von Telemetrieanwendungen von WebSphere Message Broker Version 6 auf IBM WebSphere MQ Telemetry und WebSphere Message Broker Version 7.0](#).

## Integration von WebSphere MQ Telemetry in Warteschlangenmanager

Der MQTT-Client ist als Publish/Subscribe-Anwendung in WebSphere MQ integriert. Er kann Themen in WebSphere MQ entweder veröffentlichen oder abonnieren, neue Themen erstellen oder vorhandene Themen verwenden. Er empfängt Publizierungen von WebSphere MQ, wenn MQTT-Clients (ihn selbst eingeschlossen) oder andere WebSphere MQ-Anwendungen Informationen in den Themen seiner Subskriptionen veröffentlichen. Bei der Festlegung der Attribute einer Veröffentlichung gelten bestimmte Regeln.

Viele der Attribute, die den von WebSphere MQ bereitgestellten Themen, Publizierungen, Subskriptionen und Nachrichten zugeordnet sind, werden nicht unterstützt. Unter „MQTT-Client an WebSphere MQ-Publish/Subscribe-Broker“ auf Seite 102 und „WebSphere MQ an einen MQTT-Client“ auf Seite 103 wird beschrieben, wie Attribute von Veröffentlichungen festgelegt werden. Die Einstellungen hängen davon ab, ob die Publizierung an den WebSphere MQ-Publish/Subscribe-Broker geleitet wird oder von diesem stammt.

In der Publish/Subscribe-Funktion von WebSphere MQ sind die Themen administrativen Themenobjekten zugeordnet. Dies gilt auch für Themen, die von MQTT-Clients erstellt werden. Wenn ein MQTT-Client eine Themenzeichenfolge für eine Publizierung erstellt, ordnet der WebSphere MQ-Publish/Subscribe-Broker diese einem administrativen Themenobjekt zu. Der Broker ordnet die Themenzeichenfolge in der Veröffentlichung dem nächsten übergeordneten administrativen Themenobjekt zu. Die Zuordnung ist mit derjenigen bei WebSphere MQ-Anwendungen identisch. Wenn kein Thema vorhanden ist, das von einem Benutzer erstellt wurde, wird das Veröffentlichungsthema SYSTEM.BASE.TOPIC zugeordnet. Die Attribute der Veröffentlichung leiten sich aus dem Themenobjekt ab.

Wenn eine Subskription von einer WebSphere MQ-Anwendung oder einem Administrator erstellt wird, trägt sie einen Namen. Listen Sie Subskriptionen mit WebSphere MQ Explorer oder mit `runmqsc`- oder PCF-Befehlen auf. Alle MQTT-Clientsubskriptionen haben einen Namen. Sie erhalten einen Namen im folgenden Format: *ClientIdentifizier:Topic name*

### MQTT-Client an WebSphere MQ-Publish/Subscribe-Broker

Ein MQTT-Client hat eine Publizierung an WebSphere MQ gesendet. Der Telemetrieservice (MQXR) konvertiert die Publizierung in eine WebSphere MQ-Nachricht. Die WebSphere MQ-Nachricht enthält drei Teile:

1. MQMD
2. RFH2
3. Nachricht

MQMD-Eigenschaften werden auf ihre Standardwerte gesetzt, es sei denn, dies ist in Tabelle 3 auf Seite 102 anders angegeben.

MQMD-Feld	Typ	Wert
<b>Format</b>	MQCHAR8	MQFMT_RF_HEADER_2
<b>UserIdentifizier</b>	MQCHAR12	Legen Sie einen der folgenden Werte fest: MqttClient.ClientIdentifizier MqttConnectOptions.UserName Eine Benutzer-ID, die vom WebSphere MQ-Administrator für den Telemetrie Kanal festgelegt wird
<b>Priority</b>	MQLONG	MQPRI_PRIORITY_AS_Q_DEF (Standardeinstellung bei WebSphere MQ, während JMS den Standardwert 4 hat.)

MQMD-Feld	Typ	Wert
<b>Persistence</b>	MQLONG	QoS=0→MQPER_NOT_PERSISTENT QoS=1→MQPER_PERSISTENT QoS=2→MQPER_PERSISTENT

Der Header RFH2 enthält keinen Ordner <msd> zum Definieren des Typs der JMS-Nachricht. Der Telemetrieservice (MQXR) erstellt die WebSphere MQ-Nachricht als JMS-Standardnachricht. Der JMS-Standardnachrichtentyp ist eine `javax.jms.BytesMessage`-Nachricht. Eine Anwendung kann in Form von Nachrichteneigenschaften auf die zusätzlichen Kopfzeileninformationen zugreifen; siehe [Nachrichteneigenschaften](#).

RFH2-Werte werden wie in [Tabelle 4 auf Seite 103](#) dargestellt festgelegt. Die Eigenschaft 'Format' wird im festen RFH2-Header festgelegt, während die übrigen Werte in RFH2-Ordern festgelegt werden.

RFH2-Eigenschaft	Typ/Ordner	Kopfzeile
Format	MQCHAR8	MQFMT_NONE
ClientIdentifier	mqtt/clientId	Kopieren Sie <code>MqttClient.ClientIdentifier</code> mit einer Länge von 1...23 Bytes.
QoS	mqtt/qos	Kopieren Sie QoS aus der eingehenden MQTT-Nachricht.
Nachrichten-ID	mqtt/msgid	Kopieren Sie die Nachrichten-ID (Message ID) aus der eingehenden MQTT-Nachricht, wenn QoS den Wert 1 oder 2 hat.
MQIsRetained	mmps/Ret	Festgelegt, wenn die ursprüngliche MQTT-Veröffentlichung mit der Einstellung der Eigenschaft RETAIN gesendet wurde und die Nachricht als ständige Veröffentlichung empfangen wird.
MQTopicString	mmps/Top	Das Thema, in dem die MQTT-Nachricht veröffentlicht wurde.

Die Nutzdaten in einer MQTT-Publizierung werden dem Inhalt einer WebSphere MQ-Nachricht zugeordnet:

Nachrichteninhalte	Typ	Inhalt
Buffer	MQBYTEN	Kopie der Bytes der eingehenden MQTT-Nachricht. Die Länge kann null sein.

## WebSphere MQ an einen MQTT-Client

Ein Client hat ein Veröffentlichungsthema abonniert. Eine WebSphere MQ-Anwendung hat Informationen im Thema veröffentlicht, was dazu führt, dass vom WebSphere MQ-Publish/Subscribe-Broker eine Publizierung an den MQTT-Subskribenten gesendet wird. Möglich ist auch, dass eine WebSphere MQ-Anwendung eine nicht erwartete Nachricht direkt an einen MQTT-Client gesendet hat. In [Tabelle 6 auf Seite 104](#) wird beschrieben, wie die festgelegten Nachrichtenheader in der Nachricht festgelegt werden, die an den MQTT-Client gesendet wird. Alle anderen Daten im WebSphere MQ-Nachrichtenheader sowie alle anderen Header werden gelöscht. Die Nachrichtendaten in der WebSphere MQ-Nachricht werden unverändert als Nachrichtennutzdaten in der MQTT-Nachricht gesendet. Die MQTT-Nachricht wird vom Telemetrieservice (MQXR) an den MQTT-Client gesendet.

Tabelle 6. Eigenschaften des festgelegten MQTT-Headers

MQTT-Feld	Typ	Wert
<b>DUP</b>	Boolesch	Festgelegt, wenn der Wert QoS = 1 oder 2 eingestellt wurde und die Nachricht in einer früheren Übertragung an diesen Client gesendet, jedoch nach einer gewissen Zeit noch nicht bestätigt wurde.
<b>QoS</b>	int	<p>Wie der Wert von QoS in einer ausgehenden Publizierung vom Publish/Subscribe-Broker in WebSphere MQ festgelegt wird, hängt von der eingehenden Publizierung ab. Hierbei ist relevant, ob die eingehende Publizierung von einem MQTT-Client oder von einer WebSphere MQ-Anwendung gesendet wurde.</p> <p><b>MQTT</b> Legen Sie in der eingehenden Veröffentlichung einen niedrigeren QoS-Wert (Servicequalität) fest. Dasselbe gilt für die vom Subskribenten angeforderte QoS.</p> <p><b>WebSphere MQ</b> Verringern Sie den Wert der von der eingehenden Veröffentlichung abgeleiteten QoS:  MQPER_NOT_PERSISTENT→QoS=0  MQPER_PERSISTENT→QoS=2</p> <p>Dasselbe gilt für die vom Subskribenten angeforderte QoS. Wenn die Nachricht ohne Subskription an den Client gesendet wird, wird die Servicequalität (QoS) standardmäßig auf 2 gesetzt. Ein Client kann diesen Wert ändern, indem er DEFAULT.QoS mit einer anderen QoS abonniert.</p>
<b>RETAIN</b>	Boolesch	Festgelegt, wenn für die eingehende Veröffentlichung die Eigenschaft 'retained' (ständig) festgelegt wurde.

In Tabelle 7 auf Seite 104 wird beschrieben, wie die variablen Nachrichtenheader in der MQTT-Nachricht festgelegt werden, die an den MQTT-Client gesendet wird.

Tabelle 7. Eigenschaften des variablen MQTT-Headers

MQTT-Feld	Typ	Wert
<b>Topic name</b>	Zeichenfolge	Die Themenzeichenfolge, mit der die Nachricht veröffentlicht wurde.
<b>Message ID</b>	Zeichenfolge	Die letzten 2 Bytes der Eigenschaft MQMD.MsgId der Veröffentlichung, wenn sie in SYSTEM.MQTT.TRANSMIT.QUEUE gestellt wird.
<b>Payload</b>	byte[]	Direktkopie der Bytes aus der eingehenden Veröffentlichung an den Publish/Subscribe-Broker. Die Länge kann null sein.

## Telemetry-Dämon für Geräte

Der WebSphere MQ Telemetry-Dämon für Geräte ist eine erweiterte MQTT V3-Clientanwendung. Damit können Sie Nachrichten von anderen MQTT-Clients speichern und weiterleiten (Store-and-forward-Verfahren). Der Dämon stellt wie ein MQTT-Client eine Verbindung zu WebSphere MQ her, Sie können jedoch auch andere MQTT-Clients mit ihm verbinden. Sie können ihn außerdem mit anderen Telemetriedämonen verbinden.



Er dient vier grundlegenden Zwecken:

### 1. Verbinden lokaler MQTT-Clients in einem Publish/Subscribe-Netz

Sie können den Sensor und einen Aktuator eines Geräts als separate MQTT-Clients mit dem Dämon verbinden. Der Sensor veröffentlicht seine gemessenen Werte und der Aktuator abonniert die gemessenen Werte und ändert sein Verhalten auf Basis dieser Werte. Die gemessenen Werte werden lokal geprüft.

### 2. Filtern welche Subskriptionen und welche Nachrichten für den Warteschlangenmanager und das Gerät veröffentlicht werden.

Im vorherigen Beispiel kann ein WebSphere Message Broker-Nachrichtenfluss das Thema abonnieren, in dem der Dämon die gemessenen Werte veröffentlicht. Der Fluss aktualisiert eine Webseite und zeigt den Status des Geräts an.

Der Dämon kann die Subskription, die vom Aktuator erstellt wurde, auch an den Warteschlangenmanager weiterleiten. Ein WebSphere Message Broker-Fluss veröffentlicht eine Nachricht in dem Thema, das von dem MQTT-Client abonniert wurde, der den Aktuator bedient. Der MQTT-Client ändert die Geräteeinstellungen.

Der Nachrichtenfluss kann unter Verwendung eines HTTPInput-Knotens von WebSphere Message Broker über eine Webseite gestartet werden.

### 3. Konzentrieren Sie mehrere MQTT-Clients in einer Verbindung mit dem Telemetrieserver.

Statt einer gesonderten Verbindung der einzelnen Geräte mit dem Telemetrieserver leitet der Dämon Veröffentlichungen und Subskriptionen über eine einzelne TCP/IP-Verbindung weiter. Der Dämon verringert so die Anzahl der TCP/IP-Verbindungen, die vom Telemetrieservice (MQXR) verwaltet werden müssen.

Einzelne MQTT-Clients verbinden sich mit dem Dämon. Die einzelnen Clients sind für den Warteschlangenmanager nicht sichtbar. Der Dämon stellt für alle Clients, die sich mit ihm verbinden, eine Verbindung zum Warteschlangenmanager her.

### 4. Speichern und Weiterleiten von Nachrichten zwischen Geräten und dem Warteschlangenmanager

Der Dämon übernimmt bei der Verbindung mit dem Warteschlangenmanager die Zuständigkeit für den Schutz der Telemetriegeräte vor Verbindungsfehlern, die von kurzer Dauer sind.

Ein Gerät unterstützt möglicherweise nur die Nachrichtenübermittlung auf Basis des "Fire-and-Forget"-Prinzips. Wenn die Verbindung zum Warteschlangenmanager nur zeitweise verfügbar oder störanfällig ist, kann das Gerät Informationen nicht planbar oder zuverlässig übertragen.

Dieses Problem kann gelöst werden, indem das Gerät über eine lokale Verbindung, die stets verfügbar ist, mit dem Dämon verbunden wird. Der Dämon kann die Nachrichten, die zu und vom Warteschlangenmanager fließen, in seinem Speicher puffern. Er kann sich eine zuverlässige Servicequalität zu Nutze machen, damit die Nachrichten über eine störanfällige Verbindung an den und vom Warteschlangenmanager gesendet werden können.

**Anmerkung:** Der Dämon verfügt über keinen permanenten Speicher für "unvollständig verarbeitete" Nachrichten. Die Nachrichten werden im Speicher gepuffert.

## Statusunabhängige und statusabhängige MQTT-Sitzungen

MQTT-Clients können eine statusabhängige Sitzung mit dem Warteschlangenmanager erstellen. Wenn ein statusabhängiger MQTT-Client getrennt wird, behält der Warteschlangenmanager die vom Client erstellten Subskriptionen und unvollständig verarbeitete Nachrichten. Sobald der Client die Verbindung wiederherstellt, löst er unvollständig verarbeitete Nachrichten auf. Er sendet alle Nachrichten, die für die Zustellung eingereicht wurden und empfängt Nachrichten, die für seine Subskriptionen veröffentlicht wurden, solange er getrennt war.

Wenn ein MQTT-Client eine Verbindung zu einem Telemetriekanal herstellt, startet er entweder eine neue Sitzung oder setzt eine alte Sitzung fort. Eine neue Sitzung verfügt weder über noch nicht bestätigte, ausstehende Nachrichten, noch über Subskriptionen oder Veröffentlichungen, die zur Zustellung anstehen. Wenn ein Client eine Verbindung herstellt, gibt er an, ob der Start mit einer neuen Sitzung erfolgen oder eine bestehende Sitzung fortgesetzt werden soll (siehe [Sitzungen bereinigen](#)).

Wenn der Client eine bestehende Sitzung fortsetzt, erfolgt die Verarbeitung, als ob die Verbindung niemals unterbrochen worden wäre. Veröffentlichungen, die zur Zustellung anstehen, werden an den Client gesendet und alle noch nicht festgeschriebenen Nachrichtenübertragungen werden ausgeführt. Wenn ein Client in einer persistenten Sitzung die Verbindung zum Telemetrieservice (MQXR) trennt, bleiben alle Subskriptionen, die vom Client erstellt wurden, erhalten. Veröffentlichungen für die Subskriptionen werden an den Client gesendet, sobald dieser die Verbindung wiederherstellt. Wenn er die Verbindung wiederherstellt, ohne die alte Sitzung fortzusetzen, werden die Veröffentlichungen vom Telemetrieservice (MQXR) gelöscht.

Die Informationen zum Sitzungsstatus werden vom Warteschlangenmanager in der Warteschlange `SYS-TEM.MQTT.PERSISTENT.STATE` gespeichert.

Der WebSphere MQ-Administrator kann eine Sitzung trennen und löschen.

## Wenn ein MQTT-Client nicht verbunden ist

Wenn ein Client nicht verbunden ist, kann der Warteschlangenmanager für ihn weiterhin Veröffentlichungen empfangen. Diese werden an den Client weitergeleitet, sobald er die Verbindung wiederherstellt. Ein Client kann ein "Last Will and Testament" erstellen, das vom Warteschlangenmanager für den Client veröffentlicht wird, wenn die Verbindung des Clients unerwartet getrennt wird.

Wenn Sie benachrichtigt werden möchten, sobald der Client unerwartet die Verbindung trennt, können Sie eine Veröffentlichung des Typs 'Last Will and Testament' registrieren (siehe [Veröffentlichung des Typs 'Last Will and Testament'](#)). Sie wird vom Telemetrieservice (MQXR) gesendet, wenn dieser feststellt, dass die Verbindung zum Client ohne eine entsprechende Trennungsanforderung des Clients unterbrochen wurde.

Ein Client kann eine ständige Veröffentlichung jederzeit veröffentlichen; siehe [Ständige Veröffentlichungen und MQTT -Clients](#). In einer neuen Subskription eines Themas kann angefordert werden, dass alle ständige Veröffentlichungen im Zusammenhang mit diesem Thema gesendet werden. Wenn Sie das Last Will and Testament als ständige Veröffentlichung erstellen, können Sie mit dessen Hilfe den Status eines Clients überwachen.

Der Client veröffentlicht beispielsweise eine ständige Veröffentlichung, wenn er sich verbindet, in der seine Verfügbarkeit bekannt gemacht wird. Gleichzeitig erstellt er eine ständige Veröffentlichung des Typs 'Last Will and Testament', die seine Nichtverfügbarkeit ankündigt. Außerdem veröffentlicht er seine Nichtverfügbarkeit als ständige Veröffentlichung unmittelbar vor einem geplanten Verbindungsabbau. Um festzustellen, ob der Client verfügbar ist, müssen Sie das Thema der ständigen Veröffentlichung abonnieren. Sie erhalten in diesem Fall immer eine der drei Veröffentlichungen.

Wenn der Client veröffentlichte Nachrichten erhalten soll, wenn er getrennt ist, müssen Sie den Client mit seiner vorherigen Sitzung erneut verbinden; siehe [„Statusunabhängige und statusabhängige MQTT-Sitzungen“](#) auf Seite 105. Seine Subskriptionen sind aktiv, bis sie gelöscht werden oder bis der Client eine neue (bereinigte) Sitzung erstellt.

## Lose Kopplung zwischen MQTT-Clients und WebSphere MQ-Anwendungen

Der Veröffentlichungsfluss zwischen MQTT-Clients und WebSphere MQ-Anwendungen wird lose verbunden. Veröffentlichungen können entweder von einem MQTT-Client oder einer WebSphere MQ-Anwendung stammen und haben keine festgelegte Reihenfolge. Die Veröffentlichungskomponenten und Subskribenten sind lose verbunden. Sie interagieren indirekt über Veröffentlichungen und Subskriptionen. Sie können Nachrichten von einer WebSphere MQ-Anwendung auch direkt senden.

MQTT-Clients und WebSphere MQ-Anwendungen sind auf zwei Arten lose verbunden:

1. Veröffentlichungskomponenten und Subskribenten sind durch die Zuordnung einer Veröffentlichung und einer Subskription zu einem Thema lose verbunden. Veröffentlichungskomponenten und Subskribenten haben normalerweise keine Kenntnis von der Adresse oder Identität der anderen Quelle einer Veröffentlichung oder Subskription.
2. MQTT-Clients veröffentlichen, abonnieren, empfangen Veröffentlichungen und verarbeiten Empfangsbestätigungen in separaten Threads.

Eine MQTT-Clientanwendung wartet nicht, bis eine Veröffentlichung zugestellt wurde. Sie übergibt eine Nachricht an den MQTT-Client und dann fährt die Anwendung in einem eigenen Thread fort. Die Anwendung wird über ein Zustellungstoken mit der Zustellung einer Veröffentlichung synchronisiert (siehe [Zustellungstokens](#)).

Nach der Übergabe einer Nachricht an den MQTT-Client hat die Anwendung die Option, im Rahmen des Zustellungstokens zu warten. Statt zu warten, kann der Client eine Callback-Methode bereitstellen, die aufgerufen wird, wenn die Veröffentlichung an WebSphere MQ zugestellt wird. Das Zustellungstoken kann auch ignoriert werden.

Abhängig von der Servicequalität, die der Nachricht zugeordnet ist, wird das Zustellungstoken sofort oder möglicherweise auch nach einer langen Zeit an die Callback-Methode zurückgegeben. Das Zustellungstoken kann sogar zurückgegeben werden, nachdem die Verbindung des Clients getrennt und wiederhergestellt wurde. Bei der Servicequalität "Fire and Forget" wird das Zustellungstoken sofort zurückgegeben. In den anderen beiden Fällen wird das Zustellungstoken nur zurückgegeben, wenn der Client die Bestätigung erhält, dass die Veröffentlichung an Subskribenten gesendet wurde.

Veröffentlichungen, die als Folge einer Clientsubskription an einen MQTT-Client gesendet werden, werden an die Callback-Methode `messageArrived` übermittelt. `messageArrived` wird in einem anderen Thread als die Hauptanwendung ausgeführt.

## Nachrichten direkt an einen MQTT-Client senden

Sie können eine Nachricht auf eine von zwei Arten an einen bestimmten MQTT-Client senden.

1. Eine WebSphere MQ -Anwendung kann eine Nachricht direkt ohne Subskription an einen MQTT-Client senden; siehe [Nachricht direkt an einen Client senden](#).
2. Eine alternative Methode ist die Verwendung Ihrer `ClientIdentifier`-Namenskonvention. Veranlassen Sie, dass alle MQTT-Subskribenten Subskriptionen unter Verwendung ihres eindeutigen `ClientIdentifier`-Eintrags als Thema verwenden. Nehmen Sie Veröffentlichungen für `ClientIdentifier` vor. Die Veröffentlichung wird an den Client gesendet, der das Thema `ClientIdentifier` abonniert hat. Mit diesem Verfahren können Sie eine Veröffentlichung an einen bestimmten MQTT-Subskribenten senden.

## Sicherheit in WebSphere MQ Telemetry

Der Schutz von Telemetriegeräten kann wichtig sein, da die Geräte in der Regel tragbar sind und an Orten eingesetzt werden, die nicht genau kontrolliert werden können. Sie können die Verbindung zwischen dem MQTT-Gerät und dem Telemetrieservice (MQXR) mithilfe eines virtuellen privaten Netzes (VPN) schützen. WebSphere MQ Telemetry bietet die beiden Sicherheitsmechanismen Secure Sockets Layer (SSL) und Java Authentication and Authorization Service (JAAS).

SSL wird in erster Linie zur Verschlüsselung der Kommunikation zwischen dem Gerät und dem Telemetriekanal und für die Authentifizierung der Geräteverbindung mit dem richtigen Server verwendet (siehe [Telemetriekanalauthentifizierung mit SSL](#)). Sie können mithilfe von SSL auch überprüfen, ob sich das Clientgerät mit dem Server verbinden darf (siehe [MQTT-Clientauthentifizierung mit SSL](#)).

Mit JAAS wird hauptsächlich überprüft, ob die Serveranwendung von dem Gerätebenutzer verwendet werden darf (siehe [MQTT-Clientauthentifizierung mit einem Kennwort](#)). JAAS kann in Verbindung mit LDAP verwendet werden, um ein Kennwort über ein Verzeichnis für einmalige Anmeldungen zu überprüfen.

Wenn SSL und JAAS kombiniert werden, wird die Authentifizierung über zwei Faktoren abgesichert. Sie können die von SSL verwendeten Chiffrierwerte auf Chiffrierwerte beschränken, welche die Federal Information Processing Standards erfüllen.

Mit Zehntausenden von Benutzern oder mehr ist die Bereitstellung einzelner Sicherheitsprofile nicht immer in der Praxis umsetzbar. Außerdem ist es häufig umständlich, die Profile für die Autorisierung einzelner Benutzer für den Zugriff auf WebSphere MQ-Objekte zu verwenden. Gruppieren Sie stattdessen Benutzer in Klassen für die Autorisierung von Veröffentlichungen und Subskriptionen von Themen und für das Senden von Veröffentlichungen an Client.

Konfigurieren Sie jeden Telemetriekanal so, dass Clients zu allgemeinen Client-Benutzer-IDs zugeordnet werden. Verwenden Sie eine allgemeine Benutzer-ID für jeden Client, der sich in einem bestimmten Kanal verbindet (siehe [MQTT-Clientidentität und Autorisierung](#)).

Die Autorisierung von Benutzergruppen umfasst keine Authentifizierung aller Einzelpersonen. Jeder einzelne Benutzer kann beim Client oder Server durch seinen Benutzernamen und das zugehörige Kennwort authentifiziert und dann mithilfe einer allgemeinen Benutzer-ID beim Server autorisiert werden.

## Globalisierung in WebSphere MQ Telemetry

Die Nachrichtennutzdaten im MQTT v3-Protokoll werden als Bytefeldgruppe codiert. Für gewöhnlich erstellen Anwendungen, die den Text verarbeiten, die Nachrichtennutzdaten in UTF-8. Der Telemetriekanal beschreibt die Nachrichtennutzdaten als UTF-8, nimmt jedoch keine Codepagekonvertierungen vor. Die Themenzeichenfolge der Veröffentlichung muss UTF-8 sein.

Die Anwendung ist für die Konvertierung alphabetischer Daten in die richtige Codepage und für die Konvertierung numerischer Daten in die richtige Zahlencodierung verantwortlich.

Der MQTT-Java-Client verfügt über eine komfortable `MqttMessage.toString`-Methode. Die Methode behandelt die Nachrichtennutzdaten, als ob sie in dem Standardzeichensatz der lokalen Plattform codiert wären; dies ist für gewöhnlich UTF-8. Sie konvertiert die Nutzdaten in eine Java-Zeichenfolge. Java hat eine Zeichenfolgemethode, `getBytes`, die eine Zeichenfolge in eine Bytefeldgruppe konvertiert, die mit dem Standardzeichensatz der lokalen Plattform codiert ist. Zwei MQTT-Java-Programme, die Text in den Nachrichtennutzdaten zwischen Plattformen mit demselben Standardzeichensatz austauschen, tun dies in UTF-8-einfach und effizient.

Wenn eine der Plattformen nicht den Standardzeichensatz UTF-8 hat, müssen die Anwendungen eine Konvention für den Austausch von Nachrichten erstellen. Die Veröffentlichungskomponente gibt beispielsweise eine Konvertierung aus einer Zeichenfolge in UTF-8 an, wobei die Methode `getBytes("UTF8")` verwendet wird. Zum Empfang des Nachrichtentextes geht der Subskribent davon aus, dass die Nachricht im Zeichensatz UTF-8 codiert ist.

Der Telemetrieservice (MQXR) beschreibt die Codierung aller eingehenden Veröffentlichungen von MQTT-Client als UTF-8. Er setzt `MQMD.CodedCharSetId` auf UTF-8 und `RFH2.CodedCharSetId` auf `MQCCSI_INHERIT`; siehe „[Integration von WebSphere MQ Telemetry in Warteschlangenmanager](#)“ auf Seite 102. Da das Format der Veröffentlichung auf `MQFMT_NONE` gesetzt ist, kann keine Konvertierung durch Kanäle oder MQGET erfolgen.

## Leistung und Skalierbarkeit von WebSphere MQ Telemetry

Beachten Sie die folgenden Faktoren beim Verwalten einer großen Anzahl an Clients und Verbesserung der Skalierbarkeit von WebSphere MQ Telemetry.

### Kapazitätsplanung

Informationen zu Leistungsberichten für WebSphere MQ Telemetry erhalten Sie, wenn Sie den Bericht [WebSphere MQ Telemetry-Leistungsbewertungen von WebSphere MQ Family-Leistungsberichteauswählen](#).

### Verbindungen

Verbindungen bringen unter anderem folgenden Aufwand mit sich:

- Den Einrichtungsaufwand für die Verbindung selbst im Hinblick auf die Prozessorbelegung und die Dauer
- Den Netzaufwand
- Die Speicherbelegung, wenn eine Verbindung offen gehalten, jedoch nicht verwendet wird.

Wenn Clients verbunden bleiben, besteht eine zusätzliche Belastung. Wird eine Verbindung offen gehalten, verwenden TCP/IP-Verarbeitungsabläufe und MQTT-Nachrichten das Netz, um zu prüfen, ob die

Verbindung immer noch besteht. Außerdem wird für jede Clientverbindung, die offen gehalten wird, Serverspeicher belegt.

Wenn Sie mehr als eine Nachricht pro Minute senden, halten Sie Ihre Verbindung offen, um den Aufwand für den Aufbau einer neuen Verbindung zu vermeiden. Wenn Sie weniger als eine alle 10 bis 15 Minuten senden, sollten Sie erwägen, die Verbindung zu trennen, um den Aufwand zu vermeiden, der entsteht, wenn sie offen gehalten wird. Es kann sinnvoll sein, über längere Zeiträume eine SSL-Verbindung offen, aber inaktiv zu halten, da ihr Aufbau sehr aufwändig ist.

Berücksichtigen Sie außerdem die Funktionalität des Clients. Wenn der Client über die Funktion eines Store-and-forward-Verfahrens verfügt, kann es hilfreich sein, Nachrichten zu stapeln und die Verbindung zwischen dem Versenden der Stapel zu trennen. Wird die Verbindung des Clients jedoch getrennt, kann er keine Nachrichten vom Server empfangen. Der Zweck Ihrer Anwendung beeinflusst daher Ihre Entscheidung.

Wenn Ihr System über einen Client verfügt, der viele Nachrichten sendet (beispielsweise bei Dateiübertragungen), warten Sie nicht bei jeder Nachricht auf eine Serverantwort. Senden Sie stattdessen alle Nachrichten und prüfen Sie am Ende, ob alle empfangen wurden. Sie können auch die Methode der Servicequalität (Quality of Service, QoS) nutzen.

Sie können die Servicequalität je nach Nachricht variieren, indem unwichtige Nachrichten mit der Servicequalität 0 und wichtige Nachrichten mit der Servicequalität 2 übermittelt werden. Der Nachrichtendurchsatz kann bei der Servicequalität 0 etwa doppelt so hoch sein wie bei der Servicequalität 2.

## Namenskonventionen

Wenn Sie Ihre Anwendung für viele Clients entwerfen, implementieren Sie eine effektive Namenskonvention. Damit jedem Client die richtige Client-ID zugeordnet wird, verwenden Sie eine aussagekräftige Client-ID. Durch eine gute Namenskonvention lässt sich für den Administrator leichter feststellen, welche Clients gerade ausgeführt werden. Eine Namenskonvention ermöglicht dem Administrator das Filtern einer langen Liste mit Clients in WebSphere MQ Explorer und vereinfacht die Problembestimmung (siehe Client-ID).

## Durchsatz

Die Länge der Themennamen wirkt sich auf die Anzahl der Bytes aus, die über das Netz fließen. Beim Veröffentlichen oder Abonnieren kann die Anzahl der Bytes in einer Nachricht eine wichtige Rolle spielen. Begrenzen Sie daher die Anzahl der Zeichen in einem Themennamen. Wenn ein MQTT-Client ein Thema abonniert, gibt ihm WebSphere MQ einen Namen in folgendem Format:

```
ClientIdentifier:TopicName
```

Mit dem WebSphere MQ-Scriptbefehl **DISPLAY** können Sie alle Subskriptionen für einen MQTT-Client anzeigen:

```
DISPLAY SUB('ClientID1:*')
```

## Ressourcen in WebSphere MQ zur Verwendung durch MQTT-Clients definieren

Ein MQTT-Client stellt eine Verbindung zu einem fernen Warteschlangenmanager IBM WebSphere MQ her. Eine WebSphere MQ-Anwendung kann mit zwei grundlegenden Methoden Nachrichten an einen MQTT-Client senden: durch die Festlegung von SYSTEM.MQTT.TRANSMIT.QUEUE als Standardübertragungswarteschlange oder die Verwendung von Warteschlangenmanager-Aliasnamen. Definieren Sie bei sehr vielen MQTT-Clients die Standardübertragungswarteschlange eines Warteschlangenmanagers. Die Verwendung der Einstellung für die Standardübertragungswarteschlange vereinfacht den Verwaltungsaufwand; siehe Verteilte Steuerung von Warteschlangen zum Senden von Nachrichten an MQTT-Clients konfigurieren.

## Skalierbarkeit durch das Vermeiden von Subskriptionen verbessern

Wenn ein MQTT V3-Client ein Thema abonniert, wird vom Telemetrieservice (MQXR) in WebSphere MQ eine Subskription erstellt. Die Subskription leitet Veröffentlichungen für den Client an die Warteschlange SYSTEM.MQTT.TRANSMIT.QUEUE weiter. Der Name des fernen Warteschlangenmanagers im Übertragungsheader jeder Veröffentlichung wird auf die Client-ID des MQTT-Clients gesetzt, der die Subskription erstellt hat. Wenn viele Clients vorhanden sind, die alle ihre eigenen Subskriptionen erstellen, führt dies dazu, dass viele Proxy-Subskriptionen im gesamten Publish/Subscribe-Cluster oder in der Hierarchie von WebSphere MQ verwaltet werden müssen. Im Abschnitt [Nachricht direkt an einen Client senden](#) finden Sie Informationen darüber, wie Sie anstelle von Publish/Subscribe eine Punkt-zu-Punkt-basierte Lösung verwenden können.

## Sehr viele Clients verwalten

Wenn Sie viele gleichzeitig verbundene Clients unterstützen müssen, erhöhen Sie den Speicher, der für den Telemetrieservice (MQXR) zur Verfügung steht. Legen Sie hierfür die JVM-Parameter **-Xms** und **-Xmx** fest. Führen Sie folgende Schritte aus:

1. Suchen Sie die Datei `java.properties` im Konfigurationsverzeichnis des Telemetrieservice (siehe [Konfigurationsverzeichnis des Telemetrieservice \(MQXR\) unter Windows](#) bzw. [Konfigurationsverzeichnis des Telemetrieservice unter Linux](#)).
2. Folgen Sie den Anweisungen in der Datei; ein Heapspeicher von 1 GB reicht für 50.000 gleichzeitig verbundene Clients aus.

```
# Heap sizing options - uncomment the following lines to set the heap to 1G
#-Xmx1024m
#-Xms1024m
```

3. Fügen Sie in der Datei `java.properties` weitere Befehlszeilenargumente für die Übergabe an die Java Virtual Machine, auf der der Telemetrieservice (MQXR) ausgeführt wird, hinzu (siehe [JVM-Parameter an den Telemetrieservice \(MQXR\) übergeben](#)).

Wenn Sie die Anzahl der offenen Dateideskriptoren unter Linux erhöhen möchten, fügen Sie `/etc/security/limits.conf/` folgende Zeilen hinzu und melden Sie sich wieder an.

```
@mqm soft nofile 65000
@mqm hard nofile 65000
```

Für jeden Socket ist ein Dateideskriptor erforderlich. Da der Telemetrieservice einige zusätzliche Dateideskriptoren benötigt, muss diese Zahl größer als die Anzahl der erforderlichen offenen Sockets sein.

Der Warteschlangenmanager verwendet für jede nicht dauerhafte Subskription eine Objektkennung. Wenn Sie viele aktive, nicht dauerhafte Subskriptionen unterstützen möchten, erhöhen Sie die maximale Anzahl aktiver Kennungen im Warteschlangenmanager. Beispiel:

```
echo ALTER QMGR MAXHANDS(999999999) | runmqsc qMgrName
```

*Abbildung 26. Maximale Anzahl an Kennungen unter Windows ändern*

```
echo "ALTER QMGR MAXHANDS(999999999)" | runmqsc qMgrName
```

*Abbildung 27. Maximale Anzahl an Kennungen unter Linux ändern*

## Weitere Überlegungen

Berücksichtigen Sie bei der Planung Ihrer Systemvoraussetzungen den Zeitaufwand, der für den Neustart des Systems benötigt wird. Die geplante Nichtverfügbarkeit kann sich auf die Anzahl gestauter Nachrichten auswirken, die zur Verarbeitung anstehen. Konfigurieren Sie das System so, dass die Nachrichten in einer angemessenen Zeit erfolgreich verarbeitet werden können. Überprüfen Sie den Plattenspeicher, Hauptspeicher und die Verarbeitungskapazität. Bei einigen Clientanwendungen können bei einer Wiederherstellung der Clientverbindung möglicherweise Nachrichten gelöscht werden. Wenn Nachrichten

gelöscht werden sollen, legen Sie `CleanSession` in den Parametern der Clientverbindung fest (siehe [Sitzungen bereinigen](#)). Sie können in einem MQTT-Client auch Informationen mit der bestmöglichen Servicequalität 0 veröffentlichen; siehe [Servicequalität](#). Verwenden Sie beim Senden von Nachrichten aus WebSphere MQ nicht persistente Nachrichten. Nachrichten mit diesen Servicequalitäten werden bei einem Neustart des Systems oder Verbindung nicht wiederhergestellt.

## Von IBM WebSphere MQ Telemetry unterstützte Geräte

MQTT-Clients können auf einer ganzen Reihe von Geräten ausgeführt werden, von Sensoren über Aktuatoren bis hin zu Handbediengeräten und Fahrzeugsystemen.

MQTT-Clients sind klein und werden auf Geräten ausgeführt, die durch ihren geringen Speicher und niedrige Verarbeitungskapazität beschränkt sind. Das MQTT-Protokoll ist zuverlässig und verfügt über kompakte Header, wodurch es sich für Netze eignet, die durch eine geringe Bandbreite, hohen Aufwand und nicht unterbrechungsfreie Verfügbarkeit beschränkt sind.

IBM WebSphere MQ Telemetry stellt drei Clients bereit, die alle das MQTT v3-Protokoll implementieren:

- Ein Java-Client, der in allen Varianten von Java von der kleinsten CLDC (Connected Limited Device Configuration) / MIDP (Mobile Information Device Profile) über CDC (Connected Device Configuration) / Foundation, J2SE (Java Platform, Standard Edition) und J2EE (Java Platform, Enterprise Edition) ausgeführt werden kann. Die angepasste `javaRM`-Klassenbibliothek von IBM wird ebenfalls unterstützt.
- Eine C-Referenzimplementierung zusammen mit einem vordefinierten nativen Client für Windows - und Linux -Systeme. Die Referenzimplementierung in der Programmiersprache C ermöglicht die Portierung von MQTT auf eine breite Palette an Geräten und Plattformen.
- Der erweiterte Client, IBM WebSphere MQ Telemetry Daemon for Devices, der in C geschrieben ist und auf jeder geeigneten Linux - oder Windows -Plattform ausgeführt werden kann.

Einige Windows -Systeme unter Intel, einschließlich Windows XP, RedHat, Ubuntu und einige Linux -Systeme auf ARM-Plattformen wie Eurotech Viper implementieren Versionen von Linux , die den C-Client ausführen, aber IBM bietet keine Serviceunterstützung für die Plattformen. Sie müssen Probleme im Zusammenhang mit dem Client auf einer unterstützten Plattform reproduzieren, wenn Sie sich an Ihr IBM Support Center wenden möchten.

Die Java ME-Plattform wird im Allgemeinen auf kleinen Geräten wie Aktuatoren, Sensoren, Mobiltelefonen und anderen eingebetteten Geräten verwendet. Die Java-SE-Plattform wird im Allgemeinen auf integrierten Geräten mit höheren Endgeräten wie Desktop-Computern und Servern installiert.

**Anmerkung:** Das [Eclipse Paho-Projekt](#) und [MQTT.org](#) verfügen über kostenlose Downloads der neuesten Telemetrieclients und Beispiele für eine Reihe von Programmiersprachen.

### Zugehörige Konzepte

„Telemetry-Dämon für Geräte“ auf Seite 104

Der WebSphere MQ Telemetry-Dämon für Geräte ist eine erweiterte MQTT V3-Clientanwendung. Damit können Sie Nachrichten von anderen MQTT-Clients speichern und weiterleiten (Store-and-forward-Verfahren). Der Dämon stellt wie ein MQTT-Client eine Verbindung zu WebSphere MQ her, Sie können jedoch auch andere MQTT-Clients mit ihm verbinden. Sie können ihn außerdem mit anderen Telemetriedämonen verbinden.

## Objekte verwalten

Die Eigenschaften bzw. *Attribute* von IBM WebSphere MQ-Objekten werden durch Warteschlangenmanager festgelegt. Die Werte dieser Attribute bestimmen, wie diese Objekte von WebSphere MQ verarbeitet werden. In den Anwendungen werden Objekte über die MQI (Message Queue Interface) gesteuert. Programme verweisen auf Objekte anhand eines *MQ-Objektdeskriptors* (MQOD).

Wenn Sie eine Objektverwaltungsoperation mit einem WebSphere MQ-Befehl ausführen, beispielsweise ein Objekt definieren, ändern oder löschen, prüft der Warteschlangenmanager, ob Sie über die entsprechende Berechtigung zur Ausführung dieses Vorgangs verfügen. Ebenso überprüft der Warteschlangenmanager auch bei Anwendungen, die versuchen, ein Objekt mit einem MQOPEN-Aufruf zu öffnen, ob

diese über die entsprechenden Berechtigungen verfügen, bevor der Zugriff auf das Objekt gestattet wird. Dabei werden die Prüfungen für den Namen des Objekts vorgenommen, das geöffnet werden soll.

Die Objektverwaltung beinhaltet folgende Tasks:

- Warteschlangenmanager starten und stoppen
- Objekte, insbesondere Warteschlangen, für Anwendungen erstellen
- Die Arbeit mit Kanälen, um Kommunikationspfade zu Warteschlangenmanagern auf anderen (fernen) Systemen zu erstellen.
- *Warteschlangenmanager-Cluster* erstellen, um die Verwaltung zu vereinfachen und die Arbeitslast gleichmäßig zu verteilen.

Eine Übersicht über die Methoden der Erstellung und Verwaltung von WebSphere MQ-Objekten finden Sie im Abschnitt [„Objekte verwalten“](#) auf Seite 135.

Weitere Informationen zu den Objektverwaltungstasks finden Sie in den folgenden Abschnitten:

## Objekte

Bei vielen Verwaltungstasks werden verschiedene Arten von IBM WebSphere MQ-Objekten bearbeitet.

Informationen zur Benennung von IBM WebSphere MQ-Objekten finden Sie im Abschnitt [„IBM WebSphere MQ-Objekte benennen“](#) auf Seite 131.

Informationen zu den auf einem Warteschlangenmanager erstellten Standardobjekten finden Sie im Abschnitt [„Systemstandardwertobjekte“](#) auf Seite 137.

Informationen zu den verschiedenen Arten von IBM WebSphere MQ-Objekten finden Sie in den folgenden Abschnitten:

### Zugehörige Konzepte

[„Einführung in die Nachrichtenwarteschlangensteuerung“](#) auf Seite 45

Mit den WebSphere MQ-Produkten können Programme über ein Netz ungleicher Komponenten (Prozessoren, Betriebssysteme, Subsysteme und Kommunikationsprotokolle) unter Verwendung einer konsistenten Anwendungsprogrammierschnittstelle miteinander kommunizieren.

[„Objektattribute“](#) auf Seite 136

Die Eigenschaften eines Objektes werden über die Objektattribute definiert. Einige dieser Attribute können Sie angeben, andere hingegen können nur angezeigt werden.

### Zugehörige Verweise

[Die MQSC-Befehle](#)

## Warteschlangen

In diesem Abschnitt finden Sie einführende Informationen zu WebSphere MQ-Warteschlangen und -Warteschlangenattributen.

Eine WebSphere MQ-*Warteschlange* ist ein benanntes Objekt, in das Anwendungen Nachrichten einreihen und aus dem Anwendungen Nachrichten abrufen können.

Nachrichten werden in einer Warteschlange gespeichert. Wenn also die Anwendung, welche die Nachricht einreicht, eine Antwort auf ihre Nachricht erwartet, kann sie währenddessen andere Arbeiten erledigen. Anwendungen greifen über die im Abschnitt [Übersicht über das Message Queue Interface](#) beschriebene Message Queue Interface auf eine Warteschlange zu.

Bevor Nachrichten in eine Warteschlange eingereicht werden können, muss diese bereits erstellt sein. Eine Warteschlange gehört zu einem Warteschlangenmanager, der wiederum Eigner vieler Warteschlangen sein kann. Die Namen der einzelnen Warteschlangen müssen allerdings innerhalb des betreffenden Warteschlangenmanagers eindeutig sein.

Eine Warteschlange wird von einem Warteschlangenmanager verwaltet. In den meisten Fällen werden die einzelnen Warteschlangen von dem jeweiligen Warteschlangenmanager physisch verwaltet, ohne dass dies den Anwendungsprogrammen offensichtlich ist. Gemeinsam genutzte Warteschlangen in WebSphere



MQ for z/OS können von jedem Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange verwaltet werden.

Warteschlangen können mithilfe von WebSphere MQ-Befehlen (MQSC), PCF-Befehlen oder plattformspezifischen Schnittstellen wie den Betriebs- und Steuerkonsolen von WebSphere MQ for z/OS erstellt werden.

Lokale Warteschlangen für temporäre Jobs können *dynamisch* von der Anwendung aus erstellt werden. So können beispielsweise *Warteschlangen für zu beantwortende Nachrichten* erstellt werden (die nach Beendigung der Anwendung nicht mehr benötigt werden). Weitere Informationen finden Sie in „Dynamische und Modellwarteschlangen“ auf Seite 119.

Bevor Sie eine Warteschlange verwenden können, müssen Sie sie zunächst öffnen und dabei angeben, wofür die Warteschlange vorgesehen ist. So können Warteschlangen beispielsweise für folgende Zwecke geöffnet werden:

- Ausschließlich zum Durchsuchen (nicht zum Abrufen) von Nachrichten
- Zum Abrufen von Nachrichten (mit gemeinsamem Zugriff mit anderen Programmen oder mit exklusivem Zugriff)
- Zum Einreihen von Nachrichten in die Warteschlange
- Zur Abfrage der Warteschlangenattribute
- Zur Definition der Warteschlangenattribute

Eine vollständige Liste der Optionen, die beim Öffnen einer Warteschlange angegeben werden können, finden Sie im Abschnitt MQOPEN - Objekt öffnen.

## Warteschlangenattribute

Einige Warteschlangenattribute werden bei der Definition der Warteschlange angegeben und können später nicht mehr geändert werden (z. B. der Warteschlangentyp). Andere Warteschlangenattribute können danach eingeteilt werden, von wem sie geändert werden können:

- Attribute, die vom Warteschlangenmanager während der Warteschlangenverarbeitung geändert werden können (z. B. die aktuelle Warteschlangenlänge)
- Attribute, die nur über Befehle geändert werden können (z. B. die Textbeschreibung der Warteschlange)
- Attribute, die von Anwendungen unter Verwendung des Aufrufs MQSET geändert werden können (z. B. ob Einreihvorgänge in die Warteschlange erlaubt sein sollen).

Die Werte aller Attribute können mit dem Aufruf MQINQ abgerufen werden.

Folgende Attribute werden für mehrere Warteschlangentypen verwendet:

### **QName**

Den Namen der Warteschlange

### **QType**

Der Warteschlangentyp

### **QDesc**

Textbeschreibung der Warteschlange

### **InhibitGet**

Ob Programme Nachrichten aus der Warteschlange abrufen dürfen (allerdings ist es unter keinen Umständen möglich, Nachrichten aus fernen Warteschlangen abzurufen)

### **InhibitPut**

Ob Programme zum Einreihen von Nachrichten in die Warteschlange berechtigt sein sollen

### **DefPriority**

Die Standardpriorität für Nachrichten, die in die Warteschlange eingereiht werden

### **DefPersistence**

Die Standardpersistenz für Nachrichten, die in die Warteschlange eingereiht werden.

### **Scope (wird unter z/OS) nicht unterstützt)**

Steuert, ob auch in einem Namensservice ein Eintrag für diese Warteschlange vorhanden ist

Eine umfassende Beschreibung dieser Attribute finden Sie im Abschnitt Attribute für Warteschlangen.

### **Zugehörige Konzepte**

„Ferne Warteschlangen“ auf Seite 116

Eine Warteschlange ist für ein Programm *fern*, wenn sie zu einem Warteschlangenmanager gehörig ist, mit dem das Programm nicht verbunden ist.

„Aliaswarteschlangen“ auf Seite 116

Eine *Aliaswarteschlange* ist ein WebSphere MQ-Objekt, mit dessen Hilfe der Zugriff auf eine andere Warteschlange oder ein Thema möglich ist. Somit können mehrere Programme mit derselben Warteschlange arbeiten, indem sie für den Zugriff jeweils einen anderen Namen verwenden.

„Warteschlangen definieren“ auf Seite 121

Warteschlangen können in IBM WebSphere MQ über den WebSphere MQ-Scriptbefehl DEFINE oder den PCF-Befehl 'Create Queue' definiert werden.

„Von IBM WebSphere MQ verwendete Warteschlangen“ auf Seite 121

In IBM WebSphere MQ werden einige lokale Warteschlangen für bestimmte Aspekte des Betriebsablaufs verwendet.

### **Zugehörige Tasks**

Referenzinformationen zum Entwickeln von Anwendungen

### **Zugehörige Verweise**

Die MQSC-Befehle

„Lokale Warteschlangen“ auf Seite 114

Übertragungs-, Initialisierungs-, Befehls-, Standard-, Kanal- und Ereigniswarteschlangen sowie Warteschlangen für nicht zustellbare Nachrichten sind verschiedene Typen lokaler Warteschlangen.

„Gemeinsam genutzte Warteschlangen und Clusterwarteschlangen“ auf Seite 117

In diesem Abschnitt werden die Begriffe 'Gemeinsam genutzte Warteschlange' und 'Clusterwarteschlange' definiert und erläutert und anschließend beide Konzepte miteinander verglichen.

„Dynamische und Modellwarteschlangen“ auf Seite 119

In diesem Abschnitt erhalten Sie einen Einblick in dynamische Warteschlangen, Sie finden hier Erläuterungen zu den Eigenschaften temporärer und permanenter dynamischer Warteschlangen, zur Verwendung dynamischer Warteschlangen und zu dabei besonders zu berücksichtigenden Aspekten sowie zu Modellwarteschlangen.

### *Lokale Warteschlangen*

Übertragungs-, Initialisierungs-, Befehls-, Standard-, Kanal- und Ereigniswarteschlangen sowie Warteschlangen für nicht zustellbare Nachrichten sind verschiedene Typen lokaler Warteschlangen.

Eine Warteschlange gilt für ein Programm als *lokal*, wenn sie dem Warteschlangenmanager zugeordnet ist, mit dem das Programm verbunden ist. Nachrichten können aus lokalen Warteschlangen abgerufen und darin eingereiht werden.

Das Warteschlangendefinitionsobjekt enthält die Definitionsinformationen der Warteschlange sowie die in die Warteschlange eingereihten physischen Nachrichten.

Jeder Warteschlangenmanager kann über einige lokale Warteschlangen verfügen, die für spezielle Zwecke eingesetzt werden:

### **Übertragungswarteschlangen**

Wenn eine Anwendung eine Nachricht an eine ferne Warteschlange sendet, speichert der lokale Warteschlangenmanager die Nachricht in einer speziellen lokalen Warteschlange, der sogenannten *Übertragungswarteschlange*.

Ein *Nachrichtenkanalagent* ist ein der Übertragungswarteschlange zugeordnetes Kanalprogramm, das die Nachricht an das nächste Ziel übermittelt. Bei dem nächsten Ziel handelt es sich um den War-

teschlangenmanager, mit dem der Nachrichtenkanal verbunden ist. Dieser Warteschlangenmanager muss nicht zwangsläufig das endgültige Ziel der Nachricht darstellen. Nach ihrer Übermittlung an das nächste Ziel wird die Nachricht aus der Übertragungswarteschlange gelöscht. Möglicherweise muss die Nachricht auf ihrem Weg zum Zielort viele Warteschlangenmanager durchlaufen. Auf jedem Warteschlangenmanager entlang der Route muss eine Übertragungswarteschlange definiert werden. Diese enthalten jeweils Nachrichten, die auf die Übertragung zum nächsten Ziel warten. Eine normale Übertragungswarteschlange enthält Nachrichten für das nächste Ziel, die jedoch nicht alle denselben endgültigen Zielort haben müssen. In einer Clusterübertragungswarteschlange sind Nachrichten für mehrere Ziele enthalten. Die Korrelations-ID (`correlID`) der einzelnen Nachrichten gibt den Kanal an, in den die Nachricht zur Übertragung an das nächste Ziel gestellt wird.

An einem Warteschlangenmanager können mehrere Übertragungswarteschlangen definiert werden. Sie können auch mehrere Übertragungswarteschlangen für dasselbe Ziel definieren, die jeweils für eine andere Berechtigungsklasse verwendet werden. So ist es unter Umständen sinnvoll, für kurze und lange Nachrichten an dasselbe Ziel unterschiedliche Übertragungswarteschlangen zu erstellen. Die Nachrichten können dann über verschiedene Nachrichtenkanäle übertragen werden, sodass die kürzeren Nachrichten nicht von den längeren aufgehalten werden. Auf anderen Plattformen als z/OS werden Nachrichten standardmäßig in die einzelne Clusterübertragungswarteschlange `SYSTEM.CLUSTER.TRANSMIT.QUEUE` eingereiht. Auf diesen anderen Plattformen haben Sie die Möglichkeit, die Standardeinstellung zu ändern und die Nachrichtenübertragung an verschiedene Clusterwarteschlangenmanager auf mehrere Clusterübertragungswarteschlangen aufzuteilen. Wenn Sie das Warteschlangenmanagerattribut `DEFCLXQ` auf `CHANNEL` setzen, erstellt jeder Clustersenderkanal eine separate Clusterübertragungswarteschlange. Eine weitere Option auf den anderen Plattformen ist die manuelle Definition von Clusterübertragungswarteschlangen für Clustersenderkanäle.

Übertragungswarteschlangen können einen Nachrichtenkanalagenten veranlassen, Nachrichten weiterzusenden; siehe [WebSphere MQ -Anwendungen mit Auslösern starten](#).

### **Initialisierungswarteschlangen**

Eine *Initialisierungswarteschlange* ist eine lokale Warteschlange, in die der Warteschlangenmanager eine Auslösenachricht einreicht, wenn in einer Anwendungswarteschlange ein Auslöserereignis eintritt.

Ein Auslöserereignis ist ein Ereignis, das ein Programm dazu veranlassen soll, die Verarbeitung einer Warteschlange zu starten. Bei einem solchen Ereignis könnte es sich beispielsweise um den Eingang von mehr als 10 Nachrichten handeln. Weitere Informationen zur Funktionsweise der Auslösefunktion finden Sie unter [WebSphere MQ -Anwendungen mit Auslösern starten](#).

### **Warteschlange für nicht zustellbare Nachrichten**

Eine *Warteschlange für nicht zustellbare Nachrichten* ist eine lokale Warteschlange, in die der Warteschlangenmanager Nachrichten einreicht, die nicht zugestellt werden können.

Wenn der Warteschlangenmanager eine Nachricht in die Warteschlange für nicht zustellbare Nachrichten stellt, fügt er ihr einen Header hinzu. In den Headerinformationen ist unter anderem auch angegeben, warum die Nachricht vom Warteschlangenmanager in die Warteschlange für nicht zustellbare Nachrichten gestellt wurde. Darüber hinaus sind der Zielort der ursprünglichen Nachricht sowie der Zeitpunkt (Datum und Uhrzeit) aufgeführt, an dem der Warteschlangenmanager die Nachricht in die Warteschlange für nicht zustellbare Nachrichten eingereiht hat.

Auch Anwendungen können die Warteschlange für Nachrichten verwenden, die sie nicht zustellen können. Weitere Informationen finden Sie im Abschnitt [Verwendung der Warteschlange für nicht zustellbare Nachrichten](#).

### **Systembefehlswarteschlange**

Eine *Systembefehlswarteschlange* ist eine Warteschlange, an die entsprechend berechnete Anwendungen WebSphere MQ-Befehle senden können. Je nachdem, welche Befehle auf Ihrer Plattform unterstützt werden, empfangen diese Warteschlangen PCF-, MQSC- und CL-Befehle, die dann vom Warteschlangenmanager bearbeitet werden können.

## Systemstandardwarteschlangen

In den *Systemstandardwarteschlangen* sind die ursprünglichen Definitionen der Warteschlangen für Ihr System enthalten. Wenn Sie eine Warteschlangendefinition erstellen, wird die Definition vom Warteschlangenmanager aus der entsprechenden Systemstandardwarteschlange kopiert. Die Vorgehensweise beim Erstellen einer Warteschlangendefinition ist anders als beim Erstellen einer dynamischen Warteschlange. Die Definition der dynamischen Warteschlange basiert auf der als Schablone für die dynamische Warteschlange ausgewählten Modellwarteschlange.

## Ereigniswarteschlangen

In *Ereigniswarteschlangen* sind Ereignisnachrichten enthalten. Diese Nachrichten werden vom Warteschlangenmanager oder einem Kanal gemeldet.

### Ferne Warteschlangen

Eine Warteschlange ist für ein Programm *fern*, wenn sie zu einem Warteschlangenmanager gehörig ist, mit dem das Programm nicht verbunden ist.

Wenn eine Kommunikationsverbindung eingerichtet ist, kann ein Programm eine Nachricht an eine ferne Warteschlange senden. Nicht möglich ist es dagegen, dass ein Programm eine Nachricht aus einer fernen Warteschlange abrufen.

In dem bei der Definition einer fernen Warteschlange erstellten Warteschlangendefinitionsobjekt sind nur die Informationen enthalten, die erforderlich sind, damit der lokale Warteschlangenmanager die Warteschlange, an welche die Nachricht gehen soll, ausfindig machen kann. Dieses Objekt wird auch als *lokale Definition einer fernen Warteschlange* bezeichnet. Alle Attribute der fernen Warteschlange sind auf dem Warteschlangenmanager gespeichert, der Eigner der Warteschlange ist, da es sich für diesen Warteschlangenmanager um eine lokale Warteschlange handelt.

Beim Öffnen einer fernen Warteschlange ist zur Identifikation der Warteschlange eine der folgenden Angaben erforderlich:

- Der Name der lokalen Definition der fernen Warteschlange.

Um eine lokale Definition einer fernen Warteschlange zu erstellen, verwenden Sie den Befehl `DEFINE QREMOTE`; verwenden Sie unter WebSphere MQ für IBM iden Befehl `CRTMQMQ`.

Vom Blickpunkt einer Anwendung aus besteht kein Unterschied zum Öffnen einer lokalen Warteschlange. Für die Anwendung ist es nicht relevant, ob es sich um eine lokale oder eine ferne Warteschlange handelt.

- Der Name des fernen Warteschlangenmanagers und der Name, unter dem dieser ferne Warteschlangenmanager die Warteschlange kennt.

Lokale Definitionen ferner Warteschlangen umfassen zusätzlich zu den im Abschnitt „Warteschlangenattribute“ auf Seite 113 beschriebenen allgemeinen Attributen drei weitere Attribute. Dies sind *RemoteQName* (der Name, den der Warteschlangenmanager, der Eigner der Warteschlange ist, kennt), *RemoteQMGrName* (der Name des Warteschlangenmanagers, der Eigner der Warteschlange ist) und *XmitQName* (der Name der lokalen Übertragungswarteschlange, die beim Weiterleiten von Nachrichten an andere Warteschlangenmanager verwendet wird). Eine detailliertere Beschreibung dieser Attribute finden Sie im Abschnitt Attribute für Warteschlangen.

Wenn Sie den Aufruf `MQINQ` für die lokale Definition einer fernen Warteschlange verwenden, gibt der Warteschlangenmanager nur die Attribute der lokalen Definition zurück, also die Namen der fernen Warteschlange, des fernen Warteschlangenmanagers und der Übertragungswarteschlange. Die Attribute der entsprechenden lokalen Warteschlange im fernen System werden nicht zurückgegeben.

Weitere Informationen finden Sie auch im Abschnitt Übertragungswarteschlangen.

### Aliaswarteschlangen

Eine *Aliaswarteschlange* ist ein WebSphere MQ-Objekt, mit dessen Hilfe der Zugriff auf eine andere Warteschlange oder ein Thema möglich ist. Somit können mehrere Programme mit derselben Warteschlange arbeiten, indem sie für den Zugriff jeweils einen anderen Namen verwenden.

Die aus der Auflösung eines Aliasnamens resultierende Warteschlange (die sogenannten Basiswarteschlange) kann eine lokale Warteschlange, die lokale Definition einer fernen Warteschlange oder eine gemeinsam genutzte Warteschlange (eine nur in WebSphere MQ for z/OS verfügbare Art von lokaler Warteschlange) sein. Auch kann es sich je nach Plattformunterstützung um eine vordefinierte oder eine dynamische Warteschlange handeln.

Ein Aliasname kann auch in ein Thema aufgelöst werden. Wenn eine Anwendung gerade Nachrichten in eine Warteschlange einreicht, kann der Warteschlangennamenname zu einem Alias für ein Thema gemacht und die Anwendung auf diese Weise zu Veröffentlichungen zu dem Thema gebracht werden. Hierfür muss der Anwendungscode nicht geändert werden.

**Anmerkung:** Ein Aliasname kann nicht in eine andere lokal definierte Aliaswarteschlange aufgelöst werden.

Ein Beispiel für die Verwendung von Aliaswarteschlangen: Ein Systemadministrator erteilt für den Basiswarteschlangennamen (also die Warteschlange, in die der Aliasname aufgelöst wird) und den Aliaswarteschlangennamen verschiedene Zugriffsberechtigungen. Es kann dann passieren, dass Programme oder Benutzer berechtigt sind, die Aliaswarteschlange zu verwenden, nicht jedoch die Basiswarteschlange.

Alternativ dazu kann die Berechtigung so definiert werden, dass Einreihoperationen für die Basiswarteschlange erlaubt, für den Aliasnamen dagegen nicht zulässig sind.

Bei manchen Anwendungen bietet sich Systemadministratoren durch die Verwendung von Aliaswarteschlangen die Möglichkeit, ganz einfach die Definition eines Aliaswarteschlangenobjekts zu ändern, ohne dass eine Änderung der Anwendung erforderlich ist.

Wenn Programme versuchen, den Aliasnamen zu verwenden, nimmt WebSphere MQ Berechtigungsprüfungen für diesen Namen vor. Dabei wird nicht geprüft, ob das Programm auch zum Zugriff auf den Namen berechtigt ist, in den der Aliasname aufgelöst wird. Es kann also sein, dass ein Programm zum Zugriff auf einen Aliaswarteschlangennamen berechtigt ist, jedoch über keine Berechtigung für den aufgelösten Warteschlangennamen verfügt.

Neben den allgemeinen Warteschlangenattributen, die in „Warteschlangen“ auf Seite 112 beschrieben werden, verfügen Aliaswarteschlangen über ein Attribut *BaseQName*. Hierbei handelt es sich um den Basiswarteschlangennamen, in den der Aliasname aufgelöst wird. Eine detailliertere Beschreibung dieser Attribute finden Sie im Abschnitt *BaseQName (MQCHAR48)*.

Die Attribute *InhibitGet* und *InhibitPut* (siehe „Warteschlangen“ auf Seite 112) von Aliaswarteschlangen gehören zum Aliasnamen. Wird beispielsweise der Aliaswarteschlangennamenname ALIAS1 in den Basiswarteschlangennamen BASE aufgelöst, gelten Beschränkungen für ALIAS1 nur für ALIAS1 und nicht für BASE. Beschränkungen für BASE dagegen gelten auch für ALIAS1.

Die Attribute *DefPriority* und *DefPersistence* gehören ebenfalls zum Aliasnamen. Somit können Sie beispielsweise verschiedenen Aliassen derselben Basiswarteschlange unterschiedliche Standardprioritäten zuordnen. Diese Prioritäten können außerdem geändert werden, ohne dass eine Änderung der Anwendungen, welche die Aliasse verwenden, erforderlich ist.

#### *Gemeinsam genutzte Warteschlangen und Clusterwarteschlangen*

In diesem Abschnitt werden die Begriffe 'Gemeinsam genutzte Warteschlange' und 'Clusterwarteschlange' definiert und erläutert und anschließend beide Konzepte miteinander verglichen.

## **Gemeinsam genutzte Warteschlangen**

Eine *gemeinsam genutzte Warteschlange* ist eine bestimmte Art von lokaler Warteschlange, auf deren Nachrichten ein oder mehrere Warteschlangenmanager aus einer Gruppe mit gemeinsamer Warteschlange zugreifen können. **Gemeinsam genutzte Warteschlangen sind nur in WebSphere MQ for z/OS verfügbar.** (Dies ist nicht zu verwechseln mit einer Warteschlange, die von mehreren Anwendungen *gemeinsam genutzt* wird, die denselben Warteschlangenmanager verwenden. Gemeinsam genutzte Warteschlangen befinden sich in einer Coupling-Facility. Alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange können auf diese Warteschlangen zugreifen. Die Namen der einzelnen

gemeinsam genutzten Warteschlangen müssen innerhalb der Gruppe mit gemeinsamer Warteschlange eindeutig sein.

## Clusterwarteschlangen

Eine Clusterwarteschlange wird von einem Clusterwarteschlangenmanager anderen Warteschlangenmanagern im Cluster zur Verfügung gestellt.

Der Cluster-Warteschlangenmanager erstellt eine lokale Definition für die Warteschlange, in der auch der Name des Clusters angegeben ist, in dem die Warteschlange zur Verfügung gestellt werden soll. Mit dieser Definition wird die Warteschlange den anderen Warteschlangenmanagern im Cluster bekannt gemacht. Die anderen Warteschlangenmanager im Cluster können ohne entsprechende Definition einer fernen Warteschlange Nachrichten in eine Clusterwarteschlange einreihen. Eine Clusterwarteschlange kann in mehreren Clustern bekannt gemacht werden. Weitere Informationen finden Sie in den Abschnitten [Cluster](#) und [Warteschlangenmanager-Cluster konfigurieren](#).

## Vergleich zwischen gemeinsam genutzten Warteschlangen und Clusterwarteschlangen

Diese Informationen sollen Ihnen helfen, gemeinsam genutzte Warteschlangen und Clusterwarteschlangen miteinander zu vergleichen und dann zu entscheiden, welche für Ihr System besser geeignet sind.

## Aufwand für Verlagerungsfunktion

In Clusterwarteschlangen werden Nachrichten von der Verlagerungsfunktion gesendet, sodass zum Anwendungsaufwand noch der Verlagerungsaufwand hinzukommen. Es entsteht ein zusätzlicher Aufwand im Netz, weil Kanäle Nachrichten abrufen und einreihen. Dieser Aufwand entsteht bei gemeinsam genutzten Warteschlangen nicht, die deshalb bei der Verlagerung von Nachrichten zwischen Warteschlangenmanagern in einer Gruppe mit gemeinsamer Warteschlange weniger Verarbeitungskapazität in Anspruch nehmen.

## Verfügbarkeit von Nachrichten

Wird eine Nachricht in eine Warteschlange eingereicht, senden Clusterwarteschlangen die Nachricht an einen der Warteschlangenmanager mit aktiven Kanälen, die mit Ihrem Warteschlangenmanager verbunden sind. Wenn auf dem fernen Warteschlangenmanager Anwendungen, die zur Verarbeitung der Nachrichten verwendet werden, nicht aktiv sind, werden die Nachrichten nicht verarbeitet, sondern warten darauf, dass die Anwendung gestartet wird. Wurde ein Warteschlangenmanager heruntergefahren, gilt entsprechend, dass Nachrichten auf dem Warteschlangenmanager erst zur Verfügung gestellt werden, nachdem der Warteschlangenmanager erneut gestartet wurde. Diese Instanzen zeigen eine schlechtere Nachrichtenverfügbarkeit als gemeinsam genutzte Warteschlangen.

Bei der Verwendung gemeinsam genutzter Warteschlangen kann jede Anwendung in der Gruppe mit gemeinsamer Warteschlange Nachrichten, die gesendet werden, abrufen. Wird einer der Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange heruntergefahren, sind die Nachrichten für die anderen Warteschlangenmanager verfügbar, sodass eine höhere Nachrichtenverfügbarkeit als bei der Verwendung von Clusterwarteschlangen erreicht wird.

## Kapazität

Eine Coupling-Facility ist kostenintensiver als ein Datenträger. Deshalb sind die Kosten für die Speicherung von 1.000.000 Nachrichten in einer lokalen Warteschlange niedriger als für eine Coupling-Facility, die über ausreichend Kapazität zur Speicherung derselben Anzahl Nachrichten verfügt.

## Nachrichten an andere Warteschlangenmanager senden

Nachrichten in einer gemeinsam genutzten Warteschlange sind nur innerhalb einer Gruppe mit gemeinsamer Warteschlange verfügbar. Wenn Sie einen Warteschlangenmanager außerhalb der Gruppe mit

gemeinsamer Warteschlange verwenden möchten, müssen Sie die Verlagerungsfunktion nutzen. Für einen Lastausgleich zwischen mehreren fernen verteilten Warteschlangenmanagern können Sie Clustering einsetzen.

## Lastausgleich

Mithilfe des Clusterings können Sie eine Gewichtung festlegen, nach der gesendete Nachrichten auf Kanäle und Warteschlangenmanager verteilt werden sollen. Sie können beispielsweise festlegen, dass 60% der Nachrichten an einen bestimmten Warteschlangenmanager gesendet werden sollen und 40% der Nachrichten an einen anderen. Diese Instanz ist, was die Verarbeitung betrifft, nicht von der Verfügbarkeit des fernen Warteschlangenmanagers abhängig. Selbst wenn das System mit dem ersten Warteschlangenmanager überlastet und das System mit dem zweiten Warteschlangenmanager unausgelastet sein sollten, geht der größere Anteil der Nachrichten weiterhin an den ersten Warteschlangenmanager.

Bei gemeinsam genutzten Warteschlangen können zwei CICS-Systeme Nachrichten abrufen. Ist ein System überlastet, kann das andere System einen Großteil der Arbeitslast übernehmen.

### *Dynamische und Modellwarteschlangen*

In diesem Abschnitt erhalten Sie einen Einblick in dynamische Warteschlangen, Sie finden hier Erläuterungen zu den Eigenschaften temporärer und permanenter dynamischer Warteschlangen, zur Verwendung dynamischer Warteschlangen und zu dabei besonders zu berücksichtigenden Aspekten sowie zu Modellwarteschlangen.

Wenn ein Anwendungsprogramm einen Aufruf MQOPEN absetzt, um eine Modellwarteschlange zu öffnen, erstellt der Warteschlangenmanager mit den Attributen der Modellwarteschlange dynamisch eine Instanz einer lokalen Warteschlange. Abhängig vom Wert des Felds *DefinitionType* der Modellwarteschlange erstellt der Warteschlangenmanager entweder eine temporäre oder eine permanente dynamische Warteschlange (siehe [Dynamische Warteschlangen erstellen](#)).

### **Eigenschaften temporärer dynamischer Warteschlangen**

*Temporäre dynamische Warteschlangen* weisen folgende Eigenschaften auf:

- Die Warteschlangen können nicht gemeinsam genutzt werden, es ist also kein Zugriff von Warteschlangenmanagern aus einer Gruppe mit gemeinsamer Warteschlange möglich.
  - Gruppen mit gemeinsamer Warteschlange sind nur in WebSphere MQ for z/OS verfügbar.
- Sie enthalten nur nicht persistente Nachrichten.
- Sie sind nicht wiederherstellbar.
- Beim Start des Warteschlangenmanagers werden sie gelöscht.
- Sie werden auch gelöscht, wenn die Anwendung, die den Aufruf MQOPEN zum Erstellen der Warteschlange ausgegeben hat, die Warteschlange schließt oder selbst beendet wird.
  - Eventuell vorhandene festgeschriebene Nachrichten in der Warteschlange werden gelöscht.
  - Stehen zu diesem Zeitpunkt nicht festgeschriebene Aufrufe MQGET, MQPUT oder MQPUT1 für die Warteschlange aus, wird die Warteschlange als logisch gelöscht markiert und wird erst physisch gelöscht (nach Festschreibung dieser Aufrufe), wenn der Vorgang zum Schließen der Warteschlange verarbeitet oder die Anwendung beendet wird.
  - Wird die Warteschlange zu diesem Zeitpunkt gerade verwendet (von der erstellenden oder einer anderen Anwendung), wird sie als logisch gelöscht markiert und erst physisch gelöscht, wenn sie von der letzten Anwendung, welche die Warteschlange verwendet, geschlossen wird.
  - Zugriffsversuche auf eine logisch gelöschte Warteschlange schlagen mit dem Ursachencode MQRC\_Q\_DELETED fehl (es sei denn, der Zugriff erfolgt zum Schließen der Warteschlange).
  - MQCO\_NONE, MQCO\_DELETE und MQCO\_DELETE\_PURGE werden alle als MQCO\_NONE behandelt, wenn sie in einem Aufruf MQCLOSE zu dem entsprechenden Aufruf MQOPEN angegeben sind, mit dem die Warteschlange erstellt wurde.

### **Eigenschaften permanenter dynamischer Warteschlangen**

*Permanente dynamische Warteschlangen* weisen folgende Eigenschaften auf:

- Sie enthalten persistente und nicht persistente Nachrichten.
- Im Falle eines Systemfehlers sind sie wiederherstellbar.
- Sie werden gelöscht, wenn sie von einer Anwendung mit der Option MQCO\_DELETE oder MQCO\_DELETE\_PURGE erfolgreich geschlossen werden. (Bei dieser Anwendung muss es sich nicht zwangsläufig um die Anwendung handeln, die den Aufruf MQOPEN zum Erstellen der Warteschlange ausgegeben hat.)
  - Falls sich noch (festgeschriebene oder nicht festgeschriebene) Nachrichten in der Warteschlange befinden, schlägt eine Schließenanforderung mit der Option MQCO\_DELETE fehl. Eine Schließenanforderung mit der Option MQCO\_DELETE\_PURGE kann selbst dann erfolgreich ausgeführt werden, wenn sich festgeschriebene Nachrichten in der Warteschlange befinden (wobei die Nachrichten im Rahmen des Schließvorgangs gelöscht werden), sie schlägt jedoch fehl, wenn für die Warteschlange nicht festgeschriebene Aufrufe MQGET, MQPUT oder MQPUT1 ausstehen.
  - Wenn die Löschanforderung erfolgreich ist, die Warteschlange jedoch gerade (von der erstellenden oder einer anderen Anwendung) verwendet wird, wird sie als logisch gelöscht markiert und erst physisch gelöscht, wenn sie von der letzten Anwendung, die die Warteschlange verwendet, geschlossen wird.
- Die Warteschlangen werden nicht gelöscht, wenn sie von einer Anwendung geschlossen werden, die nicht zum Löschen der Warteschlange berechtigt ist, es sei denn, diese Anwendung hat den Aufruf MQOPEN zum Erstellen der Warteschlange ausgegeben. Für die Benutzer-ID (bzw. bei Angabe von MQOO\_ALTERNATE\_USER\_AUTHORITY für die alternative Benutzer-ID), mit welcher der entsprechende Aufruf MQOPEN validiert wurde, werden Berechtigungsprüfungen vorgenommen.
- Sie können wie normale Warteschlangen gelöscht werden.

### Verwendung dynamischer Warteschlangen

Dynamische Warteschlangen können in folgenden Fällen eingesetzt werden:

- Für Anwendungen, nach deren Beendigung keine Warteschlangen beibehalten werden müssen.
- Für Anwendungen, bei denen Antworten auf Nachrichten von einer anderen Anwendung verarbeitet werden müssen. Solche Anwendungen können durch Öffnen einer Modellwarteschlange dynamisch eine Empfangswarteschlange für Antworten erstellen. Eine Clientanwendung kann beispielsweise folgende Aktionen ausführen:
  1. Eine dynamische Warteschlange erstellen.
  2. Im Feld *ReplyToQ* der Nachrichtendeskriptorstruktur der Anforderungsnachricht deren Namen angeben.
  3. Die Anforderung in eine von einem Server verarbeitete Warteschlange stellen.

Der Server kann die Antwortnachricht dann in die Empfangswarteschlange für Antworten einreihen. Der Client wiederum kann nun die Antwort verarbeiten und die Empfangswarteschlange für Antworten mit der Löschoption schließen.

### Bei der Verwendung dynamischer Warteschlangen zu berücksichtigende Aspekte

Bei der Verwendung dynamischer Warteschlangen sind folgende Punkte zu beachten:

- In einem Client-Server-Modell müssen die einzelnen Clients jeweils eine eigene dynamische Empfangswarteschlange für Antworten erstellen und verwenden. Falls eine dynamische Empfangswarteschlange für Antworten von mehreren Clients gemeinsam genutzt wird, kann es sein, dass die Empfangswarteschlange für Antworten erst mit Verzögerung gelöscht wird, da für die Warteschlange nicht festgeschriebene Aktivitäten ausstehen oder die Warteschlange gerade von einem anderen Client verwendet wird. Außerdem wird die Warteschlange möglicherweise als logisch gelöscht markiert. Nachfolgende API-Anforderungen (außer MQCLOSE) können dann nicht mehr auf die Warteschlange zugreifen.
- Müssen in Ihrer Anwendungsumgebung dynamische Warteschlangen von mehreren Anwendungen gemeinsam genutzt werden, ist darauf zu achten, dass die Warteschlange erst geschlossen wird (mit der Löschoption), wenn die gesamte Aktivität für die Warteschlange festgeschrieben wurde. Diese Aktion sollte der letzte Benutzer ausführen. Auf diese Weise wird gewährleistet, dass die



Warteschlange ohne Verzögerungen gelöscht wird. Außerdem wird der Zeitraum, in dem die Warteschlange nicht zugänglich ist, da sie als logisch gelöscht markiert ist, auf ein Minimum reduziert.

## Modellwarteschlangen

Eine *Modellwarteschlange* ist eine Schablone einer Warteschlangendefinition, die beim Erstellen einer dynamischen Warteschlange verwendet wird.

Von einem WebSphere MQ-Programm aus kann dynamisch eine lokale Warteschlange erstellt werden, wobei die als Schablone für die Warteschlangenattribute zu verwendende Modellwarteschlange angegeben wird. Zu diesem Zeitpunkt können einige Attribute der neuen Warteschlange geändert werden. Eine Änderung des Attributs *DefinitionType* ist jedoch nicht möglich. Falls Sie also beispielsweise eine permanente Warteschlange benötigen, wählen Sie eine Modellwarteschlange aus, für die als Definitionstyp 'permanent' angegeben ist. Manche Dialoganwendungen können für Antworten auf ihre Anfragen dynamische Warteschlangen verwenden, da diese Warteschlangen nach Verarbeitung der Antworten vermutlich nicht beibehalten werden müssen.

Den Namen einer Modellwarteschlange geben Sie im *Objektdeskriptor* (MQOD) des Aufrufs MQOPEN an. Unter Verwendung der Attribute der Modellwarteschlange wird vom Warteschlangenmanager dynamisch eine lokale Warteschlange erstellt.

Sie können einen (vollständigen) Namen für die dynamische Warteschlange angeben oder aber nur den Namensstamm (z. B. ABC), dem dann vom Warteschlangenmanager ein eindeutiger Teil hinzugefügt wird, Sie können aber auch die Zuordnung eines eindeutigen Namens komplett dem Warteschlangenmanager überlassen. Falls der Warteschlangenmanager den Namen zuordnet, stellt er ihn in die MQOD-Struktur.

Es ist nicht möglich, direkt für eine Modellwarteschlange einen Aufruf MQPUT1 auszugeben, für die dynamische Warteschlange, die durch Öffnen einer Modellwarteschlange erstellt wurde, kann der Aufruf MQPUT1 dagegen ausgegeben werden.

MQSET und MQINQ können nicht an einer Modellwarteschlange ausgeführt werden. Wenn Sie eine Modellwarteschlange mit MQOO\_INQUIRE oder MQOO\_SET öffnen, werden an der dynamisch erstellten Warteschlange weitere MQINQ- und MQSET-Aufrufe ausgeführt.

Bei den Attributen einer Modellwarteschlange handelt es sich um eine Untergruppe der Attribute einer lokalen Warteschlange. Eine detailliertere Beschreibung finden Sie im Abschnitt [Attribute für Warteschlangen](#).

### *Warteschlangen definieren*

Warteschlangen können in IBM WebSphere MQ über den WebSphere MQ-Scriptbefehl DEFINE oder den PCF-Befehl 'Create Queue' definiert werden.

Mit den Befehlen werden Typ und Attribute der Warteschlange angegeben. Über die Attribute einer lokalen Warteschlange wird beispielsweise festgelegt, was geschieht, wenn Anwendungen in MQI-Aufrufen auf diese Warteschlange verweisen. Hier einige Beispiele für Attribute:

- Ob Anwendungen Nachrichten aus der Warteschlange abrufen können (GET-aktiviert).
- Ob Anwendungen Nachrichten in die Warteschlange einreihen können (PUT-aktiviert).
- Ob nur eine Anwendung oder aber mehrere Anwendungen auf die Warteschlange zugreifen können.
- Die maximale Anzahl an Nachrichten, die gleichzeitig in der Warteschlange gespeichert werden können (max. Warteschlangenlänge).
- Die Länge, die Nachrichten maximal haben dürfen, damit sie in die Warteschlange eingereiht werden können.

Weitere Informationen zum Definieren von Warteschlangenobjekten finden Sie unter [Scriptbefehle \(MQSC\)](#).

### *Von IBM WebSphere MQ verwendete Warteschlangen*

In IBM WebSphere MQ werden einige lokale Warteschlangen für bestimmte Aspekte des Betriebsablaufs verwendet.

Diese Warteschlangen müssen erst definiert werden, damit sie von IBM WebSphere MQ verwendet werden können.

### Initialisierungswarteschlangen

Initialisierungswarteschlangen werden für die Auslösefunktion (Triggering) eingesetzt. Wenn ein Auslöserereignis eintritt, reiht der Warteschlangenmanager eine Auslösenachricht in eine Initialisierungswarteschlange ein. Ein Auslöserereignis ist eine logische Kombination an Bedingungen, die von einem Warteschlangenmanager erkannt wird. So kann beispielsweise ein Auslöserereignis generiert werden, wenn die Anzahl der Nachrichten in einer Warteschlange die vorgegebene Warteschlangengröße erreicht hat. Bei Eintreten dieses Ereignisses reiht der Warteschlangenmanager eine Auslösenachricht in die angegebene Initialisierungswarteschlange ein. Diese Auslösenachricht wird von einem *Auslösemonitor* abgerufen, bei dem es sich um eine besondere Anwendung für die Überwachung einer Initialisierungswarteschlange handelt. Der Auslösemonitor startet daraufhin das in der Auslösenachricht angegebene Anwendungsprogramm.

Soll die Auslösefunktion in einem Warteschlangenmanager verwendet werden, muss mindestens eine Initialisierungswarteschlange für ihn definiert sein. Weitere Informationen finden Sie in den Abschnitten [Objekte für Auslösefunktion verwalten](#), [runmqtrm](#) und [WebSphere MQ-Anwendungen mithilfe von Auslösern starten](#).

### Übertragungswarteschlangen

In Übertragungswarteschlangen werden kurzfristig Nachrichten gespeichert, die an einen fernen Warteschlangenmanager gesendet werden sollen. Sie müssen zumindest eine Übertragungswarteschlange für jeden fernen Warteschlangenmanager definieren, an den der lokale Warteschlangenmanager direkt Nachrichten senden soll. Diese Warteschlangen werden auch bei der Fernverwaltung verwendet (siehe [Fernverwaltung von einem lokalen Warteschlangenmanager aus](#)). Informationen zur Verwendung von Übertragungswarteschlangen in der verteilten Steuerung von Warteschlangen finden Sie im Artikel [WebSphere MQ Verfahren für verteiltes Messaging](#).

Für jeden Warteschlangenmanager kann eine Standard-Übertragungs-WS definiert werden. Wenn ein Warteschlangenmanager, der zu keinem Cluster gehört, eine Nachricht in eine ferne Warteschlange einreicht, wird standardmäßig die Standard-Übertragungs-WS verwendet. Wenn eine Übertragungswarteschlange mit demselben Namen wie dem des Ziel-Warteschlangenmanagers vorhanden ist, wird die Nachricht in diese Übertragungswarteschlange gestellt. Ist eine Definition eines Warteschlangenmanager-Aliasnamens vorhanden, deren Parameter **RQMNAME** mit dem Zielwarteschlangenmanager übereinstimmt, und ist der Parameter **XMITQ** angegeben, wird die Nachricht in die in **XMITQ** benannte Übertragungswarteschlange eingereiht. Wenn kein **XMITQ**-Parameter vorhanden ist, wird die Nachricht in die in der Nachricht angegebene lokale Warteschlange gestellt.

### Clusterübertragungswarteschlangen

Zu jedem Warteschlangenmanager in einem Cluster gibt es eine Clusterübertragungswarteschlange **SYSTEM.CLUSTER.TRANSMIT.QUEUE** und eine Modell-Clusterübertragungswarteschlange **SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE**. Definitionen dieser Warteschlangen werden standardmäßig bei der Definition eines Warteschlangenmanagers erstellt. Wenn das Warteschlangenmanager-Attribut **DEFCLXQ** auf **CHANNEL** gesetzt ist, wird automatisch eine permanente dynamische Clusterübertragungs-WS für jeden erstellten Clustersenderkanal erstellt. Die Warteschlangen heißen **SYSTEM.CLUSTER.TRANSMIT.ChannelName**. Cluster-Übertragungs-WS können auch manuell definiert werden.

Ein Warteschlangenmanager, der zu einem Cluster gehört, sendet Nachrichten in einer dieser Warteschlangen an andere Warteschlangenmanager im selben Cluster.

Bei der Namensauflösung hat eine Cluster-Übertragungs-WS Vorrang vor der standardmäßigen Übertragungswarteschlange und eine spezifische Cluster-Übertragungs-WS hat Vorrang vor **SYSTEM.CLUSTER.TRANSMIT.QUEUE**.

### Warteschlangen für nicht zustellbare Nachrichten

In Warteschlangen für nicht zustellbare Nachrichten werden Nachrichten gespeichert, die nicht an ihr vorgegebenes Ziel gesendet werden können. Eine Nachricht kann nicht weitergeleitet werden, wenn beispielsweise die Zielwarteschlange voll ist. Die bereitgestellte Warteschlange für nicht zustellbare Nachrichten hat den Namen **SYSTEM.DEAD.LETTER.QUEUE**.

Bei einer verteilten Steuerung von Warteschlangen muss für jeden beteiligten Warteschlangenmanager eine Warteschlange für nicht zustellbare Nachrichten definiert werden.

### **Befehlswarteschlangen**

Die Befehlswarteschlange `SYSTEM.ADMIN.COMMAND.QUEUE` ist eine lokale Warteschlange, an die entsprechend berechnete Anwendungen WebSphere MQ-Scriptbefehle zur Verarbeitung senden können. Diese Befehle werden anschließend von einer IBM WebSphere MQ-Komponente, dem sogenannten Befehlsserver, abgerufen. Der Befehlsserver überprüft die Befehle, sendet die gültigen Befehle weiter zur Verarbeitung durch den Warteschlangenmanager und gibt Antworten an die entsprechende Empfangswarteschlange für Antworten zurück.

Die Befehlswarteschlange wird automatisch bei der Erstellung eines Warteschlangenmanagers erstellt.

### **Empfangswarteschlangen für Antworten**

Wenn eine Anwendung eine Anforderungsnachricht sendet, kann die Anwendung, die diese Nachricht enthält, eine Antwortnachricht an die Anwendung zurückgeben, von der die Nachricht stammt. Diese Nachricht wird in eine sogenannte Empfangswarteschlange für Antworten eingereiht, bei der es sich in der Regel um eine lokale Warteschlange der Anwendung handelt, die die Nachricht sendet. Der Name der Empfangswarteschlange wird von der sendenden Anwendung als Teil des Nachrichtendeskriptors angegeben.

### **Ereigniswarteschlangen**

Mithilfe von Instrumentierungsereignissen können Warteschlangenmanager unabhängig von MQI-Anwendungen überwacht werden.

Bei Eintreten eines Instrumentierungsereignisses reiht der Warteschlangenmanager eine Ereignisnachricht in eine Ereigniswarteschlange ein. Diese Nachricht kann von einer Überwachungsanwendung gelesen werden, die entweder den Administrator informieren oder aber eine Korrekturmaßnahme einleiten kann, wenn das Ereignis auf ein Problem hindeutet.

**Anmerkung:** Auslöserereignisse und Instrumentierungsereignisse sollten nicht verwechselt werden. Auslöserereignisse treten unter anderen Bedingungen auf und generieren auch keine Ereignisnachrichten.

Weitere Informationen zu Instrumentierungsereignissen finden Sie im Abschnitt [Instrumentierungsereignisse](#).

## **IBM WebSphere MQ-Warteschlangenmanager**

Dieser Abschnitt enthält eine Einführung in *Warteschlangenmanager* und die Services für die Warteschlangensteuerung, die sie für die Anwendungen bereitstellen.

Damit ein Programm die Services eines Warteschlangenmanagers nutzen kann, muss es eine Verbindung zu diesem Warteschlangenmanager herstellen. Diese Verbindung kann vom Programm direkt (über den `MQCONN`- oder `MQCONNX`-Aufruf) oder indirekt (abhängig von der Plattform und der Umgebung, auf der bzw. in der das Programm aktiv ist) hergestellt werden.

Warteschlangenmanager stellen Services zur Warteschlangensteuerung für Anwendungen bereit und verwalten die Warteschlangen, die zu ihnen gehören. Warteschlangenmanager sorgen dafür, dass:

- Objektattribute entsprechend den empfangenen Befehlen geändert werden;
- besondere Ereignisse wie Auslöser- oder Instrumentierungsereignisse generiert werden, wenn die entsprechenden Bedingungen erfüllt sind;
- Nachrichten entsprechend dem `MQPUT`-Aufruf einer Anwendung in die richtige Warteschlange eingereiht werden. Ist dies nicht möglich, wird die Anwendung unter Angabe des entsprechenden Ursachen-codes darüber informiert.

Jede Warteschlange gehört nur zu einem Warteschlangenmanager und wird als *lokale Warteschlange* dieses Warteschlangenmanagers bezeichnet. Der Warteschlangenmanager, mit dem die Anwendung verbunden ist, wird als *lokaler Warteschlangenmanager* dieser Anwendung bezeichnet. Für die Anwendung sind die Warteschlangen ihres lokalen Warteschlangenmanagers lokale Warteschlangen.

Eine *ferne Warteschlange* ist eine Warteschlange, die zu einem anderen Warteschlangenmanager gehört. Als *fern* werden alle Warteschlangenmanager bezeichnet, bei denen es sich nicht um den lokalen Warteschlangenmanager handelt. Ein ferner Warteschlangenmanager kann sich auf einer fernen Maschine im Netz oder auf derselben Maschine wie der lokale Warteschlangenmanager befinden. WebSphere MQ unterstützt das Vorhandensein mehrerer Warteschlangenmanager auf ein und derselben Maschine.

In einigen MQI-Aufrufen können Warteschlangenmanagerobjekte verwendet werden. So können Sie beispielsweise mit dem MQI-Aufruf MQINQ die Attribute des Warteschlangenmanager-Objekts abfragen.

## Attribute von Warteschlangenmanagern

Jedem Warteschlangenmanager sind eine Reihe von Attributen (oder Eigenschaften) zugeordnet, die diesen Warteschlangenmanager definieren. Einige dieser Attribute werden bei der Erstellung eines Warteschlangenmanagers festgelegt; andere Attribute hingegen können über WebSphere MQ-Befehle geändert werden. Mit Ausnahme der für die SSL-Verschlüsselung (Secure Sockets Layer) verwendeten Attribute können Sie die Werte aller Attribute mit dem MQINQ-Aufruf abfragen.

Zu den *festgelegten* Attributen gehören die folgenden:

- Der Name des Warteschlangenmanagers.
- Die Plattform, auf der der Warteschlangenmanager aktiv ist (beispielsweise Windows).
- Die Systemsteuerbefehle, die der Warteschlangenmanager unterstützt.
- Die Priorität, die den vom Warteschlangenmanager verarbeiteten Nachrichten maximal zugeordnet werden kann.
- Der Name der Warteschlange, an die Programme WebSphere MQ-Befehle senden können.
- Die Länge, die Nachrichten maximal haben dürfen, um vom Warteschlangenmanager noch verarbeitet werden zu können
- Ob der Warteschlangenmanager beim Einreihen und Abrufen von Nachrichten Synchronisationspunkte unterstützt.

Zu den Attribute, die *geändert* werden können, gehören die folgenden:

- Eine Beschreibung des Warteschlangenmanagers im Textformat.
- Die ID des Zeichensatzes, den der Warteschlangenmanager bei der Verarbeitung von MQI-Aufrufen für Zeichenfolgen verwendet.
- Das Zeitintervall, mit dem der Warteschlangenmanager die Anzahl von Auslösenachrichten einschränkt.
- Der Name der Warteschlange für nicht zustellbare Nachrichten, die dem Warteschlangenmanager zugeordnet ist.
- Der Name der Übertragungswarteschlange des Warteschlangenmanagers.
- Die maximale Anzahl geöffneter Kennungen für jede Verbindung.
- Die Aktivierung und Inaktivierung verschiedener Kategorien von Ereignisberichten.
- Die maximale Anzahl nicht festgeschriebener Nachrichten innerhalb einer Arbeitseinheit.

## Warteschlangenmanager und Workload-Management

Sie können einen Warteschlangenmanager-Cluster einrichten, für den mehrere Definitionen ein und derselben Warteschlange vorhanden sind (bei den Warteschlangenmanagern im Cluster kann es sich beispielsweise um Klone handeln). Nachrichten für eine bestimmte Warteschlange können von jedem Warteschlangenmanager verarbeitet werden, der über eine Instanz dieser Warteschlange verfügt. Mithilfe eines Workload-Management-Algorithmus wird der Warteschlangenmanager ermittelt, der die Nachricht verarbeiten soll. Auf diese Weise wird die Arbeitslast auf die einzelnen Warteschlangenmanager verteilt. Weitere Informationen hierzu finden Sie im Abschnitt [Der Cluster-Workload-Management-Algorithmus](#).

## **Prozessdefinitionen**

Über Prozessdefinitionsobjekte können Anwendungen ohne Bedieneringriff gestartet werden; dazu werden die Attribute der Anwendung für den Warteschlangenmanager definiert.

Das Prozessdefinitionsobjekt definiert eine Anwendung, die als Reaktion auf ein Auslöserereignis auf einem IBM WebSphere MQ -Warteschlangenmanager gestartet wird. Zu den Prozessdefinitionsattributen gehören die Anwendungs-ID, der Anwendungstyp sowie die anwendungsspezifischen Daten. Weitere Informationen finden Sie im Eintrag "Initialisierungswarteschlangen" im Abschnitt [„Von IBM WebSphere MQ verwendete Warteschlangen“](#) auf Seite 121.

Damit eine Anwendung ohne Bedieneringriff gestartet werden kann (siehe Abschnitt [WebSphere MQ-Anwendungen mithilfe von Auslösern starten](#)), müssen ihre Attribute dem Warteschlangenmanager bekannt sein. Diese Attribute werden in einem *Prozessdefinitionsobjekt* definiert.

Das Attribut *ProcessName* wird festgelegt, wenn das Objekt erstellt wird; Sie können andere Attribute ändern, indem Sie die IBM WebSphere MQ -Befehle verwenden.

Informationen zu den Werten *aller* Attribute sind im Abschnitt [MQINQ – Objektattribute abfragen](#) zu finden.

Eine vollständige Beschreibung der Attribute von Prozessdefinitionen finden Sie im Abschnitt [Attribute für Prozessdefinitionen](#).

## **Namenslisten**

Eine *Namensliste* ist ein WebSphere MQ-Objekt, das eine Liste mit Clusternamen, Warteschlangennamen oder Namen von Authentifizierungsdatenobjekten enthält. In einem Cluster kann dieses Objekt eine Liste der Cluster enthalten, für die der Warteschlangenmanager die Repositories enthält.

Eine Namensliste ist ein WebSphere MQ-Objekt, das eine Liste anderer WebSphere MQ-Objekte enthält. Namenslisten werden in der Regel von Anwendungen wie beispielsweise Auslösemonitoren verwendet (in diesem Fall ist in diesem Objekt eine Gruppe von Warteschlangen angegeben). Namenslisten haben den Vorteil, dass sie unabhängig von Anwendungen verwaltet werden können; sie können aktualisiert werden, ohne dass eine der Anwendungen gestoppt werden muss, die dieses Objekt verwenden. Der Ausfall einer Anwendung hat keinerlei Auswirkungen auf die Namensliste - sie kann von den anderen Anwendungen weiterhin verwendet werden.

Namenslisten werden in Warteschlangenmanager-Clustern für Listen mit Clustern verwendet, auf die von mehreren WebSphere MQ-Objekten verwiesen wird.

Namenslisten können mit MQSC-Befehlen definiert und geändert werden.

Programme können über die MQI feststellen, welche Warteschlangen in diesen Namenslisten aufgeführt sind. Die Verwaltung der Namenslisten ist Aufgabe des Anwendungsentwicklers und des Systemadministrators.

Eine umfassende Beschreibung der Attribute von Namenslisten finden Sie im Abschnitt [Attribute für Namenslisten](#).

## **Authentifizierungsdatenobjekte**

Dieser Abschnitt enthält eine Einführung in Authentifizierungsdatenobjekte von Warteschlangenmanagern sowie Links zu weiteren Informationen.

Das Authentifizierungsdatenobjekt eines Warteschlangenmanagers ist Teil der SSL- und TLS-Unterstützung (Secure Sockets Layer; Transport Layer Security) von WebSphere MQ. Es stellt die Definitionen bereit, die für eine Überprüfung auf widerrufenen Zertifikate erforderlich sind. Zertifikate, die nicht mehr als vertrauenswürdig eingestuft werden, werden von Zertifizierungsstellen widerrufen.

In diesem Abschnitt wird die Verwendung der Befehle `setmqaut`, `dspmqaut`, `dmpmqaut`, `rcimqobj`, `rcdmqimg` und `dspmqfls` in Zusammenhang mit Authentifizierungsdatenobjekten beschrieben. Eine Übersicht über SSL und TLS und die Verwendung der Authentifizierungsdatenobjekte finden Sie unter [WebSphere MQ -Unterstützung für SSL und TLS](#).

Weitere Informationen zu SSL und TLS finden Sie unter [Konzepte von Secure Sockets Layer \(SSL\) und Transport Layer Security \(TLS\)](#).

Ein Authentifizierungsdatenobjekt stellt die Definitionen bereit, die für die Überprüfung auf widerrufenen Zertifikate erforderlich sind.

Eine umfassende Beschreibung der Attribute von Authentifizierungsdatenobjekten finden Sie im Abschnitt [Authentifizierungsdatenobjekte](#).

### **Kommunikationsinformationsobjekte**

IBM WebSphere MQ Multicast bietet eine geringe Latenzzeit, eine hohe Ausgabefächerung und eine zuverlässige Multicasting-Nachrichtenübertragung. Für die Multicasting-Übertragung ist ein Kommunikationsinformationsobjekt (COMMINFO-Objekt) erforderlich.

Bei COMMINFO-Objekten handelt es sich um IBM WebSphere MQ-Objekte, die Attribute in Zusammenhang mit Multicast-Übertragungen enthalten. Weitere Informationen über diese Attribute finden Sie unter [DEFINE COMMINFO](#). Weitere Informationen zur Erstellung von COMMINFO-Objekten finden Sie im Abschnitt [Erste Schritte mit Multicasting](#).

### **Zugehörige Konzepte**

[„IBM WebSphere MQ Multicasting“ auf Seite 137](#)

IBM WebSphere MQ Multicast bietet eine geringe Latenzzeit, eine hohe Ausgabefächerung und eine zuverlässige Multicasting-Nachrichtenübertragung.

### **Kanäle**

Ein *Kanal* ist eine Kommunikationsverbindung, die von verteilten Warteschlangenmanagern verwendet wird.

*Kanäle* sind Objekte, die einen Kommunikationspfad von einem Warteschlangenmanager zu einem anderen bereitstellen. Kanäle werden bei der verteilten Steuerung von Warteschlangen für die Übertragung von Nachrichten zwischen Warteschlangenmanagern verwendet und schirmen Anwendungen von den verwendeten Kommunikationsprotokollen ab. Die Warteschlangenmanager können sich dabei auf derselben oder auf verschiedenen Plattformen befinden.

Damit Warteschlangenmanager miteinander kommunizieren können, müssen Sie einen Kanal auf der Seite des Warteschlangenmanagers definieren, der Nachrichten senden soll, sowie einen Kanal auf der Seite des Warteschlangenmanagers, der diese Nachrichten empfangen soll.

Es gibt zwei Kanaltypen in WebSphere MQ:

- Unidirektionale *Nachrichtenkanäle*, über die Nachrichten von einem Warteschlangenmanager an einen anderen übertragen werden. Weitere Informationen hierzu finden Sie im Abschnitt [Kanalexitaufrufe und Datenstrukturen](#).
- Bidirektionale *MQI-Kanäle*, über die MQI-Aufrufe von einem WebSphere MQ MQI-Client an einen Warteschlangenmanager und Antworten von einem Warteschlangenmanager an einen WebSphere MQ-Client übertragen werden; weitere Informationen finden Sie im Abschnitt [„Was ist ein Kanal?“ auf Seite 127](#).

### **Zugehörige Konzepte**

[„Konzepte der übergreifenden Kommunikation“ auf Seite 53](#)

In WebSphere MQ bezeichnet übergreifende Kommunikation das Senden von Nachrichten von einem Warteschlangenmanager an einen anderen. Der empfangende Warteschlangenmanager kann sich auf demselben System oder auf einem anderen System ganz in der Nähe oder am anderen Ende der Welt befinden. Er kann auf derselben Plattform wie der lokale Warteschlangenmanager oder jeder anderen von WebSphere MQ unterstützten Plattform aktiv sein. Dies wird als *verteilte* Umgebung bezeichnet. WebSphere MQ wickelt die Kommunikation in einer verteilten Umgebung wie dieser mithilfe des verteilten Warteschlangenmanagements ab.

### **Zugehörige Tasks**

[Verwaltung ferner WebSphere MQ-Objekte](#)

### **Zugehörige Verweise**

[Kanalexitaufrufe und Datenstrukturen](#)

[„Kommunikation“ auf Seite 129](#)

WebSphere MQ MQI-Clients kommunizieren über MQI-Kanäle mit dem Server.

Was ist ein Kanal?

Ein Kanal ist eine logische Kommunikationsverbindung zwischen einem WebSphere MQ MQI-Client und einem WebSphere MQ-Server bzw. zwischen zwei WebSphere MQ-Servern.

Ein Kanal verfügt über zwei Definitionen, nämlich je eine an jedem Ende der Verbindung. An beiden Enden muss derselbe *Kanalname* verwendet werden und die *Kanaltypen* müssen kompatibel sein.

In WebSphere MQ gibt es zwei Kanalgruppen mit jeweils verschiedenen Kanaltypen:

### Zugehörige Konzepte

„Nachrichtenkanäle“ auf Seite 127

Ein Nachrichtenkanal ist eine unidirektionale Verbindung, mit der über *Nachrichtenkanalagenten* (Message Channel Agents; MAC) zwei Warteschlangenmanager miteinander verbunden werden.

„MQI-Kanäle“ auf Seite 127

Ein MQI-Kanal verbindet einen WebSphere MQ mit einem Warteschlangenmanager auf einer Servermaschine und wird eingerichtet, wenn Sie einen MQCONN -oder MQCONNX -Aufruf von einer WebSphere MQ MQI-Clientanwendung absetzen.

„Kanäle stoppen“ auf Seite 128

Bei Ausgabe eines STOP CHANNEL-Befehls für einen Serververbindungskanal in WebSphere MQ können Sie angeben, wie der Clientverbindungskanal gestoppt werden soll.

### Nachrichtenkanäle

Ein Nachrichtenkanal ist eine unidirektionale Verbindung, mit der über *Nachrichtenkanalagenten* (Message Channel Agents; MAC) zwei Warteschlangenmanager miteinander verbunden werden.

Über Nachrichtenkanäle werden Nachrichten zwischen den Warteschlangenmanagern übertragen. In der Client/Server-Umgebung sind keine Nachrichtenkanäle erforderlich.

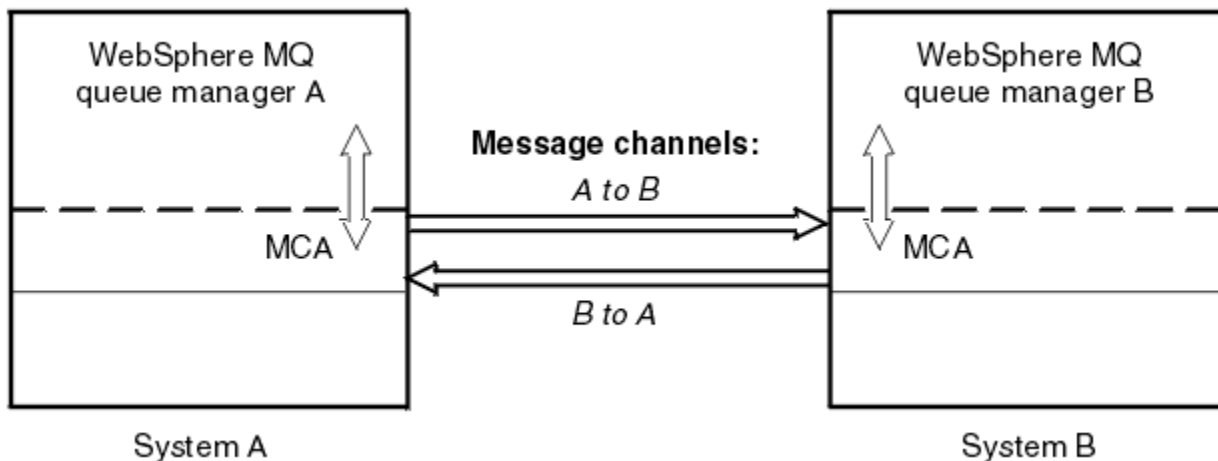


Abbildung 28. Nachrichtenkanäle zwischen zwei Warteschlangenmanagern

### MQI-Kanäle

Ein MQI-Kanal verbindet einen WebSphere MQ mit einem Warteschlangenmanager auf einer Servermaschine und wird eingerichtet, wenn Sie einen MQCONN -oder MQCONNX -Aufruf von einer WebSphere MQ MQI-Clientanwendung absetzen.

MQI-Kanäle sind bidirektionale Verbindungen, die nur für die Übertragung von MQI-Aufrufen und -Antworten verwendet werden (z. B. MQPUT-Aufrufe, die Nachrichtendaten enthalten, und MQGET-Aufrufe, mit denen Nachrichtendaten zurückgegeben werden). Es gibt verschiedene Möglichkeiten, Kanaldefinitionen zu erstellen und zu verwenden (siehe [MQI-Kanäle definieren](#)).

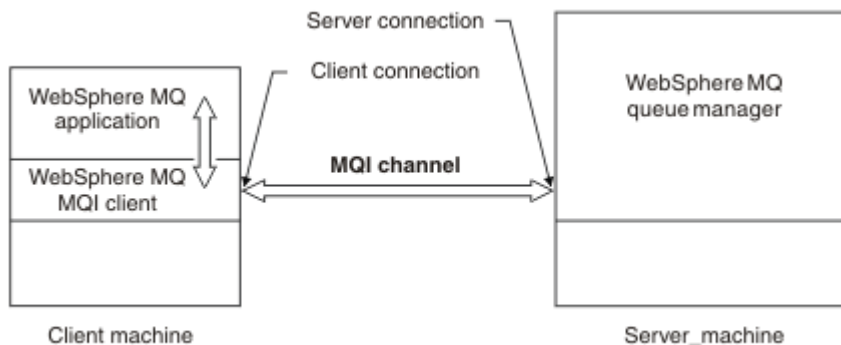


Abbildung 29. Clientverbindung und Serververbindung über einen MQI-Kanal

Mit einem MQI-Kanal kann ein Client mit einem einzelnen Warteschlangenmanager oder mit einem Warteschlangenmanager, der zu einer Gruppe mit gemeinsamer Warteschlange gehört, verbunden werden (siehe [Client mit einer Gruppe mit gemeinsamer Warteschlange verbinden](#)).

Es gibt zwei Kanaltypen für MQI-Kanaldefinitionen. Sie beschreiben den bidirektionalen MQI-Kanal.

### Clientverbindungskanal

Dieser Kanaltyp ist für den WebSphere MQ MQI-Client.

### Serververbindungskanal

Dieser Kanaltyp ist für den Server, auf dem sich der Warteschlangenmanager befindet, mit dem die in einer WebSphere MQ MQI-Clientumgebung aktive WebSphere MQ-Anwendung kommunizieren soll.

### Kanäle stoppen

Bei Ausgabe eines STOP CHANNEL-Befehls für einen Serververbindungskanal in WebSphere MQ können Sie angeben, wie der Clientverbindungskanal gestoppt werden soll.

Dies bedeutet, dass ein Clientkanal, der einen MQGET-Warteaufruf ausgibt, gesteuert werden kann, und Sie können entscheiden, wie und wann der Kanal gestoppt werden soll.

Der Befehl STOP CHANNEL kann mit drei Modi ausgeführt werden, die angeben, wie der Kanal gestoppt werden soll:

### QUIESCE

Stoppt den Kanal, nachdem alle aktuellen Nachrichten verarbeitet wurden.

Wenn der gemeinsam genutzte Datenaustausch aktiviert ist, erhält der WebSphere MQ MQI-Client abhängig von der Netzgeschwindigkeit rechtzeitig Kenntnis von der Stoppanforderung. Die Clientanwendung wird durch die Ausgabe eines nachfolgenden Aufrufs an WebSphere MQ von der Stoppanforderung unterrichtet.

### 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).

#### Kommunikation

WebSphere MQ MQI-Clients kommunizieren über MQI-Kanäle mit dem Server.

Auf der WebSphere MQ MQI-Client- und -Serverseite der Verbindung muss jeweils eine Kanaldefinition erstellt werden. Informationen zum Erstellen von Kanaldefinitionen finden Sie im Abschnitt [MQI-Kanäle definieren](#).

In der folgenden Tabelle sind die möglichen Übertragungsprotokolle aufgeführt:

Tabelle 8. Übertragungsprotokolle für MQI-Kanäle				
Clientplattform	LU 6.2	TCP/IP	NetBIOS	SPX
UNIX and Linux-Systeme	Ja <sup>1</sup>	Ja		
Windows	Ja	Ja	Ja	Ja
<b>Anmerkung:</b>				
1. LU6.2 wird unter Linux (Plattform POWER ), Linux (Plattformx86-64 ), Linux (Plattform zSeries s390x ) oder Solaris (Plattformx86-64 ) nicht unterstützt.				

In Übertragungsprotokolle - Kombination von WebSphere MQ-Client- und Serverplattformen sind die möglichen Kombinationen von WebSphere MQ-Client- und Serverplattformen mit diesen Übertragungsprotokollen aufgeführt.

Eine WebSphere MQ-Anwendung auf einem WebSphere MQ MQI-Client kann alle MQI-Aufrufe so verwenden, als ob ein lokaler Warteschlangenmanager verwendet wird. Der **MQCONN**- oder **MQCONNX**-Aufruf ordnet die WebSphere MQ-Anwendung dem ausgewählten Warteschlangenmanager zu und erstellt auf diese Weise eine *Verbindungskennung*. Alle weiteren Aufrufe, die diese Verbindungskennung verwenden, werden daraufhin von dem verbundenen Warteschlangenmanager verarbeitet. Im Unterschied zur Kommunikation der Warteschlangenmanager untereinander, die verbindungs- und zeitunabhängig erfolgt, ist für die WebSphere MQ MQI-Clientkommunikation eine aktive Verbindung zwischen Client und Server erforderlich.

Das Übertragungsprotokoll wird über die Verwendung der Kanaldefinition festgelegt und hat keine Auswirkung auf die Anwendung. So kann beispielsweise eine Windows-Anwendung zu einem Warteschlangenmanager eine Verbindung über TCP/IP, zu einem anderen über NetBIOS herstellen.

## Leistungsaspekte

Das von Ihnen verwendete Übertragungsprotokoll kann sich auf die Leistung des WebSphere MQ-Client/Server-Systems auswirken. Als Anwahlunterstützung über langsame Telefonleitungen kann die Verwendung der WebSphere MQ-Kanalkomprimierung sinnvoll sein.

### Clientverbindungskanäle

Dieser Abschnitt enthält eine Einführung in Clientverbindungskanalobjekte sowie einen Link zu weiteren Informationen.

*Clientverbindungskanäle* sind Objekte, die einen Kommunikationspfad von einem WebSphere MQ MQI-Client zu einem Warteschlangenmanager bereitstellen. Clientkommunikationskanäle werden bei der verteilten Steuerung von Warteschlangen für die Übertragung von Nachrichten zwischen einem Warteschlangenmanager und einem Client verwendet. Sie schirmen Anwendungen von den verwendeten Kommunikationsprotokollen ab. Der Client kann sich auf derselben Plattform wie der Warteschlangenmanager oder auf einer anderen Plattform befinden.

Weitere Informationen zu Clientverbindungskanälen und ihrer Verwendung finden Sie im Abschnitt [„Intercommunication“](#) auf Seite 38.

## Empfangsprogramme

*Empfangsprogramme* sind Prozesse, die Netzanforderungen von anderen Warteschlangenmanagern oder Clientanwendungen annehmen und die zugeordneten Kanäle starten.

*Empfangsprogramme* sind Prozesse, die Netzanforderungen von anderen Warteschlangenmanagern oder Clientanwendungen annehmen und die zugeordneten Kanäle starten. Empfangsprogrammprozesse werden mit dem Steuerbefehl `runmqtsr` gestartet.

*Empfangsprogrammobjekte* sind WebSphere MQ-Objekte, mit denen Empfangsprogrammprozesse auf Warteschlangenmanagerebene gestartet und gestoppt werden können. Über die Definition der Attribute eines Empfangsprogramms können Sie:

- den Empfangsprogrammprozess definieren;
- angeben, ob der Empfangsprogrammprozess beim Starten bzw. Stoppen des Warteschlangenmanagers automatisch gestartet bzw. gestoppt werden soll.

**In WebSphere MQ for z/OS werden Empfangsprogrammobjekte nicht unterstützt.**

## Services

Über *Serviceobjekte* können die Programme definiert werden, die beim Starten oder Stoppen eines Warteschlangenmanagers ausgeführt werden sollen.

Es kann sich um folgende Programmtypen handeln:

### Server

Für Serviceobjekte des Typs *Server* ist der Parameter `SERVTYPE` auf `SERVER` gesetzt. Bei einem Serviceobjekt des Typs `SERVER` handelt es sich um die Definition eines Programms, das beim Start des angegebenen Warteschlangenmanagers ausgeführt wird. Es kann immer nur jeweils eine Instanz eines Serverprozesses ausgeführt werden. Der Status eines Serverprozesses kann während der Ausführung mit dem WebSphere MQ-Scriptbefehl `DISPLAY SVSTATUS` überwacht werden. Serverserviceobjekte sind in der Regel Definitionen von Programmen wie beispielsweise Steuerroutinen für nicht zustellbare Nachrichten oder Auslösemonitore; die Programme, die ausgeführt werden können, sind allerdings nicht auf die von WebSphere MQ bereitgestellten Programme beschränkt. Darüber hinaus kann ein Serverserviceobjekt auch einen Befehl enthalten, mit dem das Programm nach der Beendigung des angegebenen Warteschlangenmanagers beendet wird.

### Befehle

Ein *Befehl* ist ein Serviceobjekt, für das der Parameter `SERVTYPE` auf `COMMAND` gesetzt ist. Ein Befehlsserviceobjekt ist die Definition eines Programms, das beim Starten oder Stoppen des angegebenen Warteschlangenmanagers ausgeführt wird. Mehrere Instanzen eines Befehlsprozesses können gleichzeitig ausgeführt werden. Befehlsserviceobjekte unterscheiden sich von Serverserviceobjekten darin, dass das Programm bei der Ausführung nicht vom Warteschlangenmanager überwacht wird. Befehlsserviceobjekte sind Definitionen von Programmen mit einer kurzen Lebensdauer, mit denen eine bestimmte Aufgabe wie beispielsweise das Starten einer oder mehrerer Tasks ausgeführt wird.

## Zugehörige Konzepte

[Mit Services arbeiten](#)

## Topic-Objekte

Ein *Themenobjekt* ist ein WebSphere MQ-Objekt, mit dem Themen bestimmte, nicht standardmäßige Attribute zugewiesen werden können.

Ein *Thema* wird von einer Anwendung definiert, die für eine bestimmte *Themenzeichenfolge* eine Veröffentlichung oder Subskription durchführt. Eine Themenzeichenfolge kann eine Hierarchie von Themen angeben, indem die Themen durch einen Schrägstrich (/) voneinander getrennt werden. Dies kann durch eine *Themenstruktur* dargestellt werden. Wenn eine Anwendung beispielsweise in den Themenzeichenfolgen `/Sport/American Football` und `/Sport/Soccer` veröffentlicht, wird eine Themenstruktur erstellt, die einen übergeordneten Knoten `Sport` mit zwei untergeordneten Elementen, `American Football` und `Soccer`, enthält.

Themen übernehmen ihre Attribute vom ersten übergeordneten Verwaltungsknoten, der in ihrer Themenstruktur gefunden wird. Wenn eine bestimmte Themenstruktur keine administrativen Themenknoten enthält, übernehmen alle Themen ihre Attribute vom Basisthemenobjekt SYSTEM.BASE.TOPIC.

Sie können jedem Knoten einer Themenstruktur ein Themenobjekt hinzufügen, indem Sie die Themenzeichenfolge des betreffenden Knotens im Attribut TOPICSTR des Themenobjekts angeben. Sie können auch weitere Attribute für den administrativen Themenknoten definieren. Weitere Informationen über diese Attribute finden Sie unter MQSC-Befehle oder unter [Verwaltungstasks automatisieren](#). Jedes Themenobjekt übernimmt standardmäßig seine Attribute von dem übergeordneten administrativen Themenknoten, der ihm am nächsten liegt.

Themenobjekte können auch dazu verwendet werden, die vollständige Themenstruktur vor Anwendungsentwicklern zu verbergen. Wenn ein Themenobjekt mit dem Namen FOOTBALL.US für das Thema / Sport/American Football erstellt wird, kann eine Anwendung Veröffentlichungen für das Objekt FOOTBALL.US statt für die Zeichenfolge /Sport/American Football durchführen oder abonnieren; das Ergebnis ist dasselbe.

Wenn Sie in einer Themenzeichenfolge für ein Themenobjekt das Zeichen #, +, / oder \* eingeben, wird es wie ein normales Zeichen innerhalb der Zeichenfolge behandelt, d. h., es wird als Teil der Themenzeichenfolge betrachtet, die einem Themenobjekt zugeordnet ist.

Weitere Informationen zu Themenobjekten finden Sie im Abschnitt [Einführung in WebSphere MQ Publish/Subscribe-Messaging](#).

## IBM WebSphere MQ-Objekte benennen

Die Namenskonvention für die Benennung von WebSphere MQ-Objekten hängt vom jeweiligen Objekt ab. Auch für die Namen der Maschinen und die Benutzer-IDs für IBM WebSphere MQ gelten einige Einschränkungen.

Jede Instanz eines Warteschlangenmanagers hat einen eigenen Namen. Dieser Name muss im Netz bzw. unter den verbundenen Warteschlangenmanagern eindeutig sein, damit ein Warteschlangenmanager eindeutig den Ziel-Warteschlangenmanager angeben kann, an den eine Nachricht gesendet werden soll.

Bei den anderen Objekttypen ist jedem Objekt ein Name zugewiesen, über den es identifiziert werden kann. Diese Namen müssen innerhalb eines Warteschlangenmanagers und innerhalb eines Objekttyps eindeutig sein. So können beispielsweise eine Warteschlange und ein Prozess denselben Namen haben, nicht jedoch zwei Warteschlangen.

In WebSphere MQ können Namen eine Länge von bis zu 48 Zeichen haben; ausgenommen hiervon sind *Kanalnamen* die maximal eine Länge von 20 Zeichen haben können. Weitere Informationen zur Benennung von IBM WebSphere MQ-Objekten finden Sie im Abschnitt [„Regeln für die Benennung von IBM WebSphere MQ-Objekten“](#) auf Seite 132.

Für die Namen der Maschinen und die Benutzer-IDs für IBM WebSphere MQ gelten ebenfalls einige Einschränkungen:

- Achten Sie darauf, dass der Maschinenname keine Leerzeichen enthält. IBM WebSphere MQ unterstützt keine Maschinennamen, die Leerzeichen enthalten. Wenn Sie IBM WebSphere MQ auf eine Maschine mit einem solchen Namen installieren, können Sie keine Warteschlangenmanager erstellen.
- Bei IBM WebSphere MQ-Berechtigungen dürfen die Namen der Benutzer-IDs und -Gruppen nicht länger als 20 Zeichen sein (Leerzeichen sind nicht zulässig).
- Ein WebSphere MQ for Windows-Server unterstützt keine Verbindung mit einem Windows-Client, der unter einer Benutzer-ID ausgeführt wird, die das Zeichen @ enthält, zum Beispiel abc@d.

### Zugehörige Konzepte

[„IBM WebSphere MQ-Dateinamen“](#) auf Seite 134

Die einzelnen Warteschlangenmanager, Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services und Authentifizierungsdatenobjekte in WebSphere MQ werden jeweils durch eine Datei dargestellt. Da Objektnamen nicht notwendigerweise auch gültige Dateinamen sind, wandelt der Warteschlangenmanager die Objektnamen bei Bedarf in gültige Dateinamen um.

## Zugehörige Verweise

„Regeln für die Benennung von IBM WebSphere MQ-Objekten“ auf Seite 132

Für die Namen von IBM WebSphere MQ-Objekten gelten Längeneinschränkungen. Außerdem wird bei diesen Namen die Groß-/Kleinschreibung beachtet. Nicht alle Zeichen werden für jeden Objekttyp unterstützt, und bei vielen Objekten gelten Regeln hinsichtlich der Eindeutigkeit der Namen.

## **Regeln für die Benennung von IBM WebSphere MQ-Objekten**

Für die Namen von IBM WebSphere MQ-Objekten gelten Längeneinschränkungen. Außerdem wird bei diesen Namen die Groß-/Kleinschreibung beachtet. Nicht alle Zeichen werden für jeden Objekttyp unterstützt, und bei vielen Objekten gelten Regeln hinsichtlich der Eindeutigkeit der Namen.

Es gibt viele verschiedene Arten von IBM WebSphere MQ-Objekten, wobei Objekte unterschiedlicher Typen gleiche Namen haben können, da sie in getrennten Namensbereichen verwaltet werden. Eine lokale Warteschlange und ein Senderkanal können zum Beispiel den gleichen Namen haben. Ein Objekt darf jedoch nicht den Namen eines anderen Objekts des gleichen Namensbereichs haben. Eine lokale Warteschlange darf zum Beispiel nicht den gleichen Namen wie eine Modellwarteschlange haben, und ein Senderkanal darf nicht den gleichen Namen wie ein Empfängerkanal haben.

Die folgenden IBM WebSphere MQ-Objekte liegen in separaten Namensbereichen vor:

- Authentifizierungsdaten
- Kanal
- Clientkanal
- Empfangsprogramm
- Namensliste
- Prozess
- Warteschlange
- Service
- Speicherklasse
- Subskription
- Thema

## **Zeichenlänge von Objektnamen**

Im Allgemeinen dürfen die Namen von IBM WebSphere MQ-Objekten bis zu 48 Zeichen lang sein. Diese Regel gilt für die folgenden Objekte:

- Authentifizierungsdaten
- Cluster
- Empfangsprogramm
- Namensliste
- Prozessdefinition
- Warteschlange
- Warteschlangenmanager
- Service
- Subskription
- Thema




Allerdings gibt es hierzu auch Einschränkungen:

1. Die Namen von Kanalobjekten und Clientverbindungskanälen dürfen maximal 20 Zeichen lang sein. Weitere Informationen zu Kanälen finden Sie im Abschnitt Kanäle definieren.

2. Themenzeichenfolgen dürfen maximal 10240 Byte lang sein. Bei allen Namen von IBM WebSphere MQ-Objekten wird die Groß-/Kleinschreibung beachtet.
3. Die Namen von Speicherklassen dürfen maximal 8 Zeichen lang sein.
4. Die Namen von Coupling-Facility-(CF-)Strukturen dürfen maximal 12 Zeichen lang sein.

## Zeichen in Objektnamen

Für die Namen von IBM WebSphere MQ-Objekten sind folgende Zeichen gültig:

Zeichen	Einschränkungen
Großbuchstaben (A - Z)	• --
Kleinbuchstaben (a - z)	<ul style="list-style-type: none"> <li>• In MQSC-Scripts müssen Namen mit Kleinbuchstaben in einfache Anführungszeichen eingeschlossen werden, da die Kleinbuchstaben ansonsten in Großbuchstaben umgewandelt werden.</li> <li>• Systeme, auf denen EBCDIC Katakana verwendet wird, unterstützen in Objektnamen keine Kleinbuchstaben.</li> </ul>
Numerische Zeichen (0 - 9)	• --
Punkt (.)	• --
Unterstrich (_)	•  --
Schrägstrich (/)	•  Unter Windows darf das erste Zeichen eines Warteschlangenmanagersnamens kein Schrägstrich sein.
Prozentzeichen (%)	•  --

Darüber hinaus gelten für die in Objektnamen verwendeten Zeichen einige generelle Regeln:

1. Führende oder eingebettete Leerzeichen sind nicht erlaubt.
2. Landessprachliche Zeichen sind nicht erlaubt.
3. Namen, die kürzer als die Feldlänge sind, können am Ende mit Leerzeichen aufgefüllt werden. Alle zu kurzen Namen, die vom Warteschlangenmanager zurückgegeben werden, sind am Ende mit Leerzeichen aufgefüllt.

## Warteschlangenamen

Der Name einer Warteschlange besteht aus zwei Teilen:

- Der Name eines Warteschlangenmanagers
- Aus dem lokalen Namen der Warteschlange, wie er dem Warteschlangenmanager bekannt ist

Jeder Teil des Warteschlangennamens ist 48 Zeichen lang.

Bei der Angabe einer lokalen Warteschlange können Sie den Namen des Warteschlangenmanagers weglassen (durch Ersetzen seines Namens durch Leerzeichen oder durch eine führende Null). Alle Warteschlangennamen, die von IBM WebSphere MQ an ein Programm zurückgegeben werden, enthalten jedoch den Namen des Warteschlangenmanagers.

Zur Angabe einer fernen Warteschlange muss ein Programm im vollständigen Warteschlangennamen den Namen des Warteschlangenmanagers angeben (es sei denn, es liegt eine lokale Definition der fernen Warteschlange vor).

Der von einer Anwendung angegebene Warteschlangename kann der Name (oder der Aliasname) einer lokalen Warteschlange oder der Name einer lokalen Definition einer fernen Warteschlange sein. Die Anwendung selbst muss nicht wissen, um welche Art von Warteschlange es sich handelt, es sei denn, sie erwartet eine Nachricht aus dieser Warteschlange (in diesem Fall muss es sich um eine lokale Warteschlange handeln). Wenn die Anwendung das Warteschlangenobjekt öffnet, löst der Aufruf MQOPEN den Namen auf, wodurch festgestellt wird, an welcher Warteschlange die nachfolgenden Operationen durchgeführt werden sollen. Dieser Mechanismus ist insofern signifikant, als dass die Anwendung inhärent nicht von bestimmten Warteschlangen abhängig ist, die an bestimmten Orten im Netz der Warteschlangenmanager definiert sind. Falls ein Systemadministrator die Warteschlangen im Netz umstellt und deren Definitionen ändert, hat dies den Vorteil, dass die Anwendungen, die auf diese Warteschlangen zugreifen, nicht geändert werden müssen.

## Reservierte Objektnamen

Objektnamen, die mit SYSTEM beginnen, sind für vom Warteschlangenmanager definierte Objekte reserviert. Mit den Befehlen **Alter**, **Define** und **Replace** können Sie diese Objektdefinitionen an Ihre Installation anpassen. Eine vollständige Liste der für IBM WebSphere MQ definierten Namen finden Sie im Abschnitt [Warteschlangennamen](#).

### Zugehörige Tasks

[Auswählen eines Installationsnamens](#)

### IBM WebSphere MQ-Dateinamen

Die einzelnen Warteschlangenmanager, Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services und Authentifizierungsdatenobjekte in WebSphere MQ werden jeweils durch eine Datei dargestellt. Da Objektnamen nicht notwendigerweise auch gültige Dateinamen sind, wandelt der Warteschlangenmanager die Objektnamen bei Bedarf in gültige Dateinamen um.

Der Standardpfad eines Warteschlangenmanager-Verzeichnisses setzt sich wie folgt zusammen:

- Ein Präfix, das in den WebSphere MQ-Konfigurationsdaten definiert wird:
  - Auf Windows -32-Bit-Systemen ist das Standardpräfix C:\Program Files\IBM\WebSphere MQ. Auf Windows 64-Bit-Systemen lautet das Standardpräfix C:\Program Files\IBM\WebSphere MQ (x86)\. Es wird in der Zeilengruppe DefaultPrefix der Konfigurationsdatei 'mqs.ini' konfiguriert.
  - Auf Systemen mit UNIX and Linux ist /var/mqm das Standardpräfix. Es wird in der Zeilengruppe DefaultPrefix der Konfigurationsdatei 'mqs.ini' konfiguriert.

Sofern verfügbar, kann das Präfix über die WebSphere MQ-Eigenschaftenseite in IBM WebSphere MQ Explorer geändert werden; andernfalls müssen Sie das Präfix in der Konfigurationsdatei mqs.ini manuell ändern.

- Der Name des Warteschlangenmanagers wird in einen gültigen Verzeichnisnamen umgesetzt. Der Warteschlangenmanager

```
queue.manager
```

beispielsweise wird dargestellt als

```
queue!manager
```

Dieser Vorgang wird als *Namensumsetzung* bezeichnet.

In WebSphere MQ können Sie einem Warteschlangenmanager einen Namen mit bis zu 48 Zeichen geben.

Beispielsweise kann ein Warteschlangenmanager den folgenden Namen erhalten:

Jeder Warteschlangenmanager wird jedoch anhand einer Datei dargestellt; für Dateinamen bestehen sowohl Längenbeschränkungen als auch Vorgaben für die Zeichen, die verwendet werden können. Daher werden die Namen der Dateien für Objekte automatisch entsprechend den Anforderungen des Dateisystems umgewandelt.

Für die Umsetzung von Warteschlangenmanager-Namen gelten die folgenden Regeln:

1. Umsetzung einzelner Zeichen:
  - Von . !
  - Von / in &
2. Ist das Ergebnis immer noch kein gültiger Name:
  - a. Begrenzen Sie die Länge des Namens auf acht Zeichen.
  - b. Hängen Sie ein dreistelliges numerisches Suffix an.

Wird beispielsweise das Standardpräfix verwendet und lautet der Warteschlangenmanager-Name `queue.manager`, gilt Folgendes:

- In WebSphere MQ for Windows mit NTFS oder FAT32 wird der Warteschlangenmanagername wie folgt umgesetzt:

```
c:\Program Files\IBM\WebSphere MQ\mqmgs\queue!manager
```

- In WebSphere MQ for Windows mit FAT wird der Warteschlangenmanagername wie folgt umgesetzt:

```
c:\Program Files\IBM\WebSphere MQ\mqmgs\queue!ma
```

- In WebSphere MQ-Systemen für jUNIX and Linux wird der Warteschlangenmanagername wie folgt umgesetzt:

```
/var/mqm/mqmgs/queue!manager
```

Der Umsetzungsalgorithmus unterscheidet auch in Dateisystemen, bei denen die Groß-/Kleinschreibung keine Rolle spielt, zwischen Dateinamen, die sich nur in der Groß-/Kleinschreibung unterscheiden.

## Objektnamensumsetzung

Bei Objektnamen handelt es sich nicht notwendigerweise um gültige Dateisystemnamen. Daher müssen Sie Objektnamen unter Umständen umwandeln. Die Vorgehensweise unterscheidet sich von der Umsetzung von Warteschlangenmanager-Namen, da auf jeder Maschine nur wenige Warteschlangenmanager-Namen, jedoch unter Umständen eine große Anzahl von Objekten pro Warteschlangenmanager vorhanden sind. Warteschlangen, Prozessdefinitionen, Namenslisten, Kanäle, Clientverbindungskanäle, Empfangsprogramme, Services und Authentifizierungsdatenobjekte sind im Dateisystem dargestellt.

Die vom Umsetzungsprozess generierten neuen Namen sind keine direkte Entsprechung des ursprünglichen Objektnamens. Mit dem Befehl `dspmqls` können die ursprünglichen Namen umgesetzter Objektnamen angezeigt werden.

## Objekte verwalten

Dieser Abschnitt enthält eine Übersicht über das Erstellen, Ändern, Anzeigen und Löschen von Objekten.


Weitere Informationen finden Sie im Abschnitt „Objekte“ auf Seite 112.

Mit Ausnahme dynamischer Warteschlangen müssen diese Objekte im Warteschlangenmanager definiert sein, damit sie verwendet werden können.




Objekte können wie folgt definiert und verwaltet werden:

- Mit den in den Abschnitten [Referenz für Programmable Command Formats](#) und [Verwaltungsaufgaben automatisieren](#) beschriebenen PCF-Befehlen (PCF = Programmable Command Format).
- Mit den im Abschnitt [MQSC-Befehle](#) beschriebenen MQSC-Befehlen.
- WebSphere MQ Explorer (nur Windows, UNIX und Linux für Intel -Systeme)

Weitere Möglichkeiten für die Objektverwaltung sind:

- Steuerbefehle, die über die Tastatur eingegeben werden. Informationen hierzu finden Sie im Abschnitt [Steuerbefehle](#).
- MQAI-Aufrufe (IBM WebSphere MQ Administration Interface) in einem Programm. Siehe [WebSphere MQ-Verwaltungsschnittstelle \(MQAI\)](#).
-  Nur unter IBM WebSphere MQ for Windows:
  - MQAI-COM-Aufrufe (Component Object Model) in einem Programm.
  - Windows Default Configuration Application (Anwendung für Standardkonfiguration).

Sie können auch die Attribute von Objekten anzeigen oder ändern oder die Objekte löschen.

   Für häufig verwendete WebSphere MQ-Befehlsfolgen auf Windows- und UNIX and Linux-Systemen können Sie mithilfe der MQSC-Funktion Befehlsfolgen in Dateien ausführen.

## Objektattribute

Die Eigenschaften eines Objektes werden über die Objektattribute definiert. Einige dieser Attribute können Sie angeben, andere hingegen können nur angezeigt werden.

Das Attribut *MaxMsgLength* einer Warteschlange gibt beispielsweise die maximal mögliche Länge für Nachrichten an, die in dieser Warteschlange eingereiht werden können. Das Attribut *DefinitionType* gibt an, wie die Warteschlange erstellt wurde; dieses Attribut kann nur angezeigt werden.

In WebSphere MQ gibt es für den Verweis auf Attribute zwei Möglichkeiten:

- Über den PCF-Namen des Attributs (beispielsweise *MaxMsgLength*).
- Über den WebSphere MQ-Scriptbefehlsnamen des Attributs (beispielsweise MAXMSGL).

In diesem Handbuch wird vorrangig die Angabe von Attributen über WebSphere MQ-Scriptbefehle beschrieben; daher sind eher die WebSphere MQ-Scriptbefehlsnamen als die PCF-Namen der Attribute aufgeführt.

## Cluster

Sie können Warteschlangenmanager in einem Cluster anordnen. Jeder Warteschlangenmanager in einem Cluster kann die ihm zugeordneten Warteschlangen den anderen Warteschlangenmanagern im Cluster zugänglich machen. Ebenso kann jeder Warteschlangenmanager Nachrichten an die anderen Warteschlangenmanager im Cluster senden, ohne dass hierfür viele der Objektdefinitionen erforderlich sind, die standardmäßig bei einer verteilten Steuerung von Warteschlangen benötigt werden.

In einem konventionellen WebSphere MQ-Netz mit verteilter Steuerung von Warteschlangen sind die einzelnen Warteschlangenmanager unabhängig voneinander. Wenn ein Warteschlangenmanager Nachrichten an einen anderen WS-Manager senden muss, muss er eine Übertragungswarteschlange, einen Kanal zum fernen Warteschlangenmanager und eine Definition einer fernen Warteschlange für jede Warteschlange definieren, an die Nachrichten gesendet werden sollen.

Ein *Cluster* ist eine Gruppe von Warteschlangenmanager, die so konfiguriert sind, dass sie über ein Netz direkt miteinander kommunizieren können; es müssen keine Übertragungswarteschlangen, Kanäle und Definitionen ferner Warteschlangen definiert werden.

Jeder Warteschlangenmanager im Cluster verfügt über eine einzige Übertragungswarteschlange, über die er Nachrichten an jeden anderen Warteschlangenmanager im Cluster senden kann.



## Zugehörige Konzepte

[Cluster entwerfen](#)

An dieser Stelle werden das Konzept der Cluster und ihre Funktionsweise erläutert.

## Zugehörige Tasks

[WS-Manager-Cluster konfigurieren](#)

[Neuen Cluster einrichten](#)

## Systemstandardwertobjekte

Dieser Abschnitt enthält eine Einführung in Systemstandardobjekte sowie einen Link zu weiteren Informationen.

*Systemstandardobjekte* sind Objektdefinitionen, die bei der Erstellung eines Warteschlangenmanagers automatisch erstellt werden. Sie können diese Objektdefinitionen für die Verwendung in den Anwendungen Ihrer Installation kopieren und entsprechend ändern.

Standardobjektnamen beginnen mit SYSTEM; die lokale Standardschlange beispielsweise hat die Bezeichnung SYSTEM.DEFAULT.LOCAL.QUEUE, der Standardempfängerkanal die Bezeichnung SYSTEM.DEF.RECEIVER. Die Standardobjekte mit diesen Namen sind erforderlich; sie können nicht umbenannt werden.

Alle Attribute, die bei der Definition eines Objekts nicht angegeben werden, werden explizit aus dem entsprechenden Standardobjekt kopiert. Wenn Sie beispielsweise eine lokale Warteschlange definieren, werden die von Ihnen nicht angegebenen Attribute aus der Standardwarteschlange SYSTEM.DEFAULT.LOCAL.QUEUE übernommen.

Weitere Informationen zu den Systemstandardwerten finden Sie im Abschnitt [System- und Standardobjekte](#).

## IBM WebSphere MQ Multicasting

IBM WebSphere MQ Multicast bietet eine geringe Latenzzeit, eine hohe Ausgabefächerung und eine zuverlässige Multicasting-Nachrichtenübertragung.

Bei Multicast handelt es sich um eine effiziente Form der Publish/Subscribe-Nachrichtenübermittlung, da eine große Zahl von Subskribenten zugeordnet werden kann, ohne dass es zu negativen Auswirkungen beim Leistungsverhalten kommt. WebSphere MQ ermöglicht eine zuverlässige Multicast-Nachrichtenübermittlung, bei der über Bestätigungen, negative Rückmeldungen und Folgenummern eine niedrige Latenz bei der Nachrichtenübermittlung mit einer hohen Ausgabefächerung erreicht wird.

Die ausreichende Übermittlung von WebSphere MQ Multicast ermöglicht eine fast simultane Übermittlung, durch die sichergestellt wird, dass kein Empfänger einen Vorteil hat. Da WebSphere MQ Multicast für die Übermittlung von Nachrichten ein Netz verwendet, wird für die Ausgabefächerung der Daten keine Publish/Subscribe-Engine benötigt. Nach der Zuordnung eines Themas zu einer Gruppenadresse ist kein Warteschlangenmanager erforderlich, da Publisher und Subskribenten in einem Peer-to-Peer-Modus arbeiten können. Dies ermöglicht die Reduzierung der Arbeitslast auf Warteschlangenmanager-Servern und der Warteschlangenmanager-Server ist keine potenzielle Fehlerquelle mehr.

## Ursprüngliche Multicasting-Konzepte

WebSphere MQ Multicast kann unter Verwendung des Kommunikationsinformationsobjekts COMMINFO problemlos in bestehende Systeme und Anwendungen integriert werden. Zwei TOPIC-Objektfelder ermöglichen die schnelle Konfiguration bestehender TOPIC-Objekte zur Unterstützung bzw. zum Ignorieren von Multicasting-Datenverkehr.

## Für Multicasting erforderliche Objekte

Nachfolgend finden Sie eine kurze Übersicht über die beiden Objekte, die für WebSphere MQ Multicast erforderlich sind:

### **COMMINFO-Objekt**

Das COMMINFO-Objekt enthält die Attribute, die der Multicasting-Übertragung zugeordnet sind. Weitere Informationen zu den Parametern des COMMINFO-Objekts finden Sie unter [DEFINE COMMINFO](#).

Das einzige COMMINFO-Feld, das angegeben werden MUSS, ist das Feld mit dem Namen des COMMINFO-Objekts. Mit diesem Namen wird das COMMINFO-Objekt für ein Thema ermittelt. Prüfen Sie das Feld **GRPADDR** des COMMINFO-Objekts, um sicherzustellen, dass der Wert eine gültige Multicastgruppenadresse ist.

### **TOPIC-Objekt**

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. Weitere Informationen zu den Parametern des TOPIC-Objekts finden Sie unter [DEFINE TOPIC](#).

Bestehende Themen können für das Multicasting verwendet werden, indem Sie die Werte der folgenden TOPIC-Objektparameter ändern: **COMMINFO** und **MCAST**.

- **COMMINFO** Dieser Parameter gibt den Namen des Multicastkommunikationsinformationsobjekts an.
- **MCAST** Dieser Parameter gibt an, ob Multicasting an dieser Position in der Themenstruktur zulässig ist. Standardmäßig ist **MCAST** auf **ASPARENT** gesetzt, was bedeutet, dass das Multicast-Attribut des Themas vom übergeordneten Element übernommen wird. Wenn Sie **MCAST** auf **ENABLED** setzen, ist auf diesem Knoten Multicasting-Datenverkehr möglich.

## **Multicasting-Netze und -Themen**

Nachfolgend finden Sie eine Übersicht darüber, was mit Subskriptionen mit unterschiedlichen Subskriptionstypen und Themendefinitionen geschieht. Alle folgenden Beispiele setzen voraus, dass der Parameter **COMMINFO** des TOPIC-Objekts auf den Namen eines gültigen COMMINFO-Objekts gesetzt wurde:

### **Thema für Multicasting aktiviert (MCAST ENABLED)**

Wenn der Parameter der Themenzeichenfolge **MCAST** auf **ENABLED** gesetzt ist, sind Subskriptionen Multicasting-fähiger Clients zugelassen und eine Multicasting-Subskription wird vorgenommen – mit folgender Ausnahme:

- Es handelt sich um die permanente Subskription eines Multicasting-fähigen Clients.
- Es handelt sich um eine nicht verwaltete Subskription eines Multicasting-fähigen Clients.
- Es handelt sich um die Subskription eines nicht Multicasting-fähigen Clients.

In diesen Fällen wird eine andere Subskription als eine Multicast-Subskription vorgenommen und Subskriptionen werden in ein normales Publish/Subscribe herabgestuft.

### **Thema für Multicasting inaktiviert**

Wenn der Parameter der Themenzeichenfolge **MCAST** auf **DISABLED** gesetzt ist, wird immer eine Nicht-Multicasting-Subskription vorgenommen und Subskriptionen werden in ein normales Publish/Subscribe herabgestuft.

### **Thema nur für Multicasting aktiviert**

Wenn der Parameter der Themenzeichenfolge **MCAST** auf **ONLY** gesetzt ist, sind Subskriptionen Multicasting-fähiger Clients zugelassen und eine Multicasting-Subskription wird vorgenommen – mit folgender Ausnahme:

- Es handelt sich um eine permanente Subskription: Permanente Subskriptionen werden mit dem Ursachencode [2436 \(0984\) \(RC2436\): MQRC\\_DURABILITY\\_NOT\\_ALLOWED](#) abgelehnt.
- Es handelt sich um eine nicht verwaltete Subskription: Nicht verwaltete Subskriptionen werden mit dem Ursachencode [2046 \(07FE\) \(RC2046\): MQRC\\_OPTIONS\\_ERROR](#) abgelehnt.
- Es handelt sich um eine Subskription von einem nicht Multicast-fähigen Client: Diese Subskriptionen werden mit dem Ursachencode [2560 \(0A00\) \(RC2560\): MQRC\\_MULTICAST\\_ONLY](#) abgelehnt.
- Es handelt sich um eine Subskription aus einer lokal gebundenen Anwendung: Diese Subskriptionen werden mit dem Ursachencode [2560 \(0A00\) \(RC2560\): MQRC\\_MULTICAST\\_ONLY](#) abgelehnt.

## Sicherheit

In IBM WebSphere MQ gibt es mehrere Möglichkeiten, für Sicherheit zu sorgen: die Berechtigungsschnittstelle, Kanalexits (benutzerdefiniert oder von einem anderen Hersteller), SSL-Kanalsicherheit, Kanalauthentifizierungsdatensätze und Nachrichtensicherheit.

### Schnittstelle für Berechtigungsservice

Die Berechtigung für die Verwendung von MQI-Aufrufen und Befehlen und den Zugriff auf Objekte wird vom **Objektberechtigungsmanager** erteilt, der standardmäßig aktiviert ist. Der Zugriff auf IBM WebSphere MQ-Entitäten wird über IBM WebSphere MQ-Benutzergruppen und den Objektberechtigungsmanager gesteuert. Administratoren können Berechtigungen über eine Befehlszeilenschnittstelle nach Bedarf erteilen oder entziehen.

Weitere Informationen zum Erstellen von Berechtigungsservicekomponenten finden Sie im Abschnitt [Sicherheit auf Windows- und UNIX and Linux-Systemen einrichten](#).

### Benutzerdefinierte Kanalexits und Kanalexits anderer Anbieter

Kanäle können benutzerdefinierte Kanalexits oder Kanalexits anderer Anbieter verwenden. Weitere Informationen finden Sie im Abschnitt [Kanalexitprogramme für Messaging-Kanäle](#).

### Kanalsicherheit über SSL

Das SSL-Protokoll (Secure Sockets Layer) sorgt für standardisierte Kanalsicherheit, indem es Schutz vor Abhören, Manipulation und Identitätsvortäuschung bietet.

Bei SSL werden der Nachrichtenschutz sowie Integrität und gegenseitige Authentifizierung mithilfe von öffentlichen Schlüsseln und symmetrischer Verschlüsselung implementiert.

Umfassende Informationen zur Sicherheit in IBM WebSphere MQ, darunter auch ausführliche Hinweise zu SSL, finden Sie im Abschnitt [Sicherheit](#). Eine Übersicht über SSL mit Hinweisen zu den in diesem Abschnitt beschriebenen Befehlen finden Sie im Abschnitt [Kryptografische Sicherheitsprotokolle: SSL und TLS](#).

### Kanalauthentifizierungsdatensätze

Mit Kanalauthentifizierungsdatensätzen können Sie den Zugriff auf Systeme, die eine Verbindung herstellen können, auf Kanalebene steuern. Weitere Informationen finden Sie im Abschnitt [Kanalauthentifizierungsdatensätze](#).

### Nachrichtensicherheit

Verwenden Sie Erweiterte Nachrichtensicherheit eine gesondert installierte und lizenzierte Komponente von IBM WebSphere MQ, um mit IBM WebSphere MQ gesendete und empfangene Nachrichten mit einem kryptografischen Schutz zu versehen. Weitere Informationen finden Sie im Abschnitt [WebSphere MQ Advanced Message Security](#).

#### Zugehörige Tasks

[Sicherheit](#)

[Sicherheitsanforderungen planen](#)

## Clients und Server

Dieser Abschnitt enthält allgemeine Hinweise zur IBM WebSphere MQ-Unterstützung von Client/Server-Konfigurationen für Anwendungen.

Ein IBM WebSphere MQ MQI-*Client* ist eine Komponente, die es einer auf einem System aktiven Anwendung ermöglicht, MQI-Aufrufe an Warteschlangenmanager auf einem anderen System abzusetzen. Die Ausgabe eines Aufrufs wird an den Client zurückgesendet, der sie an die Anwendung weitergibt.

Ein IBM WebSphere MQ-Server ist ein Warteschlangenmanager, der für einen oder mehrere Clients Services zur Steuerung von Warteschlangen bereitstellt. Alle IBM WebSphere MQ-Objekte (wie beispielsweise Warteschlangen) sind nur auf der Warteschlangenmanager-Maschine (IBM WebSphere MQ-Servermaschine) vorhanden, nicht auf dem Client. Ein IBM WebSphere MQ-Server kann auch lokale IBM WebSphere MQ-Anwendungen unterstützen.

Ein IBM WebSphere MQ-Server unterscheidet sich von einem normalen Warteschlangenmanager darin, dass zwischen dem Server und den einzelnen Clients dedizierte Kommunikationsverbindungen bestehen. Weitere Informationen zur Erstellung von Kanälen für Clients und Server finden Sie im Abschnitt [Anwendungen mithilfe der verteilten Steuerung von Warteschlangen verbinden](#).

Allgemeine Informationen zu Clients finden Sie im Abschnitt [„Überblick über IBM WebSphere MQ MQI-Clients“](#) auf Seite 140.

## IBM WebSphere MQ-Anwendungen in einer Client/Server-Umgebung

Wenn eine Verbindung zu einem Server besteht, können IBM WebSphere MQ-Clientanwendungen die meisten MQI-Aufrufe auf dieselbe Weise wie lokale Anwendungen ausgeben. Die Clientanwendung gibt einen MQCONN-Aufruf aus, um eine Verbindung zu einem angegebenen Warteschlangenmanager herzustellen. Alle weiteren MQI-Aufrufe, in denen die von der Verbindungsanforderung zurückgegebene Verbindungskennung angegeben ist, werden von diesem Warteschlangenmanager verarbeitet.

Sie müssen die Anwendungen mit den entsprechenden Clientbibliotheken verbinden. Weitere Informationen finden Sie im Abschnitt [Anwendungen für WebSphere MQ-Clients erstellen](#).

### Zugehörige Konzepte

[„Transaktionsmanagement und -unterstützung“](#) auf Seite 147

In diesem Abschnitt wird eine Übersicht über das Transaktionsmanagement gegeben und erläutert, wie Transaktionen in WebSphere MQ unterstützt werden.

[„Funktionen des Warteschlangenmanagers erweitern“](#) auf Seite 149

Sie können die Funktionen eines Warteschlangenmanagers durch Benutzerexits, API-Exits oder installierbare Services erweitern.

## Überblick über IBM WebSphere MQ MQI-Clients

Der *WebSphere MQ MQI-Client* ist eine Komponente von IBM WebSphere MQ, die auf dem System installiert werden kann, auf dem kein Warteschlangenmanager aktiv ist.

Mithilfe eines IBM WebSphere MQ MQI-Clients kann eine Anwendung, die auf demselben System wie der Client ausgeführt wird, eine Verbindung zu einem Warteschlangenmanager herstellen, der auf einem anderen System ausgeführt wird. Dadurch kann die Anwendung MQI-Aufrufe an diesen Warteschlangenmanager absetzen. Diese Anwendungen werden als *WebSphere MQ MQI-Clientanwendungen*, der Warteschlangenmanager als *Server-Warteschlangenmanager* bezeichnet.

Eine IBM WebSphere MQ MQI-Clientanwendung und ein Server-Warteschlangenmanager kommunizieren über einen *MQI-Kanal* miteinander. Ein MQI-Kanal wird gestartet, wenn die Clientanwendung einen **MQCONN**- oder **MQCONNX**-Aufruf absetzt, um eine Verbindung zum Warteschlangenmanager herzustellen, und endet, wenn die Clientanwendung einen **MQDISC**-Aufruf ausgibt, um die Verbindung zum Warteschlangenmanager zu beenden. Die Eingabeparameter eines MQI-Aufrufs werden im MQI-Kanal in eine Richtung, die Ausgabeparameter in die entgegengesetzte Richtung übertragen.

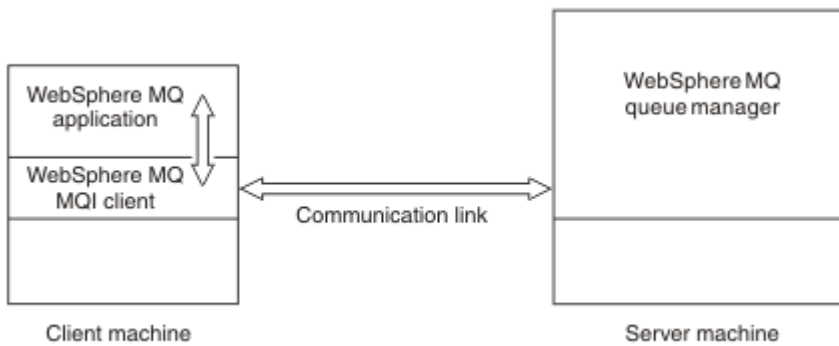


Abbildung 30. Verbindung zwischen Client und Server

Folgende Plattformen können eingesetzt werden. Die Kombinationen hängen davon ab, welches IBM WebSphere MQ -Produkt Sie verwenden, und werden in [„Plattformunterstützung für IBM WebSphere MQ-Clients“](#) auf Seite 143 beschrieben.

#### IBM WebSphere MQ MQI-Client

UNIX and Linux  
Windows

HP Integrity NonStop Server

#### IBM WebSphere MQ-Server

UNIX and Linux  
Windows

HP Integrity NonStop Server

Die MQI ist für Anwendungen verfügbar, die auf der Clientplattform ausgeführt werden. Die Warteschlangen und andere IBM WebSphere MQ -Objekte befinden sich auf einem Warteschlangenmanager, den Sie auf dem Server installiert haben.

Eine Anwendung, die in der IBM WebSphere MQ MQI-Clientsumgebung ausgeführt werden soll, muss zuerst mit der entsprechenden Clientbibliothek verknüpft werden. Wenn die Anwendung einen MQI-Aufruf ausgibt, leitet der IBM WebSphere MQ MQI-Client die Anforderung an einen Warteschlangenmanager weiter, wo sie verarbeitet wird und von dem eine Antwort an den IBM WebSphere MQ MQI-Client zurückgesendet wird.

Die Verbindung zwischen der Anwendung und dem IBM WebSphere MQ MQI-Client wird dynamisch zur Laufzeit hergestellt.

Sie können Clientanwendungen auch mithilfe von IBM WebSphere MQ classes for .NET, IBM WebSphere MQ classes for Java oder IBM WebSphere MQ classes for Java Message Service (JMS) entwickeln. Sie können Java -und JMS -Clients auf UNIX, Linux und Windows -Plattformen verwenden. Die Verwendung von Java und JMS wird hier nicht beschrieben. Ausführliche Informationen zur Installation und Konfiguration sowie zur Verwendung von IBM WebSphere MQ classes for Java und IBM WebSphere MQ classes for JMS finden Sie unter [Using WebSphere MQ classes for Java](#) und [Using WebSphere MQ classes for JMS](#).

#### Zugehörige Konzepte

[„Verwendung von IBM WebSphere MQ-Clients“](#) auf Seite 142

Mit IBM WebSphere MQ-Clients können IBM WebSphere MQ-Nachrichtenübertragung und -Warteschlangensteuerung auf effiziente Weise implementiert werden.

[„Wie erstelle ich einen IBM WebSphere MQ ?“](#) auf Seite 143

In diesem Abschnitt wird beschrieben, wie ein Client eingerichtet wird.

[„Was ist ein erweiterter Transaktionsclient?“](#) auf Seite 144

Ein erweiterter transaktionsorientierter WebSphere MQ-Client kann unter der Steuerung eines externen Transaktionsmanagers Ressourcen aktualisieren, die von einem anderen Ressourcenmanager verwaltet werden.

„Verbindung vom Client zum Server herstellen“ auf Seite 145

Clients stellen über MQCONN- oder MQCONNX-Aufrufe eine Verbindung zum Server her; die Kommunikation erfolgt über einen Kanal.

### Verwendung von IBM WebSphere MQ-Clients

Mit IBM WebSphere MQ-Clients können IBM WebSphere MQ-Nachrichtenübertragung und -Warteschlangensteuerung auf effiziente Weise implementiert werden.

Sie können eine Anwendung einsetzen, die eine MQI und einen Warteschlangenmanager verwendet, die sich beide (physisch oder virtuell) auf unterschiedlichen Maschinen befinden. Hier die Vorteile eines solchen Szenarios:

- Es ist keine vollständige WebSphere MQ-Implementierung auf der Clientmaschine erforderlich.
- Die Hardwarevoraussetzungen für das Clientsystem werden reduziert.
- Die Anforderungen bezüglich der Systemverwaltung werden reduziert.
- Eine auf einem Client aktive WebSphere MQ-Anwendung kann eine Verbindung zu mehreren Warteschlangenmanagern auf unterschiedlichen Systemen herstellen.
- Es können alternative Kanäle mit unterschiedlichen Übertragungsprotokollen verwendet werden.

### Zugehörige Verweise

„Welche Anwendungen werden auf einem IBM WebSphere MQ MQI-Client ausgeführt?“ auf Seite 142  
In der Clientumgebung wird die MQI vollständig unterstützt.

„Plattformunterstützung für IBM WebSphere MQ-Clients“ auf Seite 143

IBM WebSphere MQ auf allen Serverplattformen akzeptiert Clientverbindungen von IBM WebSphere MQ MQI-Clients auf UNIX -oder Linux -Systemen und Windows.

*Welche Anwendungen werden auf einem IBM WebSphere MQ MQI-Client ausgeführt?*

In der Clientumgebung wird die MQI vollständig unterstützt.

Auf diese Weise kann fast jede WebSphere MQ -Anwendung zur Ausführung auf einem IBM WebSphere MQ MQI-Clientsystem konfiguriert werden, indem die Anwendung auf dem IBM WebSphere MQ MQI-Client mit der MQIC-Bibliothek und nicht mit der MQI-Bibliothek verknüpft wird. Die Ausnahmen sind:

- MQGET mit Signal.
- Anwendungen, bei denen eine Synchronisationspunktkoordination mit anderen Ressourcenmanagern erforderlich ist, benötigen einen erweiterten transaktionsorientierten Client.

Wenn Vorauslesen aktiviert ist, um den Durchsatz beim nicht permanenten Messaging zu optimieren, stehen nicht alle MQGET-Optionen zur Verfügung. In der folgenden Tabelle sind alle zulässigen Optionen aufgeführt; außerdem ist angegeben, ob sie zwischen MQGET-Aufrufen geändert werden können.

Tabelle 9. Zulässige MQGET-Optionen bei aktiviertem Vorauslesen			
	Bei aktiviertem Vorauslesen zulässig und kann zwischen MQGET-Aufrufen ausgetauscht werden	Bei aktiviertem Vorauslesen zulässig, kann aber nicht zwischen MQGET-Aufrufen ausgetauscht werden <sup>1</sup>	MQGET-Optionen, die bei aktiviertem Vorauslesen nicht zulässig sind <sup>2</sup>
MQGET MD-Werte	MsgId <sup>3</sup> CorrelId <sup>3</sup>	Encoding CodedCharSetId	
MQGET MQGMO-Optionen	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF QUIESCING MQGMO_BROWSE_FIRST <sup>4</sup> MQGMO_BROWSE_NEXT <sup>4</sup> MQGMO_BROWSE_MESSAGE _UNDER_CURSOR <sup>4</sup>	MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT MQGMO_LOGICAL_ORDER MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_AVAILABLE MQGMO_ALL_SEGMENTS_AVAILABLE MQGMO_MARK_BROWSE_HANDLE MQGMO_MARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_HANDLE MQGMO_UNMARKED_BROWSE_MSG MQGMO_PROPERTIES_FORCE_MQRFH2 MQGMO_NO_PROPERTIES MQGMO_PROPERTIES_IN_HANDLE MQGMO_PROPERTIES_COMPATIBILITY	MQGMO_SET_SIGNAL MQGMO_SYNCPOINT MQGMO_MARK_SKIP _BACKOUT MQGMO_MSG_UNDER _CURSOR <sup>4</sup> MQGMO_LOCK MQGMO_UNLOCK
MQGMO-Werte		MsgHandle	

1. Wenn diese Optionen zwischen MQGET-Aufrufen geändert werden, wird Ursachencode MQRC\_OPTIONS\_CHANGED zurückgegeben.

2. Wenn diese Optionen im ersten MQGET-Aufruf angegeben werden, wird das Vorauslesen inaktiviert. Werden diese Optionen in einem nachfolgenden MQGET-Aufruf angegeben, wird Ursachencode MQRC\_OPTIONS\_ERROR zurückgegeben.
3. Clientanwendungen müssen erkennen können, dass bei einer Änderung der Werte für 'MsgId' und 'CorrelId' zwischen MQGET-Aufrufen Nachrichten mit den früheren Werten möglicherweise bereits an den Client gesendet wurden und im Vorauslesepuffer des Clients verbleiben, bis sie verarbeitet (oder automatisch gelöscht) werden.
4. Der erste MQGET-Aufruf bestimmt, ob Nachrichten aus einer Warteschlange angezeigt oder abgerufen werden, wenn Vorauslesen aktiviert ist. Wenn die Anwendung versucht, Anzeige und Abruf zu kombinieren, wird Ursachencode MQRC\_OPTIONS\_CHANGED zurückgegeben.
5. MQGMO\_MSG\_UNDER\_CURSOR ist bei aktiviertem Vorauslesen nicht möglich. Nachrichten können angezeigt oder abgerufen werden, wenn Vorauslesen aktiviert ist, aber eine Kombination der beiden Funktionen ist nicht möglich.

Eine Anwendung, die auf einem IBM WebSphere MQ MQI-Client ausgeführt wird, kann gleichzeitig eine Verbindung zu mehr als einem Warteschlangenmanager herstellen oder einen Warteschlangenmanagernamen mit einem Stern (\*) in einem MQCONN- oder MQCONNX-Aufruf verwenden (siehe Beispiele im Abschnitt [IBM WebSphere MQ MQI-Clientanwendungen mit Warteschlangenmanagern verbinden](#)).

#### *Plattformunterstützung für IBM WebSphere MQ-Clients*

IBM WebSphere MQ akzeptiert Clientverbindungen von IBM WebSphere MQ MQI-Clients auf UNIX - oder Linux -Systemen und Windows.

Wird WebSphere MQ als *Basisprodukt und Server* installiert, werden Verbindungen von IBM WebSphere MQ MQI-Clients auf den folgenden Plattformen akzeptiert:

- HP Integrity NonStop Server
- UNIX and Linux-Systeme
- Windows

CCSID (ID des codierten Zeichensatzes) und Kommunikationsprotokoll der Clientverbindungen hängen von der jeweiligen Plattform ab.

### **Wie erstelle ich einen IBM WebSphere MQ ?**

In diesem Abschnitt wird beschrieben, wie ein Client eingerichtet wird.

Damit ein IBM WebSphere MQ MQI-Client eingerichtet werden kann, muss der IBM WebSphere MQ-Server, zu dem dieser Client eine Verbindung herstellen soll, installiert und aktiv sein. So richten Sie einen Client ein:

1. Stellen Sie sicher, dass für den IBM WebSphere MQ MQI-Client eine unterstützte Plattform vorhanden ist und die Hardware- und Softwarevoraussetzungen erfüllt sind. Eine Beschreibung der unterstützten Plattformen finden Sie im Abschnitt [„Plattformunterstützung für IBM WebSphere MQ-Clients“](#) auf [Seite 143](#).
2. Überlegen Sie sich, wie IBM WebSphere MQ auf der Client-Workstation installiert werden soll, und gehen Sie entsprechend den Anweisungen für die ausgewählte Kombination aus Client- und Serverplattformen vor. Die Installation wird im Abschnitt [IBM WebSphere MQ-Client installieren](#) beschrieben.
3. Stellen Sie sicher, dass die Kommunikationsverbindungen konfiguriert und verbunden sind. Die Konfiguration von Kommunikationsverbindungen wird im Abschnitt [Verbindungen zwischen Server und Client konfigurieren](#) beschrieben.
4. Überprüfen Sie, ob die Installation fehlerfrei arbeitet. Die Überprüfung der Installation wird im Abschnitt [Clientinstallation überprüfen](#) beschrieben.
5. Nach der Überprüfung der IBM WebSphere MQ MQI-Clientinstallation müssen Sie sich überlegen, ob Sicherheitsfunktionen für den Client konfiguriert werden sollen. Die Clientsicherheit wird im Abschnitt [IBM WebSphere MQ MQI-Clientsicherheit einrichten](#) beschrieben.
6. Richten Sie die Kanäle zwischen dem IBM WebSphere MQ MQI-Client und -Server ein, die für die IBM WebSphere MQ-Anwendungen erforderlich sind, die auf dem Client ausgeführt werden sollen. Die

Einrichtung von Kanälen wird im Abschnitt [MQI-Kanäle definieren](#) beschrieben. Bei Verwendung von SSL müssen zusätzliche Punkte berücksichtigt werden. Diese Aspekte werden unter [Angaben, dass ein MQI-Kanal SSL verwendet](#) beschrieben. Unter Umständen müssen Sie die Kanäle mithilfe einer IBM WebSphere MQ MQI-Clientkonfigurationsdatei oder mithilfe von IBM WebSphere MQ-Umgebungsvariablen einrichten. IBM WebSphere MQ-Umgebungsvariablen werden im Abschnitt [IBM WebSphere MQ-Umgebungsvariablen verwenden](#) beschrieben.

7. Eine ausführliche Beschreibung von IBM WebSphere MQ-Anwendungen finden Sie im Abschnitt [Anwendungen entwickeln](#).
8. Beim Konzipieren, Erstellen und Ausführen von Anwendungen in der IBM WebSphere MQ MQI-Clienumgebung müssen einige Unterschiede zur Warteschlangenumgebung berücksichtigt werden. Informationen zu diesen Unterschieden finden Sie in den folgenden Abschnitten:
  - [Schnittstelle für Nachrichtenwarteschlangen \(MQI\) in einer Clientanwendung verwenden](#)
  - [Anwendungen für WebSphere MQ-Clients erstellen](#)
  - [IBM WebSphere MQ-Clientanwendungen mit Warteschlangenmanagern verbinden](#)
  - [Probleme mit IBM WebSphere MQ-MQI-Clients beheben](#)

### ***Was ist ein erweiterter Transaktionsclient?***

Ein erweiterter transaktionsorientierter WebSphere MQ-Client kann unter der Steuerung eines externen Transaktionsmanagers Ressourcen aktualisieren, die von einem anderen Ressourcenmanager verwaltet werden.

Wenn Sie nicht mit den Grundlagen des Transaktionsmanagements vertraut sind, lesen Sie den Abschnitt [„Transaktionsmanagement und -unterstützung“](#) auf Seite 147.

Der transaktionsorientierte XA-Client wird nun mit WebSphere MQ bereitgestellt.

Eine Clientanwendung kann an einer Arbeitseinheit beteiligt sein, die von einem Warteschlangenmanager verwaltet wird, mit dem sie verbunden ist. Innerhalb der Arbeitseinheit kann die Clientanwendung Nachrichten in die Warteschlangen dieses Warteschlangenmanagers einreihen bzw. Nachrichten aus diesen Warteschlangen abrufen. Anschließend kann die Clientanwendung die Arbeitseinheit mit dem **MQCMIT**-Aufruf festschreiben oder mit dem **MQBACK**-Aufruf zurücksetzen. Die Clientanwendung kann allerdings nicht innerhalb einer Arbeitseinheit die Ressourcen (wie beispielsweise die Tabellen einer DB2-Datenbank) eines anderen Ressourcenmanagers aktualisieren. Bei Verwendung eines erweiterten transaktionsorientierten WebSphere MQ-Clients besteht diese Einschränkung nicht.

Ein erweiterter transaktionsorientierter *WebSphere MQ-Client* ist ein IBM WebSphere MQ MQI client mit einigen zusätzlichen Funktionen. Mithilfe dieser Funktionen kann eine Clientanwendung innerhalb derselben Arbeitseinheit Folgendes ausführen:

- Nachrichten in Warteschlangen einreihen, die dem Warteschlangenmanager zugeordnet sind, mit dem die Anwendung verbunden ist, bzw. Nachrichten aus diesem Warteschlangenmanager abrufen.
- Die Ressourcen eines Ressourcenmanagers, bei dem es sich nicht um einen WebSphere MQ-Warteschlangenmanager handelt, verwalten.

Diese Arbeitseinheit muss von einem externen Transaktionsmanager verwaltet werden, der auf demselben System wie die Clientanwendung aktiv ist. Die Arbeitseinheit darf nicht von dem Warteschlangenmanager verwaltet werden, mit dem die Clientanwendung verbunden ist. Das heißt, dass der Warteschlangenmanager nur als Ressourcenmanager, nicht als Transaktionsmanager dienen kann. Es heißt auch, dass die Clientanwendung die Arbeitseinheit nur über die Anwendungsprogrammierschnittstelle (API) des externen Transaktionsmanagers festschreiben bzw. zurücksetzen kann. Die Clientanwendung kann daher die MQI-Aufrufe **MQBEGIN**, **MQCMIT** und **MQBACK** nicht verwenden.

Der externe Transaktionsmanager kommuniziert mit dem Warteschlangenmanager als Ressourcenmanager über denselben MQI-Kanal, der auch von der Clientanwendung verwendet wird, die mit dem Warteschlangenmanager verbunden ist. Bei einer Wiederherstellung im Anschluss an einen Ausfall, wenn keine Anwendungen aktiv sind, kann der Transaktionsmanager jedoch über einen dedizierten MQI-Kanal alle unvollständigen Arbeitseinheiten wiederherstellen, an denen der Warteschlangenmanager zum Zeitpunkt des Ausfalls beteiligt war.



In diesem Abschnitt werden WebSphere MQ MQI-Clients ohne die erweiterte transaktionsorientierte Funktion als *WebSphere MQ-Basisclients* bezeichnet. Ein erweiterter transaktionsorientierter WebSphere MQ-Client ist also ein WebSphere MQ-Basisclient mit erweiterter transaktionsorientierter Funktion.

### Zugehörige Verweise

„Plattformunterstützung für erweiterte Transaktionsclients“ auf Seite 145

Erweiterte transaktionsorientierte IBM WebSphere MQ-Clients sind mit Ausnahme von z/OS für alle Plattformen verfügbar, die einen Basisclient unterstützen.

#### *Plattformunterstützung für erweiterte Transaktionsclients*

Erweiterte transaktionsorientierte IBM WebSphere MQ-Clients sind mit Ausnahme von z/OS für alle Plattformen verfügbar, die einen Basisclient unterstützen.

Clientanwendungen, die einen erweiterten transaktionsorientierten Client verwenden, können nur zu einem Warteschlangenmanager der folgenden IBM WebSphere MQ Version 7.5-Produkte eine Verbindung herstellen:

- IBM WebSphere MQ für AIX
- IBM WebSphere MQ für HP-UX
- IBM WebSphere MQ für HP Integrity NonStop Server
- IBM WebSphere MQ für Linux
- IBM WebSphere MQ für Solaris
- IBM WebSphere MQ für Windows

Zwar gibt es keine erweiterten transaktionsorientierten Clients, die unter z/OS ausgeführt werden, jedoch können Clientanwendungen, die einen erweiterten transaktionsorientierten Client verwenden, eine Verbindung zu einem Warteschlangenmanager herstellen, der unter z/OS läuft.

Auf allen Plattformen gelten für erweiterte transaktionsorientierte Clients dieselben Hardware- und Softwarevoraussetzungen wie für den IBM WebSphere MQ-Basisclient. Programmiersprachen, die vom IBM WebSphere MQ-Basisclient und dem von Ihnen verwendeten Transaktionsmanager unterstützt werden, werden auch von einem erweiterten transaktionsorientierten Client unterstützt.

Die externen Transaktionsmanager, die auf den einzelnen Plattformen unterstützt werden, sind auf den folgenden Webseiten aufgeführt:

<b>Plattform des erweiterten transaktionsorientierten Clients</b>	<b>Webseite</b>
AIX	<a href="#">Mindestvoraussetzungen für WebSphere MQ V7.5 - AIX</a>
HP-UX	<a href="#">Mindestvoraussetzungen für WebSphere MQ V7.5 - HP-UX</a>
HP Integrity NonStop Server	<a href="#">IBM WebSphere MQ -Clientumgebung unter HP Integrity NonStop Serverplanen</a>
Linux	<a href="#">Mindestvoraussetzungen für WebSphere MQ V7.5 - Linux</a>
Solaris	<a href="#">Mindestvoraussetzungen für WebSphere MQ V7.5 - Solaris</a>
Windows	<a href="#">Mindestvoraussetzungen für WebSphere MQ V7.5 - Windows</a>

### **Verbindung vom Client zum Server herstellen**

Clients stellen über MQCONN- oder MQCONNX-Aufrufe eine Verbindung zum Server her; die Kommunikation erfolgt über einen Kanal.

Eine in der IBM WebSphere MQ-Clientumgebung aktive Anwendung muss eine aktive Verbindung zwischen Client- und Servermaschinen aufrechterhalten.

Die Verbindung wird über einen MQCONN- oder MQCONNX-Aufruf hergestellt, den die Anwendung absetzt. Clients und Server kommunizieren über *MQI-Kanäle* miteinander oder (bei Verwendung gemeinsam genutzter Verbindungen) über Verbindungen, bei denen eine MQI-Kanalinstanz gemeinsam genutzt wird.

Ist der Aufruf erfolgreich, wird die MQI-Kanalinstanz oder Verbindung aufrechterhalten, bis die Anwendung einen MQDISC-Aufruf absetzt. Dies gilt für jeden Warteschlangenmanager, zu dem eine Anwendung eine Verbindung herstellen muss.

### Zugehörige Konzepte

„Client und Warteschlangenmanager auf einem System“ auf Seite 146

Anwendungen können auch in der WebSphere MQ MQI-Clientumgebung ausgeführt werden, wenn auf derselben Maschine zusätzlich noch ein Warteschlangenmanager installiert ist.

„Clients auf verschiedenen Plattformen“ auf Seite 146

Hier ein weiteres Beispiel eines WebSphere MQ MQI-Client/Server-Systems. In diesem Beispiel kommuniziert die Servermaschine mit drei WebSphere MQ MQI-Clients, die auf unterschiedlichen Plattformen installiert sind.

„Verschiedene Versionen von Client- und Server-Software verwenden“ auf Seite 146

Bei Verwendung älterer Versionen von IBM WebSphere MQ-Produkten müssen Sie sicherstellen, dass der Server die Konvertierung der CCSID (ID des codierten Zeichensatzes) des Clients unterstützt.

#### *Client und Warteschlangenmanager auf einem System*

Anwendungen können auch in der WebSphere MQ MQI-Clientumgebung ausgeführt werden, wenn auf derselben Maschine zusätzlich noch ein Warteschlangenmanager installiert ist.

In diesem Fall können Sie eine Verbindung zu den Warteschlangenmanagerbibliotheken oder den Clientbibliotheken herstellen; allerdings müssen auch bei einer Verbindung zu den Clientbibliotheken die Kanalverbindungen definiert werden. Dies kann bei der Entwicklung einer Anwendung hilfreich sein. Sie können das Programm auf der eigenen Maschine, ohne Abhängigkeiten von anderen, testen und sicher sein, dass es auch dann arbeitet, wenn es in einer unabhängigen WebSphere MQ MQI-Clientumgebung eingesetzt wird.

#### *Clients auf verschiedenen Plattformen*

Hier ein weiteres Beispiel eines WebSphere MQ MQI-Client/Server-Systems. In diesem Beispiel kommuniziert die Servermaschine mit drei WebSphere MQ MQI-Clients, die auf unterschiedlichen Plattformen installiert sind.

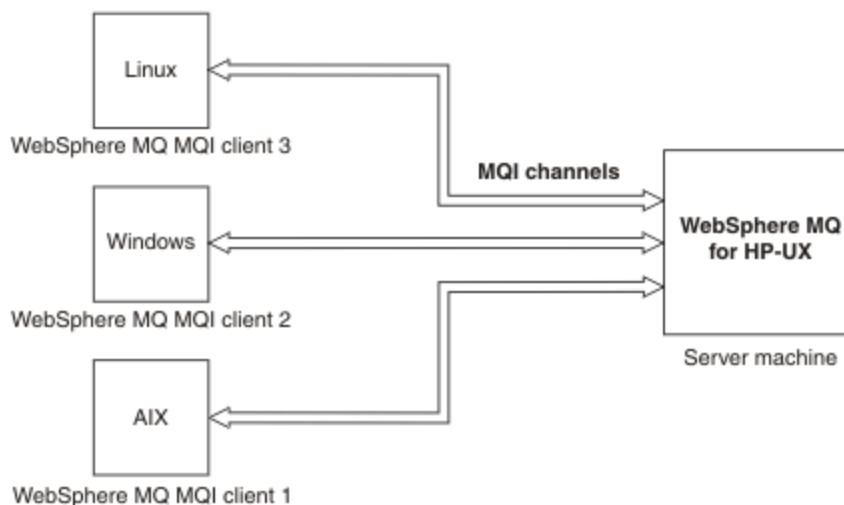


Abbildung 31. Verbindung eines WebSphere MQ-Servers mit Clients auf unterschiedlichen Plattformen

Es sind weitere komplexere Umgebungen möglich. So kann beispielsweise ein WebSphere MQ-Client mit mehreren Warteschlangenmanagern verbunden werden oder mit einer beliebigen Anzahl an Warteschlangenmanagern, die als Teil einer Gruppe mit gemeinsamem Warteschlangenmanager miteinander verbunden sind.

#### *Verschiedene Versionen von Client- und Server-Software verwenden*

Bei Verwendung älterer Versionen von IBM WebSphere MQ-Produkten müssen Sie sicherstellen, dass der Server die Konvertierung der CCSID (ID des codierten Zeichensatzes) des Clients unterstützt.

Ein IBM WebSphere MQ -Client kann eine Verbindung zu allen unterstützten Versionen des Warteschlangenmanagers herstellen. Wenn Sie eine Verbindung mit einem Warteschlangenmanager einer früheren Version herstellen, können Sie Funktionen und Strukturen aus einer späteren Version des Produkts in Ihrer IBM WebSphere MQ-Anwendung auf dem Client nicht verwenden.

Ein IBM WebSphere MQ -Warteschlangenmanager kann mit Clients unterschiedlicher Versionen mit sich selbst kommunizieren, indem er bis zur höchsten gegenseitig unterstützten Protokollebene aushandelt. Dies bedeutet, dass ältere Clients mit höheren Warteschlangenmanagerversionen verwendet werden können. Es wird empfohlen, dass sowohl der Client als auch der Server IBM WebSphere MQ -Versionen haben, die derzeit unterstützt werden, um die Problemdiagnose zu vereinfachen und die Unterstützung durch IBM zu aktivieren.

Weitere Informationen finden Sie in den unterstützten Programmiersprachen unter Entscheiden, welche Programmiersprache verwendet werden soll.

## Transaktionsmanagement und -unterstützung

In diesem Abschnitt wird eine Übersicht über das Transaktionsmanagement gegeben und erläutert, wie Transaktionen in WebSphere MQ unterstützt werden.

Ein *Ressourcenmanager* ist ein Computersubsystem, das Eigner der Ressourcen ist, die von ihm verwaltet werden; Anwendungen können auf diese Ressourcen zugreifen und sie aktualisieren. Hier einige Beispiele für Ressourcenmanager:

- Ein WebSphere MQ-Warteschlangenmanager; bei den Ressourcen handelt es sich um Warteschlangen
- Eine DB2-Datenbank; bei den Ressourcen handelt es sich um die Datenbanktabellen

Wenn eine Anwendung die Ressourcen auf einem oder auch mehreren Ressourcenmanagern aktualisiert, ist möglicherweise eine Geschäftsanforderung implementiert, mit der sichergestellt wird, dass bestimmte Aktualisierungen entweder zusammen erfolgreich oder gar nicht ausgeführt werden. Dies ist sinnvoll, da die Geschäftsdaten einen inkonsistenten Status hätten, wenn einige Aktualisierungen erfolgreich wären, andere hingegen nicht.

Bei dieser Art von Aktualisierung von Ressourcen, die auf diese Weise verwaltet werden, spricht man von einer *Arbeitseinheit* bzw. *Transaktion*. Ein Anwendungsprogramm kann eine Reihe von Aktualisierungen zu einer solchen Arbeitseinheit zusammenfassen.

Während einer solchen Arbeitseinheit sendet die Anwendung an die Ressourcenmanager Anforderungen zur Aktualisierung der Ressourcen. Die Arbeitseinheit endet, wenn von der Anwendung die Anforderung abgesetzt wird, alle Aktualisierungen festzuschreiben (Commit). Bis zu ihrer Festschreibung bleiben die Aktualisierungen für andere Anwendungen, die auf dieselben Ressourcen zugreifen, unsichtbar. Ebenso kann eine Anwendung die Arbeitseinheit auch abrechnen und eine Anforderung absetzen, alle bis dahin angeforderten Aktualisierungen zurückzusetzen. In diesem Fall werden die Aktualisierung nie sichtbar für andere Anwendungen. Diese Aktualisierungen sind in der Regel logisch miteinander verknüpft und müssen alle erfolgreich ausgeführt werden, damit die Datenintegrität gewährleistet ist. Die Datenintegrität geht verloren, wenn einige Aktualisierungen erfolgreich verlaufen, andere nicht.

Wenn eine Arbeitseinheit erfolgreich abgeschlossen ist, spricht man von einer *Festschreibung*. Nach der Festschreibung sind alle innerhalb einer Arbeitseinheit vorgenommenen Aktualisierungen permanent und können nicht mehr zurückgesetzt werden. Schlägt die Arbeitseinheit fehl, werden alle Aktualisierungen wieder *zurückgesetzt*. Dieser Prozess, bei dem Arbeitseinheiten unter Wahrung der Datenintegrität festgeschrieben oder zurückgesetzt werden, wird als *Synchronisationspunktkoordination* bezeichnet.

Der Zeitpunkt, zu dem alle Aktualisierungen innerhalb einer Arbeitseinheit festgeschrieben oder zurückgesetzt werden, wird als *Synchronisationspunkt* bezeichnet. Bei einer Aktualisierung innerhalb einer Arbeitseinheit spricht man davon, dass sie *synchronisationspunktgesteuert* ausgeführt wird. Fordert eine Anwendung eine *nicht synchronisationspunktgesteuerte* Aktualisierung an, schreibt der Ressourcenmanager die Aktualisierung sofort fest, auch wenn eine Arbeitseinheit noch nicht abgeschlossen wurde und die Aktualisierung nicht mehr zurückgesetzt werden kann.

Das Computersubsystem, das Arbeitseinheiten verwaltet, wird als *Transaktionsmanager* oder als *-Synchronisationspunktkoordinator* bezeichnet.

Bei einer *lokalen* Arbeitseinheit werden nur die Ressourcen des WebSphere MQ-Warteschlangenmanagers aktualisiert. Die Synchronisationspunktkoordination wird vom Warteschlangenmanager selbst mit einer einphasigen Festschreibung vorgenommen.

Bei einer *globalen* Arbeitseinheit werden auch die Ressourcen anderer Ressourcenmanager, beispielsweise einer XA-konformen Datenbank, aktualisiert. Hier wird eine zweiphasige Festschreibung durchgeführt; die Arbeitseinheit kann vom Warteschlangenmanager selbst oder von einem anderen XA-konformen Transaktionsmanager wie IBM TXSeries oder BEA Tuxedo koordiniert werden.

Der Transaktionsmanager muss dafür sorgen, dass entweder alle Aktualisierungen, die an Ressourcen vorgenommen werden, erfolgreich innerhalb einer Arbeitseinheit abgeschlossen werden, oder keine. Anwendungen senden Anforderungen zum Festschreiben oder Zurücksetzen einer Arbeitseinheit an den Transaktionsmanager. Beispiele für Transaktionsmanager sind CICS und WebSphere Application Server, auch wenn beide Produkte noch über andere Funktionen verfügen.

Einige Ressourcenmanager stellen eine eigene Funktion für das Transaktionsmanagement bereit. So kann beispielsweise ein WebSphere MQ-Warteschlangenmanager Arbeitseinheiten verwalten, bei denen die eigenen Ressourcen und DB2-Tabellen aktualisiert werden. Der Warteschlangenmanager benötigt hierfür keinen separaten Transaktionsmanager; wenn dies vom Benutzer gefordert wird, kann jedoch zusätzlich ein Transaktionsmanager eingesetzt werden. Ein solcher separater Transaktionsmanager wird als *externer Transaktionsmanager* bezeichnet.

Damit eine Arbeitseinheit von einem externen Transaktionsmanager verwaltet werden kann, ist zwischen dem Transaktionsmanager und allen an der Arbeitseinheit beteiligten Ressourcenmanagern eine Standardschnittstelle erforderlich. Über diese Schnittstellen können der Transaktionsmanager und die Ressourcenmanager miteinander kommunizieren. Eine dieser Schnittstellen ist die *XA-Schnittstelle*, eine Standardschnittstelle, die von einer Reihe von Transaktions- und Ressourcenmanagern unterstützt wird. Die XA-Schnittstelle ist von The Open Group in *Distributed Transaction Processing: The XA Specification* veröffentlicht.

Wenn mehrere Ressourcenmanager an einer Arbeitseinheit beteiligt sind, muss der Transaktionsmanager ein Protokoll für die *zweiphasige Festschreibung* verwenden; damit wird sichergestellt, dass entweder alle Aktualisierungen innerhalb einer Arbeitseinheit erfolgreich ausgeführt werden oder keine, und dies auch im Fall eines Systemausfalls. Wenn der Transaktionsmanager von einer Anwendung eine Anforderung zum Festschreiben einer Arbeitseinheit erhält, wird Folgendes durchgeführt:

#### **Phase 1 (Vorbereitung für die Festschreibung)**

Der Transaktionsmanager fragt die einzelnen Ressourcenmanager ab, die an dieser Arbeitseinheit beteiligt sind, um sicherzustellen, dass alle Informationen zu den beabsichtigten Aktualisierungen an den Ressourcen einen wiederherstellbaren Status haben. Dazu schreibt ein Ressourcenmanager die Informationen in der Regel in ein Protokoll und stellt sicher, dass diese Informationen auf der Festplatte gespeichert werden. Phase 1 ist abgeschlossen, wenn der Transaktionsmanager von jedem Ressourcenmanager benachrichtigt wurde, dass die Informationen zu den beabsichtigten Aktualisierungen an den Ressourcen einen wiederherstellbaren Status haben.

#### **Phase 2 (Festschreibung)**

Nach Abschluss der Phase 1 bereitet der Transaktionsmanager den unwiderruflichen Schritt vor, mit dem die Arbeitseinheit festgeschrieben wird. Dazu fordert er die einzelnen Ressourcenmanager, die an der Arbeitseinheit beteiligt sind, auf, die Aktualisierungen an ihren Ressourcen festzuschreiben. Wenn ein Ressourcenmanager diese Anforderung empfängt, muss er die Aktualisierungen festschreiben. Zu diesem Zeitpunkt ist es für ihn nicht mehr möglich, die Aktualisierungen zurückzusetzen. Phase 2 ist abgeschlossen, wenn der Transaktionsmanager von jedem Ressourcenmanager benachrichtigt wird, dass die Aktualisierungen an den Ressourcen festgeschrieben wurden.

Die XA-Schnittstelle verwendet ein solches Protokoll für die zweiphasige Festschreibung.

Weitere Informationen finden Sie im Abschnitt [Transaktionsunterstützung](#).

WebSphere MQ unterstützt auch Microsoft Transaction Server (COM +). Unter [Microsoft Transaction Server \(COM +\)](#) verwenden finden Sie Informationen zum Einrichten von WebSphere MQ für die Nutzung der COM + -Unterstützung.

## Funktionen des Warteschlangenmanagers erweitern

Sie können die Funktionen eines Warteschlangenmanagers durch Benutzerexits, API-Exits oder installierbare Services erweitern.

### Benutzerexits

Benutzerexits ermöglichen es Ihnen, eigenen Code in eine Warteschlangenmanagerfunktion einzufügen. Zu den unterstützten Benutzerexits gehören folgende:

#### Kanalexits

Über diese Exits kann die Betriebsweise von Kanälen geändert werden. Kanalexits werden im Abschnitt [Kanalexitprogramme für Messaging-Kanäle](#) beschrieben.

#### Datenkonvertierungsexits

Diese Exits erstellen Quellcodefragmente, die in Anwendungsprogramme eingefügt werden, um Daten in ein anderes Format zu konvertieren. Datenkonvertierungsexits werden im Abschnitt [Datenkonvertierungsexits schreiben](#) beschrieben.

#### Exit für Clusterauslastung

Die Funktion dieses Exits wird vom Exit-Provider definiert. Informationen zur Aufrufdefinition finden Sie im Abschnitt [MQ\\_CLUSTER\\_WORKLOAD\\_EXIT - Beschreibung des Aufrufs](#).

### API-Exits

Mit API-Exits können Sie Code schreiben, mit dem das Verhalten von WebSphere MQ-API-Aufrufen wie MQPUT und MQGET geändert werden kann, und diesen Code dann unmittelbar vor oder hinter diesen Aufrufen einfügen. Das Einfügen erfolgt automatisch; der Exit-Code wird dann vom Warteschlangenmanager an den registrierten Punkten ausgeführt. Weitere Informationen zu API-Exits finden Sie im Abschnitt [API-Exits schreiben und verwenden](#).

### Installierbare Services

Installierbare Services haben formalisierte Schnittstellen (eine API) mit mehreren Eingangspunkten.

Ein implementierter installierbarer Service wird als *Servicekomponente* bezeichnet. Sie können entweder die zusammen mit WebSphere MQ bereitgestellten Komponenten verwenden oder eigene Komponenten für die für Ihre Umgebung erforderlichen Funktionen erstellen.

Derzeit werden die folgenden installierbaren Services bereitgestellt:

#### Berechtigungsservice

Mithilfe des Berechtigungsservice können Sie eine eigene Sicherheitsfunktion erstellen.

Die Standardservicekomponente, mit der der Service implementiert wird, ist der Objektberechtigungsmanager (Object Authority Manager; OAM). Der Objektberechtigungsmanager ist standardmäßig aktiviert; Sie müssen nichts tun, um ihn zu konfigurieren. Mithilfe der Berechtigungsserviceschnittstelle können Sie andere Komponenten erstellen, um den Objektberechtigungsmanager zu ersetzen oder aber zu erweitern. Weitere Informationen zum Objektberechtigungsmanager finden Sie im Abschnitt [Sicherheit auf Windows- und UNIX and Linux-Systemen einrichten](#).

#### Namensservice

Mit dem Namensservice können Anwendungen Warteschlangen gemeinsam nutzen, indem ferne Warteschlangen wie lokale Warteschlangen angegeben werden.

Sie können eine eigene Namensservicekomponente erstellen. Dies ist beispielsweise sinnvoll, wenn der Namensservice zusammen mit IBM WebSphere MQ verwendet werden soll. Um den Namensservice zu verwenden, muss entweder eine benutzerdefinierte Komponente oder eine Komponente von einem anderen Softwareanbieter vorhanden sein. Der Namensservice ist standardmäßig inaktiv.

# IBM WebSphere MQ-Client für HP Integrity NonStop Server - Technische Übersicht

Technische Übersicht über das HP Integrity NonStop Server-Betriebssystem.

## SupportPac mit IBM WebSphere MQ-Client für HP Integrity NonStop Server

Der IBM WebSphere MQ-Client für HP Integrity NonStop Server wird in SupportPac MAT1 freigegeben.

## Technische Übersicht über das Betriebssystem HP Integrity NonStop Server

HP Integrity NonStop Server ist ein Betriebssystem, das für die höchstmögliche Verfügbarkeit entworfen wurde, ohne geplante oder ungeplante Ausfallzeiten selbst bei mehreren Hardware- oder Softwarefehlern. Es ist linear skalierbar; das heißt zum Beispiel, dass bei einer Erweiterung der Hardware um 20 Prozent auch 20 Prozent mehr nutzbare Leistung verfügbar ist. Um die Datenintegrität aufrechtzuerhalten, verfügt das Betriebssystem über einen eigenen Transaktionsmanager und ein transaktionsorientiertes Dateisystem.

Das Betriebssystem HP Integrity NonStop Server kommt häufig in folgenden Bereichen zum Einsatz:

- Finanzinstitute, z. B. für Vernetzung von Geldausgabeautomaten, Unterstützung des Onlinebankings, Kreditprüfungen, Börsengeschäfte, Trading und Finanztransaktionen zwischen Banken
- Produktion, z. B. für Back-End-Systeme von Webstores, Bestandsverwaltung und Prozesssteuerung
- Telekommunikation, z. B. für Vermittlungs-, Notfall- und sonstige Netzservices

## IBM WebSphere MQ-Client für HP Integrity NonStop Server-unterstützte Umgebungen und Funktionen

Stellt detaillierte Informationen zum IBM WebSphere MQ-Client für die HP Integrity NonStop Server-Plattform bereit und beschreibt die unterstützte Client-API und die unterstützten Umgebungen sowie die spezifische Clientfunktionalität von HP Integrity NonStop Server-Systemen.

### Unterstützte Client-API und Umgebungen

IBM WebSphere MQ-Client für HP Integrity NonStop Server unterstützt die folgenden Ausführungsumgebungen:

*Tabelle 10.*

	OSS	Wächter
C	✓	✓
JMS	✓	
COBOL	✓	✓
pTAL	✓	✓

### Funktionsübersicht

Einige Aspekte der Clientfunktionalität sind hostbetriebssystemspezifisch. Die folgende Übersicht beschreibt die Aspekte der Client-Funktionalität, die speziell für IBM WebSphere MQ-Client für HP Integrity NonStop Server gelten:

- C (nativ), pTAL, COBOL (nativ)
  - Netzwerkprotokoll: TCP (IPv4 und IPv6)
  - Transporttyp: nur Client

- Transportsicherheit: SSL/TLS
- Transaktionsunterstützung: zweiphasige Festschreibung, koordiniert durch die Transaktionsverwaltungsfunktion (TMF) (Verbindung zu einem Warteschlangenmanager auf IBM WebSphere MQ Version 7.1 oder höher erforderlich)
- Adressmodus: 32 Bit
- Java Message Service (JMS)
  - Netzprotokoll: TCP (IPv4 und IPv6)
  - Transporttyp: Nur Client (Bindungen, direkte Übertragungen oder direkte HTTP-Verbindungen werden nicht unterstützt)
  - Transportsicherheit: SSL/TLS
  - Transaktionsunterstützung: Einphasige Festschreibung
  - Ausführung: Standalone (Application Support Facility (ASF) und Java ConnectorArchitecture (JCA) werden nicht unterstützt)
  - Exits: Nur Java (native, in anderen Sprachen geschriebene Exits werden nicht unterstützt)
  - IBM WebSphere MQ Header und PCF: Die folgenden Klassen werden nicht unterstützt: com.ibm.mq.headers.\* und com.ibm.mq.pcf.\*

## Szenarien

---

Dieser Abschnitt enthält Informationen zu Szenarios, die veranschaulichen, wie die neuen Funktionen von WebSphere MQ Version 7.5 verwendet und kombiniert werden können. Die Szenarios enthalten hilfreiche Links zu Infocenter-Inhalten, die Ihnen ein besseres Verständnis zu dem Themenbereich, für den Sie sich interessieren, vermitteln.

Eine Beschreibung der verfügbaren Szenarios finden Sie in den folgenden Abschnitten:

### Einführung in IBM WebSphere MQ Version 7.5

Dieses Szenario veranschaulicht die ersten Schritte in IBM WebSphere MQ Version 7.5 auf einer Windows-Plattform. Verwenden Sie dieses Szenario, wenn IBM WebSphere MQ für Sie neu ist und Sie möglichst schnell damit arbeiten möchten.

Das Szenario enthält die folgenden Abschnitte. Alle diese Schritte können entweder über die grafische Benutzerschnittstelle oder die Befehlszeilenschnittstelle ausgeführt werden, so wie im Szenario gezeigt.

### Basiskonzepte und wichtige Begriffe

In diesem Abschnitt werden die Basiskonzepte und Schlüsselbegriffe beschrieben, die Sie kennen sollten, bevor Sie das Szenario 'Erste Schritte in IBM WebSphere MQ Version 7.5' verwenden.

### Grundlegende Konzepte

IBM WebSphere MQ ermöglicht es Anwendungen, Nachrichten aus einer Warteschlange zu lesen und in eine Warteschlange zu schreiben. Die Anwendung, die die Nachricht liest, ist unabhängig von der Anwendung, die die Nachricht schreibt. Es wird auch nicht vorausgesetzt, dass die beiden Anwendungen gleichzeitig aktiv sind. Wenn keine Anwendung zum Lesen der Nachricht verfügbar ist, wird die Nachricht in die IBM WebSphere MQ-Warteschlange eingereiht, bis sie von einer Anwendung gelesen wird.

### Schlüsselbegriffe

Es folgt eine Liste mit Schlüsselbegriffen zur Steuerung von Nachrichtenwarteschlangen (Message-Queuing).

Begriff	Beschreibung
Warteschlangenmanager	Der Warteschlangenmanager ist zuständig für die Verwaltung aller ihm zugeordneten Warteschlangen; er muss außerdem alle Nachrichten, die er empfängt, in den entsprechenden Warteschlangen speichern.
Nachrichten	Eine Nachricht ist eine Bytezeichenfolge, die Informationen für die Anwendungen bereitstellt, die diese Nachricht verwenden. Mithilfe von Nachrichten werden Informationen von einem Anwendungsprogramm an ein anderes übertragen. Die Anwendungen können auf demselben oder auf verschiedenen Computern aktiv sein.
Lokale Warteschlangen	Eine lokale Warteschlange ist eine Datenstruktur, in der Nachrichten gespeichert werden. Dabei kann es sich um eine normale Warteschlange oder eine Übertragungswarteschlange handeln. Eine normale Warteschlange enthält Nachrichten, die von einer Anwendung gelesen werden sollen, die die Nachricht direkt vom Warteschlangenmanager liest. Eine Übertragungswarteschlange enthält Nachrichten, die an einen anderen Warteschlangenmanager übertragen werden sollen.
Ferne Warteschlangen	Über eine ferne Warteschlange wird eine Nachricht an einen anderen Warteschlangenmanager adressiert.
Kanäle	Über Kanäle werden Nachrichten zwischen Warteschlangenmanagern gesendet und empfangen.
Empfangsprogramme	Empfangsprogramme sind Prozesse, die Netzanforderungen von anderen Warteschlangenmanagern oder Clientanwendungen annehmen und die zugeordneten Kanäle starten.

## Erstellen eines Warteschlangenmanagers mit dem Namen QM1

Erstellen Sie einen Warteschlangenmanager mit dem Namen QM1 für das Szenario 'Erste Schritte mit IBM WebSphere MQ Version 7.5', indem Sie entweder die Befehlszeilenschnittstelle oder den WebSphere MQ Explorer verwenden. Warteschlangenmanager sind die Hauptkomponenten in einem WebSphere MQ-Nachrichtenübermittlungsnetz.

### Vorbereitende Schritte

IBM WebSphere MQ Version 7.5 muss bereits installiert sein. Wenn nicht, finden Sie im Abschnitt [Installation und Deinstallation](#) Informationen zur Installation.

### Informationen zu diesem Vorgang

In diesem Beispiel sind alle Namen in Großbuchstaben geschrieben. Da IBM WebSphere MQ die Groß-/Kleinschreibung beachtet, müssen Sie die Namen ebenfalls groß eingeben.

### **Warteschlangenmanager über die Befehlszeilenschnittstelle erstellen**

Gehen Sie wie folgt vor, um einen Warteschlangenmanager über die Befehlszeilenschnittstelle zu erstellen und zu starten:

### Vorgehensweise

1. Erstellen Sie einen Warteschlangenmanager mit dem Namen QM1, indem Sie folgenden Befehl eingeben:

```
crtmqm QM1
```



Während das System den Warteschlangenmanager erstellt, wird folgende Ausgabe angezeigt:

```
C:\>crtmqm QM1
WebSphere MQ queue manager created.
Creating or replacing default objects for QM1.
Default objects statistics : 61 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

Der Warteschlangenmanager wird erstellt und ist gestoppt. Sie müssen den Warteschlangenmanager starten, um ihn verwalten und Nachrichten aus seinen Warteschlangen lesen und in seine Warteschlangen schreiben zu können.

2. Starten Sie den Warteschlangenmanager, indem Sie den folgenden Befehl eingeben:

```
strmqm QM1
```

Bei einem erfolgreichen Start des Warteschlangenmanagers wird folgende Ausgabe angezeigt:

```
C:\>strmqm QM1
WebSphere MQ queue manager 'QM1' starting.
5 log records accessed on queue manager 'QM1' during the log replay phase.
Log replay for queue manager 'QM1' complete.
Transaction manager state recovered for queue manager 'QM1'.
WebSphere MQ queue manager 'QM1' started.
```

Der Warteschlangenmanager ist gestartet.

## Nächste Schritte

Informationen zum Erstellen einer Warteschlange finden Sie im Abschnitt [„Warteschlange LQ1 erstellen“](#) auf Seite 154.

### ***Warteschlangenmanager mit WebSphere MQ Explorer erstellen***

Gehen Sie wie folgt vor, um einen Warteschlangenmanager mit WebSphere MQ Explorer zu erstellen und zu starten:

#### **Vorgehensweise**

1. Starten Sie WebSphere MQ Explorer.
2. Klicken Sie in der Navigatoransicht **Navigator** mit der rechten Maustaste auf den Ordner **Warteschlangenmanager** und klicken Sie dann auf **Neu > Warteschlangenmanager**. Der Assistent **Warteschlangenmanager erstellen** wird gestartet.
3. Geben Sie im Feld **Warteschlangenmanager-Name** den Namen QM1 ein.
4. Wählen Sie das Kontrollkästchen **Make this the default queue manager** aus.
5. Geben Sie im Feld **Warteschlange für nicht zustellbare Nachrichten** den Namen SYS-TEM.DEAD.LETTER.QUEUE ein.  
Dies ist der Name der Warteschlange für nicht zustellbare Nachrichten, die beim Erstellen des Warteschlangenmanagers automatisch erstellt wird.
6. Lassen Sie die übrigen Felder leer und klicken Sie auf **Fertigstellen** oder, falls diese Schaltfläche inaktiviert ist, auf **Weiter**.  
Die Schaltfläche **Fertigstellen** ist inaktiviert, wenn es bei der Portnummer einen Konflikt mit einem vorhandenen Warteschlangenmanager gibt, z. B. mit dem Warteschlangenmanager, der im Rahmen der Standardkonfiguration erstellt wird. Sie müssen mit dem Assistenten fortfahren, um die Standardportnummer zu ändern.
7. Wenn Sie auf **Weiter** geklickt haben, akzeptieren Sie die Standardwerte und klicken Sie auf jeder Seite auf **Weiter**, bis Sie zur letzten Seite des Assistenten gelangen, auf der die Schaltfläche **Fertigstellen** dann verfügbar ist. Ändern Sie die angegebene Portnummer, z. B. in 1415, und klicken Sie auf **Fertigstellen**.

WebSphere MQ zeigt das Dialogfenster **Warteschlangenmanager wird erstellt** an, während der Warteschlangenmanager erstellt und gestartet wird.

## Nächste Schritte

Informationen zum Erstellen einer Warteschlange finden Sie im Abschnitt „[Warteschlange LQ1 erstellen](#)“ auf Seite 154.

## Warteschlange LQ1 erstellen

Erstellen Sie entweder über die Befehlszeilenschnittstelle oder mit dem WebSphere MQ Explorer eine Warteschlange, die im Szenario 'Erste Schritte in IBM WebSphere MQ Version 7.5' verwendet werden soll. Warteschlangen sind Strukturen, die zum Speichern von Nachrichten verwendet werden und bei denen es sich um Warteschlangenmanager-Objekte von IBM WebSphere MQ handelt.

## Informationen zu diesem Vorgang

Es gibt drei Möglichkeiten, IBM WebSphere MQ-Objekte zu erstellen:

- über die Befehlszeile
- IBM WebSphere MQ Explorer.
- über eine programmierbare Schnittstelle

In dieser Task können IBM WebSphere MQ -Objekte über die Befehlszeile oder die IBM WebSphere MQ Explorer erstellt werden.

## Warteschlange über die Befehlszeilenschnittstelle erstellen

Für die Befehlszeilenschnittstelle gibt es die Scriptsprache IBM WebSphere MQ Script Commands (MQSC). Zur Ausführung des Scripts für einen Warteschlangenmanager wird das Scripting-Tool **runmqsc** verwendet. Gehen Sie wie folgt vor, um eine Warteschlange über die Befehlszeilenschnittstelle zu erstellen und zu starten:

## Vorgehensweise

1. Starten Sie das Scripting-Tool mit folgendem Befehl:

```
runmqsc QM1
```

Beim Start des Scripting-Tools wird folgende Ausgabe angezeigt:

```
C:\>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
```

Das Tool ist nun für WebSphere MQ-Scriptbefehle bereit.

2. Erstellen Sie eine lokale Warteschlange namens LQ1, indem Sie den folgenden MQSC-Befehl eingeben:

```
define qlocal(LQ1)
```

Während die Warteschlange erstellt wird, wird folgende Ausgabe angezeigt:

```
define qlocal(LQ1)
  2 : define qlocal(LQ1)
AMQ8006: WebSphere MQ queue created.
```

3. Mit folgendem WebSphere MQ-Scriptbefehl stoppen Sie das Scripting-Tool wieder:

```
end
```

Bei Beendigung des Scripting-Tools wird folgende Ausgabe angezeigt:

```
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.  
  
C:\>
```

## Nächste Schritte

Sie können jetzt eine Nachricht in die Warteschlange einreihen. Informationen zum Einreihen einer Nachricht in eine Warteschlange finden Sie im Abschnitt [„Nachricht in die Warteschlange LQ1 einreihen“](#) auf Seite 155.

### **Warteschlange mit WebSphere MQ Explorer erstellen**

Gehen Sie wie folgt vor, um eine Warteschlange mit WebSphere MQ Explorer zu erstellen und zu starten:

#### Vorgehensweise

1. Erweitern Sie in der Ansicht **Navigator** den Ordner **Warteschlangenmanager**.
2. Erweitern Sie den Warteschlangenmanager **QM1**.
3. Klicken Sie mit der rechten Maustaste auf den Ordner **Warteschlangen** und klicken Sie anschließend auf **Neu > Lokale Warteschlange...** Der Assistent **New Local Queue** (Neue lokale Warteschlange) wird gestartet.
4. Geben Sie im Feld **Name** den Wert LQ1ein.
5. Klicken Sie auf **Fertigstellen**.

Die neue Warteschlange LQ1 wird in der Ansicht **Inhalt** angezeigt. Falls die Warteschlange nicht in der Ansicht **Inhalt** angezeigt wird, klicken Sie auf die Schaltfläche **Aktualisieren** am oberen Rand der Ansicht **Inhalt**.

## Nächste Schritte

Sie können jetzt eine Nachricht in die Warteschlange einreihen. Informationen zum Einreihen einer Nachricht in eine Warteschlange finden Sie im Abschnitt [„Nachricht in die Warteschlange LQ1 einreihen“](#) auf Seite 155.

### **Nachricht in die Warteschlange LQ1 einreihen**

Reihen Sie eine Nachricht in die Warteschlange LQ1ein, um sie mit dem Szenario 'Einführung in IBM WebSphere MQ Version 7.5 ' zu verwenden, indem Sie entweder die Befehlszeilenschnittstelle oder IBM WebSphere MQ Explorerverwenden.

#### Informationen zu diesem Vorgang

IBM WebSphere MQ stellt standardmäßig die Beispielanwendung amqsput bereit. Diese Anwendung reiht eine Nachricht in eine vordefinierte Warteschlange ein.

### **Nachricht über die Befehlszeilenschnittstelle in die Warteschlange einreihen**

Gehen Sie wie folgt vor, um über die Befehlszeilenschnittstelle eine Nachricht in die Warteschlange einzureihen:

#### Vorgehensweise

1. Reihen Sie mithilfe der Beispielanwendung amqsput eine Nachricht in die Warteschlange LQ1 ein, indem Sie folgenden Befehl eingeben:

```
amqsput LQ1 QM1
```

Beim Start der Beispielanwendung wird folgende Ausgabe angezeigt:

```
C:\>amqspuT LQ1 QM1
Sample AMQSPuT0 start
target queue is LQ1
```

2. Geben Sie Hello World ein und drücken Sie die Eingabetaste. Sie haben eine Nachricht mit dem Text "Hello World" in die Warteschlange LQ1, die vom Warteschlangenmanager QM1 verwaltet wird, eingereiht.
3. Drücken Sie die Eingabetaste, um amqspuT zu beenden. Die folgende Ausgabe wird angezeigt:

```
C:\>amqspuT LQ1 QM1
Sample AMQSPuT0 start
target queue is LQ1
Hello World

Sample AMQSPuT0 end
```

## Nächste Schritte

Informationen zum Abrufen einer Nachricht aus einer Warteschlange finden Sie im Abschnitt „[Nachricht aus der Warteschlange LQ1 abrufen](#)“ auf Seite 156.

### ***Nachricht mit IBM WebSphere MQ Explorer in die Warteschlange einreihen***

Gehen Sie wie folgt vor, um mit IBM WebSphere MQ Explorer eine Nachricht in die Warteschlange einzureihen:

## Vorgehensweise

1. Erweitern Sie in der Ansicht **Navigator** den Ordner **Warteschlangenmanager**.
2. Erweitern Sie den von Ihnen erstellten Warteschlangenmanager QM1.
3. Klicken Sie auf den Ordner **Warteschlangen**. Die Warteschlangen des Warteschlangenmanagers werden in der Inhaltsansicht aufgelistet.
4. Klicken Sie in der Inhaltsansicht mit der rechten Maustaste auf die lokale Warteschlange LQ1 und klicken Sie dann auf **Testnachricht einreihen**.  
Der Dialog **Testnachricht einreihen** wird geöffnet.
5. Geben Sie im Feld **Nachrichtendaten** einen Text ein, z. B. Hello World, und klicken Sie dann auf **Nachricht einreihen**.  
Der Inhalt des Felds **Nachrichtendaten** wird gelöscht und die Nachricht in die Warteschlange eingereiht.
6. Klicken Sie auf **Schließen**.  
Beachten Sie in der Inhaltsansicht, dass der LQ1 **Aktuelle Warteschlangenlänge** -Wert jetzt 1 ist. Wenn die Spalte **Aktuelle Warteschlangenlänge** nicht sichtbar ist, müssen Sie möglicherweise nach rechts von **Inhaltsansichtblättern**.

## Nächste Schritte

Informationen zum Abrufen einer Nachricht aus einer Warteschlange finden Sie im Abschnitt „[Nachricht aus der Warteschlange LQ1 abrufen](#)“ auf Seite 156.

## Nachricht aus der Warteschlange LQ1 abrufen

Rufen Sie über die Befehlszeilenschnittstelle oder IBM WebSphere MQ Explorer eine Nachricht aus der Warteschlange LQ1 zur Verwendung mit dem Szenario 'Erste Schritte mit IBM WebSphere MQ Version 7.5' ab.

## Informationen zu diesem Vorgang

IBM WebSphere MQ stellt standardmäßig die Beispielanwendung amqsget bereit. Diese Anwendung liest Nachrichten aus einer Warteschlange.

## ***Nachricht über die Befehlszeilenschnittstelle aus der Warteschlange abrufen***

Gehen Sie wie folgt vor, um über die Befehlszeilenschnittstelle eine Nachricht aus der Warteschlange abzurufen:

### **Vorgehensweise**

Lesen Sie mithilfe der Beispielanwendung amqsget eine Nachricht aus der Warteschlange LQ1, indem Sie folgenden Befehl eingeben:

```
amqsget LQ1 QM1
```

Beim Start der Beispielanwendung wird folgende Ausgabe angezeigt:

```
C:\>amqsget LQ1 QM1
Sample AMQSGETO start
message <Hello World>
no more messages
Sample AMQSGETO end
```

Die Anwendung amqsget wird 30 Sekunden nach dem Lesen der Nachricht beendet.

### **Nächste Schritte**

Informationen zum Schreiben von Anwendungen zur Warteschlangensteuerung, zum Herstellen und Trennen einer Verbindung mit einem Warteschlangenmanager, zum Publish/Subscribe sowie zum Öffnen und Schließen von Objekten finden Sie im Abschnitt [Anwendung zur Warteschlangensteuerung schreiben](#).

## ***Nachricht mit IBM WebSphere MQ Explorer aus der Warteschlange abrufen***

Gehen Sie wie folgt vor, um mit IBM WebSphere MQ Explorer eine Nachricht aus der Warteschlange abzurufen:

### **Vorgehensweise**

1. Erweitern Sie in der Ansicht **Navigator** den Ordner **Warteschlangenmanager** und dann den Warteschlangenmanager QM1.
2. Klicken Sie auf den Ordner **Warteschlangen**.
3. Klicken Sie in der Ansicht **Inhalt** mit der rechten Maustaste auf QM1 und klicken Sie dann auf **Nachrichten anzeigen....** Im **Nachrichten-Browser** wird eine Liste mit den Nachrichten angezeigt, die aktuell von QM1 verwaltet werden.
4. Doppelklicken Sie auf die letzte Nachricht, um den Eigenschaftendialog zu öffnen.

Im Feld **Nachrichtendaten** auf der Seite **Daten** des Eigenschaftendialogs wird der Inhalt der Nachricht in lesbarer Form angezeigt.

### **Nächste Schritte**

Informationen zum Schreiben von Anwendungen zur Warteschlangensteuerung, zum Herstellen und Trennen einer Verbindung mit einem Warteschlangenmanager, zum Publish/Subscribe sowie zum Öffnen und Schließen von Objekten finden Sie im Abschnitt [Anwendung zur Warteschlangensteuerung schreiben](#).

### **Nächste Schritte**

In diesem Abschnitt werden die weiteren Schritte zur Beendigung des Szenarios 'Erste Schritte in IBM WebSphere MQ Version 7.5' beschrieben.

IBM WebSphere MQ stellt rollenabhängige Schulungspfade bereit, die Sie unterstützen, indem sie einen Pfad zum Erwerb von Kenntnissen für bestimmte WebSphere-Produktangebote definieren. Es gibt zwei Schulungspfade für IBM WebSphere MQ:

- Anwendungsentwickler

Diese Benutzer sind für die Erstellung der Anwendungen, die den Warteschlangenmanager verwenden, verantwortlich. In diesem Szenario schreiben sie die Anwendungen amqsput und amqsget.

- Systemadministrator

Diese Benutzer sind für die Erstellung des Warteschlangenmanagers und seiner Objekte verantwortlich. Dabei führen sie ähnliche Aufgaben aus wie Sie in diesem Szenario.

Weitere Informationen zu IBM WebSphere MQ-Schulungspfaden finden Sie unter <https://www.ibm.com/software/websphere/education/paths/>.

Die vollständige Liste der IBM WebSphere MQ-Kurse finden Sie unter <https://www.ibm.com/software/websphere/education/curriculum/appint/wmq/>.

Es ist ein Zertifizierungsprogramm verfügbar, das nachweist, dass Sie eine bestimmte Kenntnisstufe in IBM WebSphere MQ erreicht haben. Weitere Informationen finden Sie unter <https://www.ibm.com/certify/index>.

Sie können, wie die folgenden Beispiele zeigen, mit anderen Benutzern zusammenarbeiten:

- E-Mail-basierte Community professioneller IBM WebSphere MQ-Anwender (siehe <https://listserv.meduniwien.ac.at/archives/mqser-l.html>).
- Diskussionsforen um die IBM WebSphere MQ-Produktreihe (siehe <https://www.mqseries.net/>).
- Ein IBM Developer-Blog von den Entwicklern der verschiedenen IBM Messaging-Produkte (siehe <https://www.ibm.com/developerworks/blogs/page/messaging/>).
- Das offizielle, von IBM bereitgestellte Forum für IBM WebSphere MQ (siehe <https://www.ibm.com/developerworks/forums/forum.jspa?forumid=280>).
- Fragen und Antworten zu IBM WebSphere MQ auf stackoverflow.com (siehe <https://stackoverflow.com/questions/tagged/websphere-mq>).

Diese Produktdokumentation enthält weitere Abschnitte, die Sie anzeigen können. Möglicherweise interessieren Sie sich für folgende Abschnitte:

- [Verwaltung IBM WebSphere MQ](#)

IBM WebSphere MQ stellt Steuerbefehle bereit, die Sie verwenden können. Zwei dieser Befehle werden in diesem Szenario verwendet: **crtmqm** und **strmqm**. Dieser Abschnitt gibt auch eine gute Übersicht über das Message-Queuing (Steuerung von Nachrichtenwarteschlangen).

- [WebSphere MQ-Scriptbefehlsreferenz \(MQSC\)](#)

In diesem Szenario definieren Sie mit dem Befehl `define qlocal('LQ1')` eine lokale Warteschlange mit dem Namen LQ1; bei dem Befehl handelt es sich um einen WebSphere MQ-Scriptbefehl. IBM WebSphere MQ-Systemadministratoren verwalten mit diesen Befehlen ihre Warteschlangenmanager. Dieser Abschnitt gibt eine Einführung zu den Befehlen und zeigt ihre Verwendung, bevor die Befehle ausführlich in alphabetischer Reihenfolge beschrieben werden.

- [Warteschlangenmanager-Cluster konfigurieren](#)

Dieser Abschnitt beschreibt, wie Warteschlangenmanager in virtuellen Gruppen, sogenannten Clustern, organisiert, verwendet und verwaltet werden. Das Clustering stellt sicher, dass jeder Warteschlangenmanager innerhalb eines Clusters alle anderen Warteschlangenmanager in seinem Cluster kennt. Es vereinfacht außerdem die Verwaltung von komplexen Warteschlangenmanagernetzen.

Die Produktdokumentation 'Product Connectivity Scenarios' enthält Informationen zu den wichtigsten Aufgaben für die Verbindung von WebSphere Application Server zu WebSphere MQ in verschiedenen Szenarios. Jedes Szenario enthält die Anweisungen für die Implementierung einer Lösung innerhalb eines Geschäftskontexts. Auf diese Weise lernen Sie während der Durcharbeitung der Szenarios Schritt für Schritt dazu, ohne auf andere Informationsressourcen zurückgreifen zu müssen. [https://www.ibm.com/docs/prodconn\\_1.0.0/com.ibm.prodconn.doc/infocenter\\_homepage/ic\\_home.htm](https://www.ibm.com/docs/prodconn_1.0.0/com.ibm.prodconn.doc/infocenter_homepage/ic_home.htm)

## Basisdateiübertragung mithilfe der Scripts

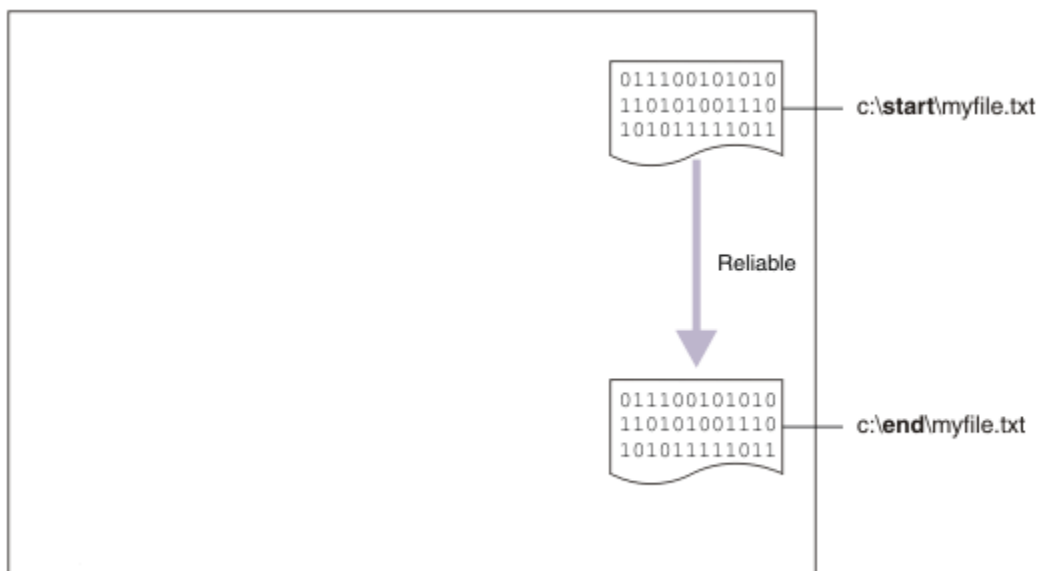
Sie können Dateien mit IBM WebSphere MQ Version 7.5 auf verschiedene Arten übertragen. In den folgenden Abschnitten werden Inhalt und Einsatzmöglichkeiten dieses Szenarios, mögliche Gründe für ein Unternehmen, die Verwendung des Szenarios zu erwägen, und die involvierten Benutzerrollen beschrieben. Außerdem erhalten Sie eine Übersicht über die mit dem Szenario vorgeschlagene Lösung.

Die verwaltete, zuverlässige und überwachte Übertragung von Dateien kann eine fundamentale Anforderung in einem Unternehmen sein. IBM WebSphere MQ Version 7.5 bietet eine Managed File Transfer-Funktionalität als Teil der integrierten Messaging-Plattform. Managed File Transfer ermöglicht Ihnen die nahtlose Integration von Dateien in Ihre Messaging-Infrastruktur, entweder durch einfache Dateiübertragungen oder durch Teilnehmer, die die vollständige Messaging-Funktionalität nutzen. Sie finden weitere Informationen zu dieser Funktion unter [Einführung in WebSphere MQ Managed File Transfer](#).

Dieses Szenario vermittelt Ihnen ein grundlegendes Verständnis davon, wie Dateien in die einfachste Messaging-Topologie von IBM WebSphere MQ integriert werden. Hierfür durchlaufen Sie ein IBM WebSphere MQ-Basisszenario, bei dem eine Datei von einem Ort an einen anderen Ort verschoben wird. Auch wenn dieses Szenario auf einen einzelnen Computer begrenzt ist, gibt es Ihnen doch die Möglichkeit, Erfahrungen hinsichtlich der Konfiguration der Umgebung zu sammeln, und stellt so eine wichtige Grundlage für spätere Szenarios dar. Die späteren Szenarios veranschaulichen, wie Sie IBM WebSphere MQ für die Übertragung von Dateien in einem Netz verwenden können. Außerdem erhalten Sie einen Einblick, wie in der Praxis bestehende Probleme im Geschäftsbereich mit der Komponente 'Managed File Transfer' gelöst werden können.

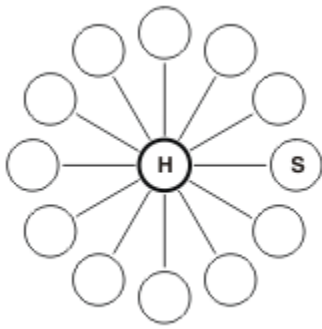
Dabei werden Grundkenntnisse über IBM WebSphere MQ vorausgesetzt. Insbesondere müssen Sie mit dem Konzept eines Warteschlangenmanagers und der Basiskonfiguration und -administration von IBM WebSphere MQ bis hin zur Verwendung von Befehlen wie **runmqsc** und mit dem IBM WebSphere MQ Explorer vertraut sein.

In diesem Szenario erfahren Sie, wie Sie mit IBM WebSphere MQ auf einem einzelnen Computer die Übertragung einer Datei von einem Ort an einen anderen Ort einleiten und verfolgen können. Dabei üben Sie die Installation, Konfiguration und Verwendung der Funktion einer zuverlässigen und verwalteten Dateiübertragung in IBM WebSphere MQ Version 7.5.



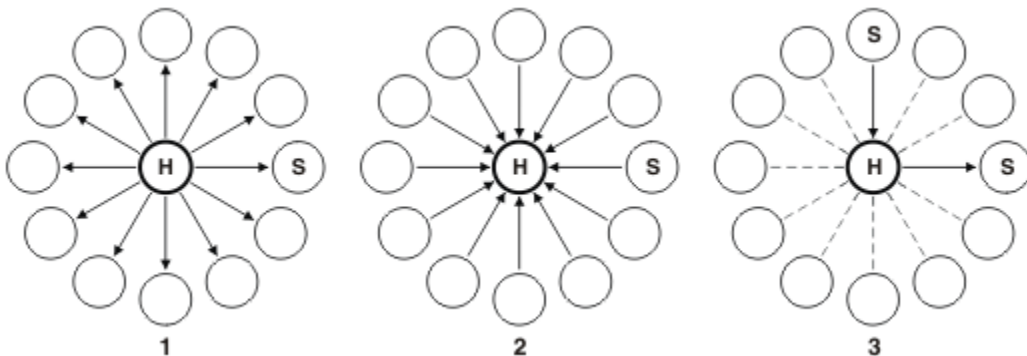
### Beispieltopologie einer Dateiübertragung

Dieses Szenario sowie die nachfolgenden Szenarios stützen sich auf eine Hub-and-Spoke-Topologie. Im Diagramm wird die konzeptionelle Hub-and-Spoke-Topologie dargestellt, die sich aus dem Hub (H) und mehreren Spokes (S) zusammensetzt.



Eine derartige Topologie ist in vielen Geschäftsszenarios üblich, beispielsweise für die Ausführung der folgenden Aktionen:

1. Senden von Dateien von einem zentralen Stützpunkt (Hub) an viele entlegene Knoten (Spokes).
2. Zusammentragen der Dateien aus vielen Knoten (Spokes) an einem einzelnen Ort (Hub).
3. Übertragen von Dateien von einem Spoke an einen anderen Spoke.



Möglicherweise unterscheidet sich Ihre Topologie von diesem Beispiel, die Grundsätze und Befehle in diesem Szenario lassen sich jedoch jederzeit erweitern, damit sie für alle IBM WebSphere MQ-Netze eingesetzt werden können.

## Lösung planen

In diesem Abschnitt wird die Dateiübertragung zu oder von einem Computer mittels Scripts beschrieben. Voraussetzungen hierfür sind: Kenntnisse des Sicherheitsmodells, Vorhandensein der Voraussetzungen und Lizenzen, Installation und Konfiguration von IBM WebSphere MQ sowie die Vorbereitung der Benutzer und Gruppen.

Wenn Sie eine Datei von Punkt A nach Punkt B übertragen möchten, erstellen Sie auf einem Computer eine Hub-and-Spoke-Topologie. Diese Topologie umfasst den Hub, einen IBM WebSphere MQ-Warteschlangenmanager und zwei Peripherien (Spokes) je Dateiübertragungsagent.

Dateiübertragungsagenten sind Java-Prozesse, die auf dem Computer ausgeführt werden und Dateien an andere Agenten und von anderen Agenten übertragen. In diesem Szenario definieren Sie eine Dateiübertragung, die mithilfe der Dateiübertragungsagenten über zwei Mechanismen eine Beispieldatei von einer Position an eine andere Position verschiebt:

1. Mit dem grafischen IBM WebSphere MQ Explorer.
2. Über die Befehlszeile.

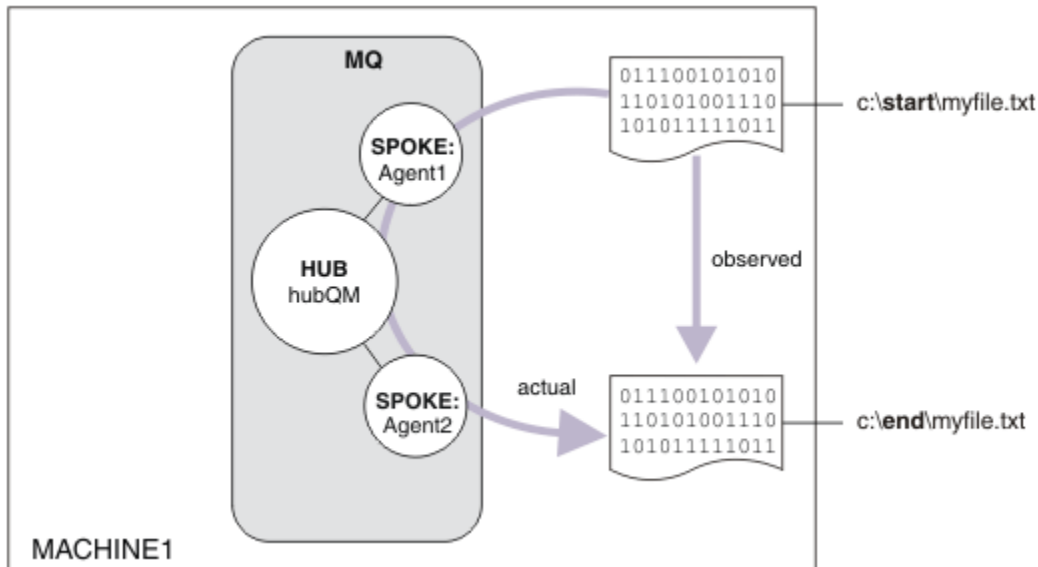
Bei dem Szenario wird vorausgesetzt, dass Sie über ein Windows-System verfügen. Bei einem UNIX-System müssen Sie die Pfade und Befehle entsprechend ersetzen und sicherstellen, dass Sie über Lese- und Schreibzugriff für alle relevanten Verzeichnisse verfügen. Ebenfalls wird in diesem Szenario davon ausgegangen, dass Sie mit den grundlegenden Funktionen eines Warteschlangenmanagers vertraut sind.



Wenn Sie Dateien an einen Computer oder von diesem übertragen möchten, muss auf dem betreffenden Computer ein Dateiübertragungsagent ausgeführt werden. Jeder Agent verbindet sich mit einem IBM WebSphere MQ-Warteschlangenmanager und verwendet IBM WebSphere MQ für die Kommunikation mit anderen Agenten. Sie finden weitere Informationen unter [Übersicht über die Topologie von WebSphere MQ Managed File Transfer](#).

IBM WebSphere MQ Version 7.5 verfügt über die beiden Beispielskripts `createHub` und `addSpoke`, mit denen Sie eine Topologie zur Dateiübertragung schnell und einfach erstellen können. Diese Skripts ermöglichen Ihnen den Aufbau dieser Topologie vom Hub ausgehend. Es handelt sich bei diesen Skripts lediglich um Beispiele, die Sie an Ihre Anforderungen anpassen können. Falls Sie die Skripts ändern möchten, kopieren Sie sie zunächst an einen Ort in Ihrem eigenen Benutzerverzeichnis.

Die Topologie wird mithilfe der Beispielskripts `createHub` und `addSpoke` erstellt.



Beim Start einer Übertragung wird ersichtlich, wie die Datei von einem Ort an einen anderen Ort kopiert wird. In diesem einfachen Szenario kann ein einzelner Dateiübertragungsagent verwendet werden, der sowohl die Quelle- als auch die Zieldateien verwaltet. Es werden jedoch zwei Dateiübertragungsagenten verwendet, damit Sie ein besseres Beispiel für die Funktionsweise dieses Aufbaus erhalten, bevor Sie zu einer Topologie mit mehreren Computern übergehen.

## Konzept des Sicherheitsmodells

Die Skripts `createHub` und `addSpoke` konfigurieren eine Topologie für die Dateiübertragung mit den folgenden Sicherheitsmerkmalen:

- Der Zugriff auf IBM WebSphere MQ teilt sich auf drei Rollen auf:
  1. IBM WebSphere MQ-Administrator zur Konfiguration von IBM WebSphere MQ und zur Ausführung der Skripts `createHub` und `addSpoke`.
  2. FTAGENTS zum Starten und Stoppen von Agenten sowie zur Interaktion mit diesen.
  3. FTUSERS zum Einleiten von Dateiübertragungen.

Sie finden weitere Informationen zur Konfiguration des Zugriffs unter [Berechtigungen für spezielle WebSphere MQ Managed File Transfer-Ressourcen](#).

- Alle eingehenden Verbindungen von Agenten werden beim Hub dem einzelnen Benutzer FTAGENT zugeordnet.
- Für die Authentifizierung einzelner Agenten wird eine schwache IP-basierte Authentifizierung verwendet.

Das Sicherheitsmodell im Script ist aus Gründen einer besseren Transparenz nicht vollständig abgeschotet. Bevor Sie diese Scripts in der Produktionsumgebung verwenden, müssen Sie mit den Anforderungen und Sicherheitsbedrohungen Ihrer eigenen Topologie vertraut sein. Daher müssen Sie abwägen, ob die folgenden möglichen Sicherheitslücken und Empfehlungen in Ihrem Fall relevant sind:

- Jeder Benutzer kann die Identität eines anderen Benutzers annehmen. Erwägen Sie eine differenziertere Unterteilung im Objektzugriffsmodell für Dateiübertragungsressourcen.
- Jeder Agent kann die Identität eines anderen Agenten annehmen. Erwägen Sie eine striktere Authentifizierung (beispielsweise TLS/SSL) und eine differenziertere Unterteilung im Objektzugriffsmodell für Dateiübertragungsressourcen.
- Die Schnittstelle zwischen der Dateisystemsicherheit und der IBM WebSphere MQ-Sicherheit wird nicht berücksichtigt. Erwägen Sie die Implementierung von Sandboxing für Dateien und machen Sie sich die Auswirkungen bewusst, die sich aus den Berechtigungen der Konfigurationsdateien des Agenten ergeben. Sie finden weitere Informationen zum Sandboxing unter [Sandboxes](#).
- Die Schnittstelle zwischen dem Agenten und dem Betriebssystem wird nicht beschrieben. Erwägen Sie die Implementierung des Sandboxing für Dateien. Ausführliche Informationen hierzu finden Sie unter [Sandboxes](#).

Sie finden weitere Informationen zur Sicherheit, zu IBM WebSphere MQ und zu Dateiübertragungen unter [Weitere Schritte](#).

## Voraussetzungen und Lizenzen

Sie benötigen die folgenden Elemente:

- Ein Testcomputer, der die Hardware- und Betriebssystemvoraussetzungen für IBM WebSphere MQ erfüllt (Details hierzu finden Sie unter <https://www.ibm.com/support/docview.wss?uid=swg27006467>), und auf dem weder IBM WebSphere MQ noch IBM WebSphere MQ-Daten installiert sind.
- In: IBM WebSphere MQ Version 7.5. Sie können eine Testversion von <https://www.ibm.com/developer-works/downloads/ws/wmq/herunterladen>.

## Computer vorbereiten

Stellen Sie sicher, dass Ihr Testcomputer die Voraussetzungen für die Installation von IBM WebSphere MQ Version 7.5 erfüllt (siehe [Voraussetzungen überprüfen](#)).

Stellen Sie sicher, dass Ihr Computer ordnungsgemäß für die Installation vorbereitet ist (siehe [Lokale Installation über die Befehlszeile überprüfen](#)).

## IBM WebSphere MQ Version 7.5 installieren

Installieren Sie den IBM WebSphere MQ Version 7.5-Server mit folgenden Komponenten: Server, IBM WebSphere MQ Explorer, IBM WebSphere MQ Advanced Managed File Transfer Agent und IBM WebSphere MQ Managed File Transfer Command Line Tools. Ausführliche Informationen finden Sie im Abschnitt [Zu installierende Komponenten auswählen](#).

Überlegen Sie sich, wie Sie IBM WebSphere MQ verwalten möchten. Zur Verwaltung von IBM WebSphere MQ haben Sie folgende Möglichkeiten:

- Einrichtung einer entsprechenden Umgebung mit dem Befehl **setmqenv**. Weitere Informationen finden Sie im Abschnitt [setmqenv](#).
- Aufruf vollständig qualifizierter IBM WebSphere MQ-Verwaltungsbefehle.

Bei diesem Szenario wird vorausgesetzt, dass Sie einen bereinigten Computer verwenden, auf dem weder IBM WebSphere MQ noch IBM WebSphere MQ File Transfer Edition installiert ist. Ist dies nicht der Fall, müssen Sie herausfinden, ob eine Koexistenz unterstützt wird, oder das Installationsverfahren und die Konfiguration der Umgebungen entsprechend anpassen. Weitere Informationen finden Sie im Abschnitt [Mehrere Installationen](#).

## Benutzer und Gruppen erstellen

Das Sicherheitsmodell setzt folgende Benutzer und Gruppen voraus:

### Benutzer

- mqmAdmin

IBM WebSphere MQ-Administrator, also ein Mitglied der Gruppe 'mqm' bzw. unter Windows ein Mitglied der Administratorgruppe. Sie müssen diesen Benutzer erstellen und als Mitglied zur Gruppe 'mqm' hinzufügen.

- ftuser

Sie müssen diesen Benutzer erstellen und als Mitglied zur Gruppe FTUSERS hinzufügen. Nehmen Sie diesen Benutzer nicht in die Gruppe 'mqm' auf, um die Möglichkeit von Angriffen auf den Warteschlangenmanager auf Verwaltungsebene in dieser Hinsicht auszuschließen.

- ftagent

Sie müssen diesen Benutzer erstellen und als Mitglied zur Gruppe FTAGENTS hinzufügen. Nehmen Sie diesen Benutzer nicht in die Gruppe 'mqm' auf, um die Möglichkeit von Angriffen auf den Warteschlangenmanager auf Verwaltungsebene in dieser Hinsicht auszuschließen.

### Gruppen

- mqm

Wird automatisch bei der IBM WebSphere MQ-Installation erstellt. Mitglieder dieser Gruppe können IBM WebSphere MQ und die zugehörigen Ressourcen verwalten.

- FTUSERS

Diese Gruppe müssen Sie erstellen. Mitglieder dieser Gruppe können Dateiübertragungen einleiten.

- FTAGENTS

Diese Gruppe müssen Sie erstellen. Mitglieder dieser Gruppe können Übertragungsagenten starten und stoppen. Diese dienen als Endpunkte, welche die Dateiübertragungen in Ihrem Netz verarbeiten.

## IBM WebSphere MQ für Dateiübertragungen konfigurieren

Konfigurieren Sie IBM WebSphere MQ für Dateiübertragungen, indem Sie mithilfe der Beispielscripts `createHub` und `addSpoke` die Topologie für die Basisdateiübertragung mit dem Script-Szenario erstellen.

### Vorgehensweise

1. Bestimmen Sie einen freien Port, mit dem sich ferne Agenten (in späteren Szenarios definiert) verbinden können, zum Beispiel 1414.
2. Bestimmen Sie einen geeigneten Namen für einen Warteschlangenmanager, der als Hub des Dateiübertragungsnetzes handeln wird. Beispiel: `hubQM`.
3. Als Benutzer `mqmAdmin` aus dem IBM WebSphere MQ -Verzeichnis 'bin', `<MQ_INSTALL_ROOT>/bin`:
  - a) Erstellen Sie den Hub.

Geben Sie den folgenden Befehl ein:

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\createHub hubQmgr=hubQM hubPort=1414
```

- b) Fügen Sie den ersten Agentenspoke hinzu (in `BINDINGS`, da es sich um einen lokalen Agenten für den Warteschlangenmanager handelt) und nennen Sie den Agenten `AGENT1`.

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT1  
hubQmgr=hubQM connectionMode=BINDINGS
```

- c) Fügen Sie den zweiten Agentenspoke hinzu (in `BINDINGS`, da es sich um einen lokalen Agenten für den Warteschlangenmanager handelt) und nennen Sie den Agenten `AGENT2`.

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT2
hubQmgr=hubQM connectionMode=BINDINGS
```

4. Gehen Sie als Benutzer `ftagent` wie folgt vor:

a) Starten Sie AGENT1 mit dem folgenden Befehl:

```
fteStartAgent -p hubQM AGENT1
```

Wenn Sie mehrere Hubs konfiguriert haben, stellt der Bereich `-p hubQM` des Befehls sicher, dass Sie den richtigen Hub wählen. Bei nur einem Hub ist diese Angabe nicht erforderlich.

b) Starten Sie AGENT2 mit dem folgenden Befehl:

```
fteStartAgent -p hubQM AGENT2
```

c) Listen Sie die Agenten auf, um zu prüfen, ob sie ordnungsgemäß ausgeführt werden.

```
fteListAgents -p hubQM
```

Sie sehen die folgende Ausgabe:

Agent Name:	Queue Manager Name:	Status:
AGENT1	hubQM	READY
AGENT2	hubQM	READY

## Lösung implementieren

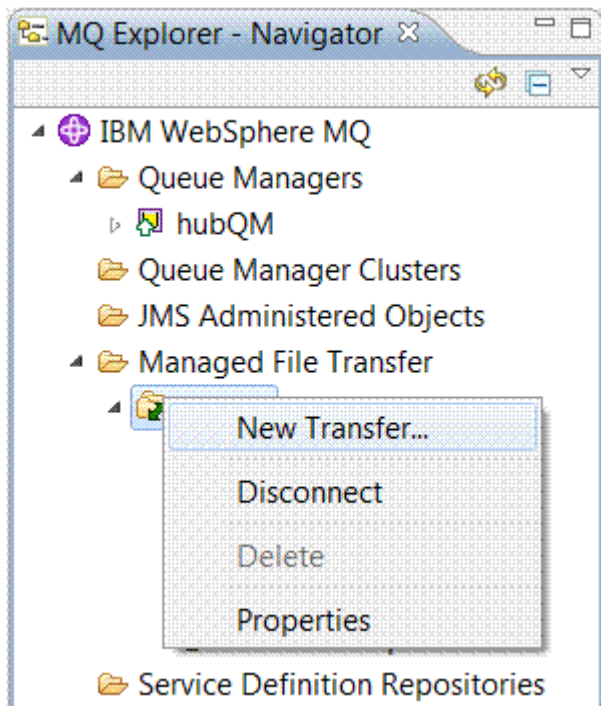
Die Implementierung der Lösung in diesem Szenario umfasst auch die Verwendung von IBM WebSphere MQ Explorer für die Definition und Einleitung einer Dateiübertragung. Sie können den Übertragungsfortschritt überwachen und seinen Erfolg durch die Untersuchung des Dateisystems überprüfen.

### Informationen zu diesem Vorgang

Um zu vermeiden, dass IBM WebSphere MQ Explorer für den Benutzer `ftuser` konfiguriert wird (der über keine Verwaltungsrechte verfügt), sollten Sie diesen Bereich des Szenarios als Benutzer `mqmAdmin` ausführen. Wenn Sie diese Funktion aktivieren möchten, lesen Sie den Abschnitt [Sicherheit](#).

### Vorgehensweise

1. Geben Sie die Quellendatei und das Zielverzeichnis an:
  - a) Erstellen Sie eine zu übertragende Beispieldatei, z. B. `C:\start\myfile.txt`.
  - b) Geben Sie ein vorhandenes Zielverzeichnis für die Übertragung dieser Datei an, beispielsweise `C:\end\`.
2. Starten Sie als Administrator `mqmAdmin` in diesem Beispiel IBM WebSphere MQ Explorer. Starten Sie das Programm über das Startmenü (oder eine entsprechende Option) oder führen Sie den Befehl **MQExplorer** aus. Sie finden ausführliche Informationen hierzu unter [WebSphere MQ Explorer starten](#).
3. Klicken Sie im Bereich **Managed File Transfer** mit der rechten Maustaste auf die Konfiguration `hubQM` und wählen Sie **Verbinden** aus.
4. Klicken Sie im Abschnitt **Verwaltete Dateiübertragung** im IBM WebSphere MQ -Navigator mit der rechten Maustaste auf `hubQM` und wählen Sie **Neue Übertragung** aus, um den Assistenten 'Neue Übertragung' zu starten.



5. Wählen Sie im Menü AGENT1 als Quellenagent und AGENT2 als Zielagent aus.

Source agent  
 Name: AGENT1  
 Type: Standard

Destination agent  
 Name: AGENT2  
 Type: Standard

6. Klicken Sie auf **Weiter**.

**New transfer**

*Click 'Add' to begin adding items to the transfer*

Source	Destination	Mode	Disposition	Overwrite

Buttons: Add..., Copy..., Edit..., Remove

7. Klicken Sie auf **Hinzufügen ...** um mit der Auswahl der Dateien zu beginnen, die von AGENT1 zu AGENT2 übertragen werden sollen.

- Klicken Sie im Teilfenster **Quelle** auf **Durchsuchen** und blättern Sie zu der zuvor bestimmten Datei. Beispiel: C:\start\myFile.txt. Stellen Sie sicher, dass das Kontrollkästchen **Nach erfolgreicher Übertragung Quelldatei löschen** aktiviert ist.

Source

Agent: AGENT1  
 Type: Standard agent  
 Type: File

File name: C:\start\myFile.txt

Remove source file if the transfer is successful

- Wählen Sie im Rahmen **Ziel** den **Typ Directory** aus und geben Sie dann das zuvor angegebene Zielverzeichnis ein, z. B. C:\end\.

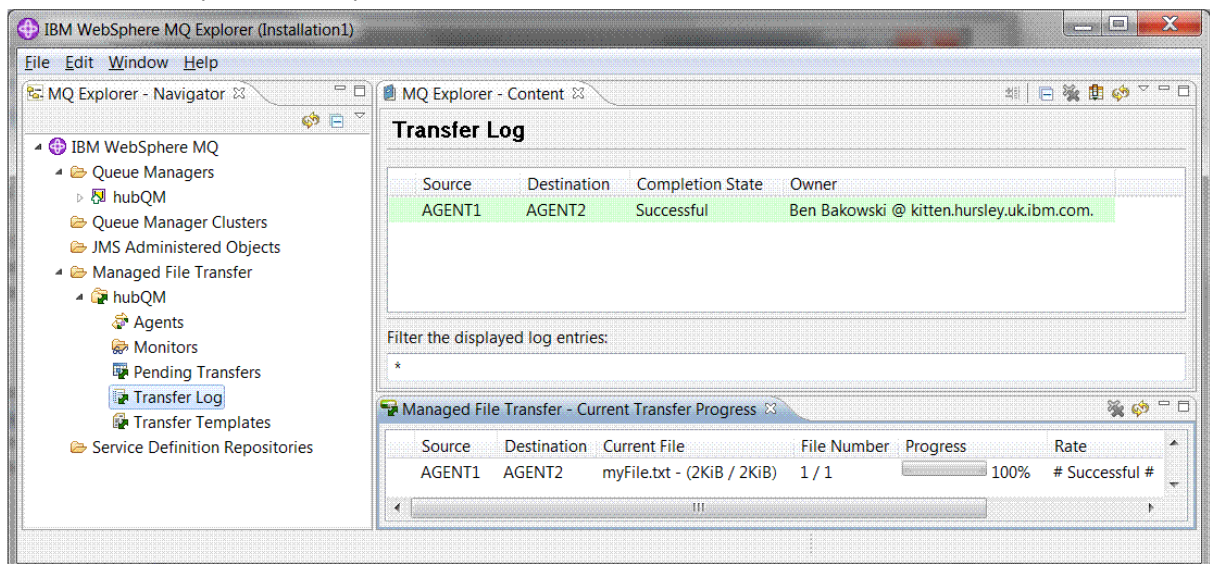
Destination

Agent: AGENT2  
 Type: Standard agent  
 Type: Directory

Directory: c:\end

Overwrite files if present

- Klicken Sie auf **OK** und dann auf **Fertigstellen**, damit die Übertragung beginnt.
- Sie können den Übertragungsfortschritt auf der Registerkarte 'Aktueller Übertragungsfortschritt' unten in IBM WebSphere MQ Explorer überwachen:



- Sie können das Dateisystem auch manuell überprüfen, um sich zu vergewissern, dass die neue Datei vorhanden ist, beispielsweise C:\end\myfile.txt.

## Datei über die Befehlszeile übertragen

In diesem Abschnitt, in dem die Verwendung der Befehlszeile für die Einleitung einer Übertragung gezeigt wird, erhalten Sie einen Einblick in die Flexibilität von Dateiübertragungen. Im vorliegenden Szenario wird dies zwar nicht abgedeckt, Sie können aber mithilfe dieser Prinzipien und der Ant-Scriptingtechnologie wesentlich leistungstärkere Dateiübertragungsszenarios definieren und implementieren.

### Vorgehensweise

1. Übertragen Sie die übertragene Datei wieder zurück in ihr ursprüngliches Verzeichnis.
2. Geben Sie als Benutzer `ftuser` den Befehl **fteCreateTransfer** ein, um die Übertragung Ihrer Datei von `C:\start\myfile.txt` an `C:\end\myfile.txt` einzuleiten:

```
fteCreateTransfer -sa AGENT1 -sm hubQM -sd delete -da AGENT2 -dm hubQM -w -dd C:\end\  
C:\start\myfile.txt
```

- `-sa AGENT1` definiert den Quellenagenten (also den Agenten, von dem die Datei übertragen wird) als AGENT1.
- `-sm hubQM` definiert den Warteschlangenmanager, mit dem sich der Quellenagent AGENT1 verbindet.
- `-sd delete` gibt an, dass die Quellendatei nach einer erfolgreichen Übertragung gelöscht werden soll.
- `-da AGENT2` definiert den Zielagenten (also den Agenten, an den die Datei übertragen wird) als AGENT2.
- `-w` legt fest, dass der Befehl **fteCreateTransfer** auf die Bestätigung seiner erfolgreichen Ausführung wartet.
- `-dd C:\end\` definiert das Zielverzeichnis als `C:\end\`.
- `C:\start\myfile.txt` definiert die zu übertragende Datei.

Sie finden weitere Informationen unter [fteCreateTransfer \(Erstellung einer neuen Dateiübertragung\)](#).

3. Vergewissern Sie sich, dass die Dateiübertragung erfolgreich verlaufen ist, indem Sie das Dateiübertragungsprotokoll in IBM WebSphere MQ Explorer überprüfen oder das Dateisystem manuell untersuchen.

### Nächste Schritte

Möglicherweise möchten Sie weitere Funktionen der Dateiübertragungsfunktionalität über externe Medien erkunden. Unter

- [„Dateiübertragung zwischen zwei Computern mithilfe der Scripts“](#) auf Seite 173
- Höherer Schutz Ihrer Umgebung. Ihre eigenen Sicherheitsanforderungen erfordern möglicherweise ein anderes Zugriffsmodell als das im vorliegenden Szenario verwendete Modell. Weitere Informationen zu den Best Practices in diesem Bereich finden Sie im Abschnitt [WebSphere MQ File Transfer Edition V7 schützen](#).

## Basisdateiübertragung im Detail

Sie können Dateien mit IBM WebSphere MQ Version 7.5 auf verschiedene Arten übertragen. Lesen Sie die Abschnitte in diesem Abschnitt, um zu verstehen, was in diesem Szenario behandelt wird, die Gründe, warum ein Unternehmen dem Szenario folgen möchte, die beteiligten Benutzerrollen und eine Übersicht über die vom Szenario vorgeschlagene Lösung.

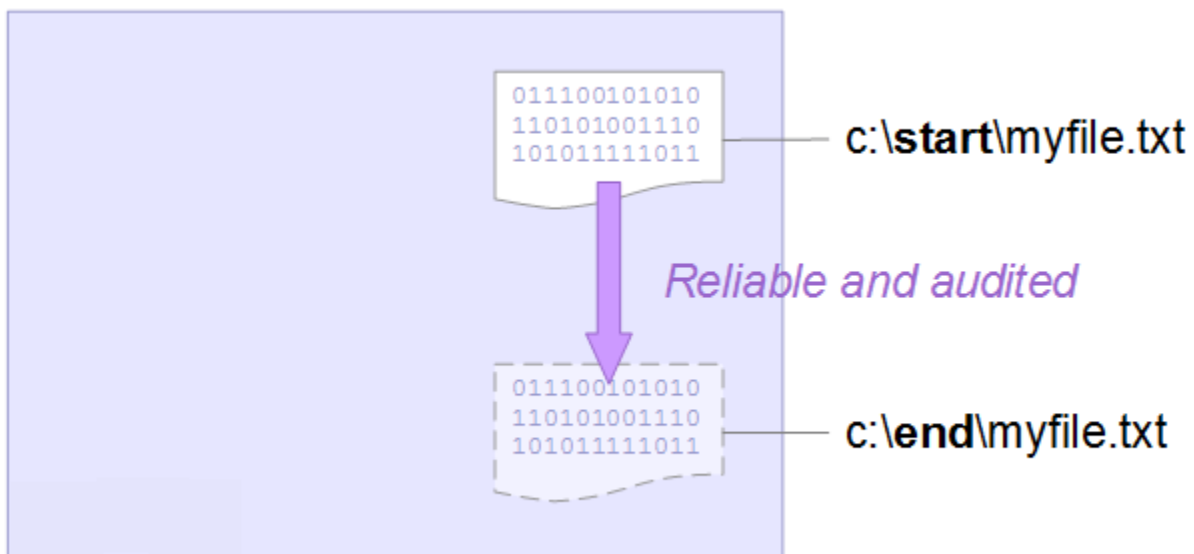
Die verwaltete, zuverlässige und überwachte Übertragung von Dateien kann eine fundamentale Anforderung in einem Unternehmen sein. IBM WebSphere MQ Version 7.5 bietet eine Managed File Transfer-Funktionalität als Teil der integrierten Messaging-Plattform. Managed File Transfer ermöglicht Ihnen die nahtlose Integration von Dateien in Ihre Messaging-Infrastruktur, entweder durch einfache Dateiübertragungen oder durch Teilnehmer, die die vollständige Messaging-Funktionalität nutzen.

Dieses Szenario vermittelt Ihnen ein grundlegendes Verständnis davon, wie Dateien in die einfachste Messaging-Topologie von IBM WebSphere MQ integriert werden. Arbeiten Sie dieses einfache IBM WebSphere MQ-Szenario durch, in dem eine Datei von einem Ort an einen anderen verschoben wird. Auch wenn dieses Szenario auf einen einzelnen Computer begrenzt ist, gibt es Ihnen doch die Möglichkeit, Erfahrungen hinsichtlich der Konfiguration der Umgebung zu sammeln, und stellt so eine wichtige Grundlage für spätere Szenarios dar. Dieses Szenario zeigt, wie IBM WebSphere MQ zur Übertragung von Dateien in einem Netz verwendet wird, und führt anschließend vor, wie mithilfe der Komponente Managed File Transfer reale Geschäftsprobleme gelöst werden können.

Dabei werden Grundkenntnisse über IBM WebSphere MQ vorausgesetzt. Insbesondere müssen Sie mit dem Konzept eines Warteschlangenmanagers und der Basiskonfiguration und -administration von IBM WebSphere MQ bis hin zur Verwendung von Befehlen wie **runmqsc** und mit dem IBM WebSphere MQ Explorer vertraut sein.

## Übersicht

Mit IBM WebSphere MQ kann die Übertragung einer Datei von einem Ort an einen anderen auf einem einzelnen Computer eingeleitet und überwacht werden. Sie können Erfahrungen hinsichtlich der Installation, Konfiguration und Verwendung der Funktion für verwaltete Dateiübertragungen (Managed File Transfer) in IBM WebSphere MQ Version 7.5 sammeln und lernen auf diese Weise, wie Sie das Produkt zur Lösung realer Geschäftsprobleme mit Dateiübertragungen einsetzen können.



Weitere Informationen zur Planung der Dateiübertragungsfunktion finden Sie im Abschnitt [Einführung in WebSphere MQ Managed File Transfer](#).

## Lösung planen

Übertragen von Dateien an einen Computer oder von einem Computer, Dateiübertragungsagenten (File Transfer Agents), Voraussetzungen, Lizenzen, Vorbereitung der Computer und Installation und Konfiguration von IBM WebSphere MQ für Dateiübertragungen in einem Basisszenario der Dateiübertragung.

Definieren Sie einen IBM WebSphere MQ -Warteschlangenmanager und zwei Dateiübertragungsagenten, um eine Datei von Punkt A nach Punkt B zu übertragen. Mit dem Konzept eines Warteschlangenmanagers sind Sie bereits vertraut, möglicherweise kennen Sie jedoch noch keine Dateiübertragungsagenten. Dateiübertragungsagenten sind Java-Prozesse, die auf dem Computer ausgeführt werden und Dateien an andere Agenten und von anderen Agenten übertragen. In diesem Szenario definieren Sie eine Dateiübertragung, die mithilfe der Dateiübertragungsagenten über zwei Mechanismen eine Beispieldatei von einer Position an eine andere Position verschiebt:

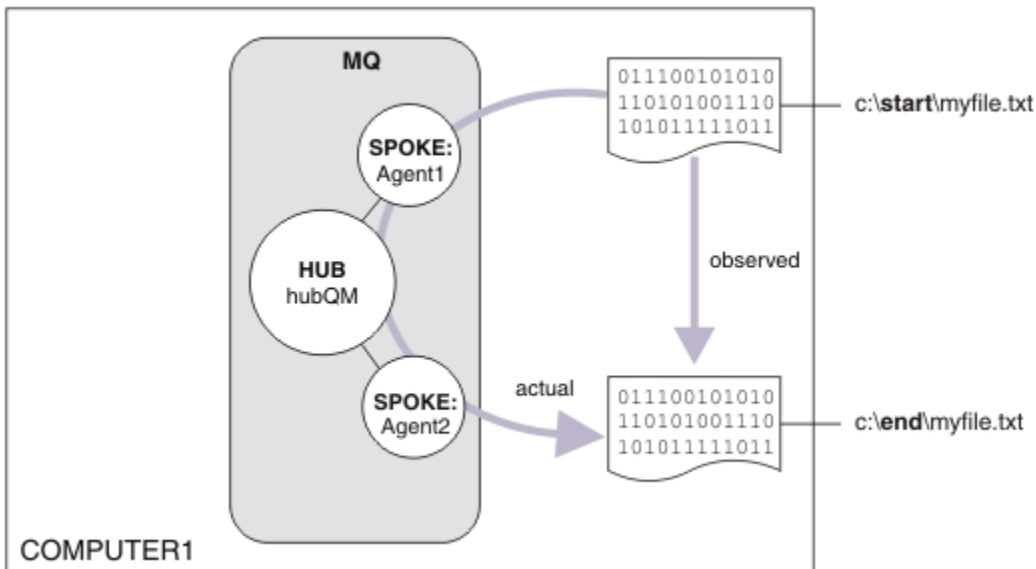
1. Mit dem grafischen IBM WebSphere MQ Explorer.
2. Über die Befehlszeile.



Der Walkthrough setzt voraus, dass Sie über ein Windows -System verfügen. Bei einem UNIX-System müssen Sie die Pfade und Befehle entsprechend ersetzen und sicherstellen, dass Sie über Lese- und Schreibzugriff für alle relevanten Verzeichnisse verfügen. Außerdem wird davon ausgegangen, dass Sie mit den grundlegenden Funktionen eines Warteschlangenmanagers vertraut sind.

Wenn Sie Dateien an einen Computer oder von diesem übertragen möchten, muss auf dem betreffenden Computer ein Dateiübertragungsagent ausgeführt werden. Jeder Agent verbindet sich mit einem IBM WebSphere MQ-Warteschlangenmanager und verwendet IBM WebSphere MQ für die Kommunikation mit anderen Agenten. Sie finden weitere Informationen unter [Übersicht über die Topologie von WebSphere MQ Managed File Transfer](#).

In diesem Szenario kann ein einzelner Dateiübertragungsagent verwendet werden, der sowohl die Quelle als auch die Zieldateien verwaltet. Beim vorliegenden Beispiel werden jedoch zwei Dateiübertragungsagenten verwendet, damit Sie einen praxisorientierten Einblick in die Funktionsweise von Dateiübertragungen in einem typischen Umfeld mit mehreren Netzen erhalten.



Dieses Diagramm zeigt die Verwendung von IBM WebSphere MQ zur Veranschaulichung einer Basisdateiübertragung. Es wird demonstriert, wie die Datei von einer Position an eine andere Position kopiert wird.

## Voraussetzungen und Lizenzen

Sie benötigen die folgenden Elemente:

- Ein Testcomputer, der die Hardware- und Betriebssystemvoraussetzungen für IBM WebSphere MQ erfüllt (Details hierzu finden Sie unter <https://www.ibm.com/support/docview.wss?uid=swg27006467>), und auf dem weder IBM WebSphere MQ noch IBM WebSphere MQ-Daten installiert sind.
- In: IBM WebSphere MQ Version 7.5. Sie können eine Testversion von <https://www.ibm.com/developerworks/downloads/ws/wmq/herunterladen>.

## Computer vorbereiten

Stellen Sie sicher, dass Ihr Testcomputer die Voraussetzungen für eine Installation von IBM WebSphere MQ Version 7.5 erfüllt (siehe [Voraussetzungen überprüfen](#)).

Stellen Sie sicher, dass Ihr Computer ordnungsgemäß auf die Installation vorbereitet wurde (siehe [Lokale Installation über die Befehlszeile überprüfen](#)).

## WebSphere MQ für Dateiübertragungen konfigurieren

1. Erstellen Sie den Warteschlangenmanager hubQM.

Geben Sie den Befehl `crtmqm hubQM` ein. Weitere Informationen finden Sie unter [crtmqm](#).

2. Starten Sie den Warteschlangenmanager `hubQM`.

Geben Sie den Befehl `strmqm hubQM` ein. Ausführliche Informationen finden Sie im Abschnitt [strmqm](#).

3. Konfigurieren Sie den Warteschlangenmanager `hubQM` für die Koordination von Dateiübertragungen.

- a. Erstellen Sie die Eigenschaftendateien und das Verzeichnis des Koordinations-Warteschlangenmanagers für IBM WebSphere MQ, indem Sie folgenden Befehl eingeben:

```
fteSetupCoordination -coordinationQMgr hubQM
```

Mit diesem Befehl werden Eigenschaftendateien und das Verzeichnis des Koordinations-Warteschlangenmanagers für IBM WebSphere MQ erstellt. In diesem Fall handelt `hubQM` als Koordinations-Warteschlangenmanager, der Prüfungs- und Dateiübertragungsdaten im Broadcastbetrieb sendet. Wenn der Befehl **`fteSetupCoordination`** nicht zur Verfügung steht, haben Sie WebSphere MQ mit den Standardeinstellungen installiert. Durch die Installation der zusätzlich bereitgestellten Pakete (z. B. Managed File Transfer) wird dieses Problem behoben.

Weitere Informationen finden Sie in den Abschnitten [IBM WebSphere MQ Managed File Transfer - Topologieübersicht](#) und [fteSetupCoordination \(Koordinationsdetails konfigurieren\)](#).

- b. Konfigurieren Sie `hubQM`, damit er als Koordinations-Warteschlangenmanager handelt. Geben Sie hierfür folgenden Befehl ein:

```
runmqsc hubQM < <filepath from previous statement>
```

- c. Definieren Sie, welcher Warteschlangenmanager Dateiübertragungsbefehle verarbeitet (in diesem Fall `hubQM`).

```
fteSetupCommands -connectionQMgr hubQM
```

Sie finden weitere Informationen unter [fteSetupCommands \(Erstellung der Datei 'command.properties'\)](#).

4. Erstellen Sie Ihren ersten Dateiübertragungsagenten, der den Namen `AGENT1` erhält.

- a. Bereiten Sie den Dateiübertragungsagenten `AGENT1` vor. Dies umfasst auch die MQSC-Skripts, die Sie für den Warteschlangenmanager ausführen müssen, mit dem sich der Agent verbindet (in diesem Fall `hubQM`). Geben Sie hierfür folgenden Befehl ein:

```
fteCreateAgent -agentName AGENT1 -agentQMgr hubQM
```

Sie finden weitere Informationen unter [fteCreateAgent \(Erstellung eines WebSphere MQ Managed File Transfer-Agenten\)](#).

- b. Konfigurieren Sie `hubQM` für die Verarbeitung des von Ihnen erstellten Agenten.

```
runmqsc hubQM < <location of AGENT1_create.mqsc>
```

Die Position der Datei hängt davon ab, wo Sie IBM WebSphere MQ installiert haben.

5. Erstellen Sie Ihren zweiten Dateiübertragungsagenten, der den Namen `AGENT2` erhält.

```
fteCreateAgent -agentName AGENT2 -agentQMgr hubQM
```

```
runmqsc hubQM < <location of AGENT2_create.mqsc>
```

6. Starten Sie `AGENT1`.

```
fteStartAgent AGENT1
```

Sie finden weitere Informationen unter [fteStartAgent \(Starten eines WebSphere MQ Managed File Transfer-Agenten\)](#).

## 7. Starten Sie AGENT2.

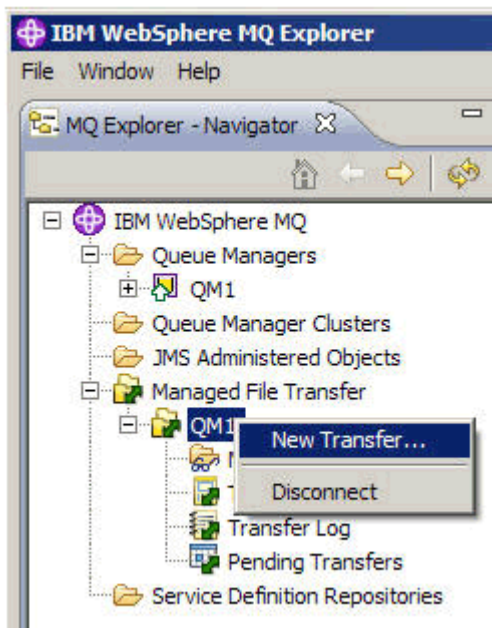
```
fteStartAgent AGENT2
```

## Lösung implementieren

Die Implementierung der Lösung in diesem Szenario umfasst die Verwendung von IBM WebSphere MQ Explorer zum Definieren und Einleiten einer Dateiübertragung. Sie können den Übertragungsfortschritt überwachen und seinen Erfolg durch die Untersuchung des Dateisystems überprüfen.

## Vorgehensweise

1. Geben Sie die Quelldatei und das Zielverzeichnis an:
  - a) Erstellen Sie eine zu übertragende Beispieldatei, z. B. C:\start\myfile.txt.
  - b) Geben Sie ein vorhandenes Zielverzeichnis für die Übertragung dieser Datei an, beispielsweise C:\end\.
2. Starten Sie IBM WebSphere MQ Explorer. Starten Sie das Programm über das Menü "Start" (bzw. über eine funktionale Entsprechung) oder führen Sie den Befehl **MQExplorer** aus. Weitere Informationen finden Sie unter [IBM WebSphere MQ Explorer starten](#).
3. Klicken Sie im IBM WebSphere MQ Explorer -Navigator auf **Verwaltete Dateiübertragung**, klicken Sie mit der rechten Maustaste auf **QM** und wählen Sie **Neue Übertragung** aus, um den Assistenten 'Neue Übertragung' zu starten.



4. Wählen Sie **AGENT1** als Quellenagent im Abschnitt **From** (Von) aus:

From:

Agent:	AGENT1
Type:	File
File:	

Include subdirectories

5. Geben Sie den Pfad zu der zuvor erstellten Datei ein, beispielsweise C:\start\myfile.txt.

From:

Agent: AGENT1

Type: File

File: C:\Users\Ben Bakowski\Desktop\FTETEST\START\myDemoTransfer.txt

Include subdirectories

6. Wählen Sie im Abschnitt **An** den Agenten **AGENT2** als Zielagenten aus.

7. Geben Sie das zuvor bestimmte Zielverzeichnis ein. Beispiel: C : \end\.

To:

Agent: AGENT2

Type: File

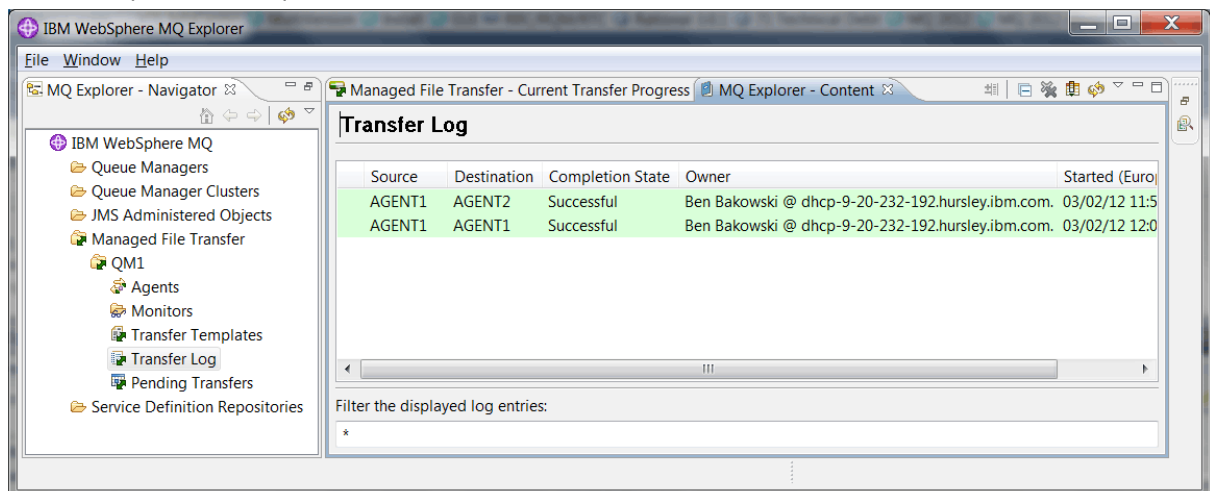
Directory: C:\Users\Ben Bakowski\Desktop\FTETEST\END

File name: myDemoTransfer.txt

Overwrite files on the destination file system that have the same name

8. Klicken Sie auf **Jetzt fertigstellen**, damit die Übertragung beginnt.

9. Sie können den Übertragungsfortschritt auf der Registerkarte **Aktueller Übertragungsfortschritt** in IBM WebSphere MQ Explorer überwachen:



10. Sie können das Dateisystem auch manuell überprüfen, um sich zu vergewissern, dass die neue Datei vorhanden ist, beispielsweise C : \end\myfile .txt.

### ***Datei über die Befehlszeile übertragen***

Die Flexibilität von Dateiübertragungen lässt sich sehr gut anhand der Initiierung einer Übertragung über die Befehlszeile veranschaulichen. Im vorliegenden Szenario wird dies zwar nicht abgedeckt, Sie können aber mithilfe dieser Prinzipien und der Ant-Scriptingtechnologie wesentlich leistungsstärkere Dateiübertragungsszenarios definieren und implementieren.

### **Vorgehensweise**

1. Löschen Sie die übertragene Datei aus der vorherigen Demonstration, beispielsweise C : \end\myfile .txt.

2. Leiten Sie mit dem Befehl **fteCreateTransfer** die Übertragung Ihrer Datei von C:\start\myfile.txt an C:\end\myfile.txt ein:

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT2 -dm hubQM -w -dd C:\end\ C:\start\myfile.txt
```

- -sa AGENT1 definiert den Quellenagenten (also den Agenten, von dem die Datei übertragen wird) als AGENT1.
- -sm hubQM definiert den Warteschlangenmanager, mit dem sich der Quellenagent AGENT1 verbindet.
- -da AGENT2 definiert den Zielagenten (also den Agenten, an den die Datei übertragen wird) als AGENT2.
- -w legt fest, dass der Befehl **fteCreateTransfer** auf die Bestätigung seiner erfolgreichen Ausführung wartet.
- -dd C:\end\ definiert das Zielverzeichnis als C:\end\.
- C:\start\myfile.txt definiert die zu übertragende Datei.

Sie finden weitere Informationen unter [fteCreateTransfer \(Erstellung einer neuen Dateiübertragung\)](#).

3. Vergewissern Sie sich, dass die Dateiübertragung erfolgreich verlaufen ist, indem Sie das Dateiübertragungsprotokoll in IBM WebSphere MQ Explorer überprüfen oder das Dateisystem manuell untersuchen.

## Nächste Schritte

Möglicherweise möchten Sie weitere Funktionen der Dateiübertragungsfunktionalität über externe Medien erkunden. Unter

- [„Dateiübertragung zwischen zwei Computern mithilfe der Scripts“](#) auf Seite 173
- Höherer Schutz Ihrer Umgebung. Ihre eigenen Sicherheitsanforderungen erfordern möglicherweise ein anderes Zugriffsmodell als das im vorliegenden Szenario verwendete Modell. Sie finden weitere Informationen zu Best Practices in diesem Bereich unter [https://www.ibm.com/developerworks/web-sphere/library/techarticles/0902\\_wyatt/0902\\_wyatt.html](https://www.ibm.com/developerworks/web-sphere/library/techarticles/0902_wyatt/0902_wyatt.html).

## Dateiübertragung zwischen zwei Computern mithilfe der Scripts

Erweitert das Basisszenario der Dateiübertragung mit Scripts zu einer IBM WebSphere MQ-Nachrichtenübertragungstopologie mit mehreren Computern.

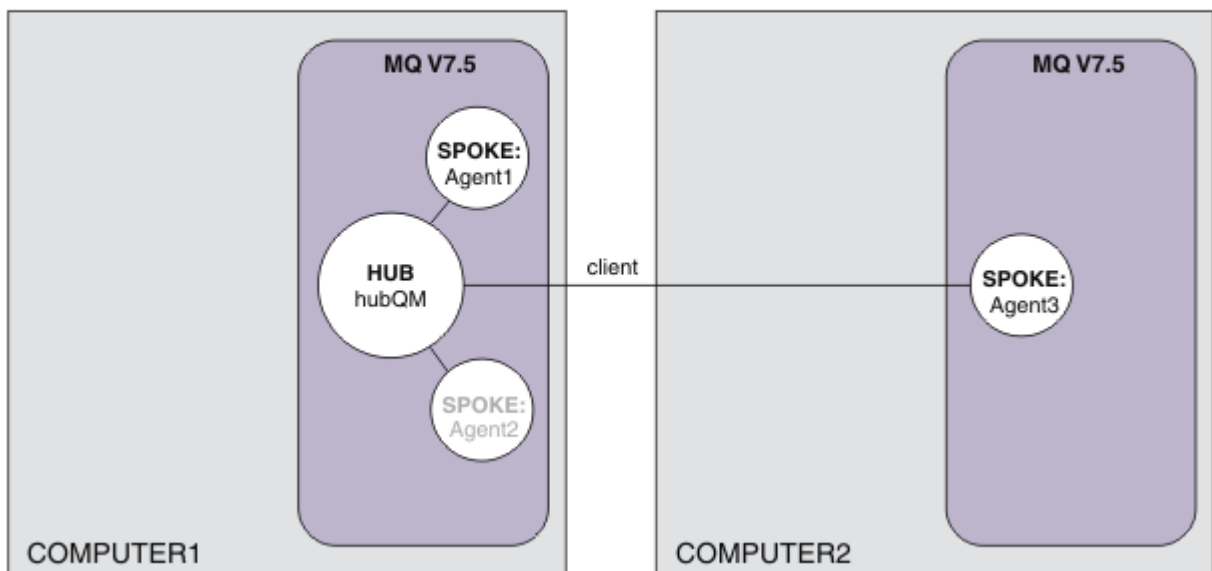
Nach der einfachen Demonstration der Funktion 'Managed File Transfer' für eine verwaltete Dateiübertragung (siehe [„Basisdateiübertragung mithilfe der Scripts“](#) auf Seite 159) sind Sie nun mit den grundlegenden Prinzipien von verwalteten Dateiübertragungen vertraut. Da Sie erkannt haben, dass diese Funktion auf einem einzelnen Computer nur von geringem Nutzen ist, erweitern Sie das Szenario nun, um zu erkunden, wie Dateiübertragungen in eine IBM WebSphere MQ-Nachrichtenübertragungstopologie mit mehreren Computern integriert werden können.

Beginnen Sie also, die Vorteile zu nutzen, die durch die zu Grunde liegende IBM WebSphere MQ-Technologie ermöglicht werden: eine zuverlässige und einmalige Zustellung von Dateien. Um dies zu erreichen, wird die Topologie um einen zweiten Computer erweitert, der an Dateiübertragungen beteiligt ist. Im vorliegenden Szenario installieren und konfigurieren Sie den separat erhältlichen Managed File Transfer Agent und erhalten einen Einblick in die Sicherheitsaspekte, die bei einer Dateiübertragungstopologie mit mehreren Computern wichtig sind. Am Ende dieses Szenarios wird eine Dateiübertragung von einem Computer zu einem anderen veranschaulicht. Auf diesem Vorgang basiert das nächste Szenario, in dem Sie Prüffunktionen hinzufügen können, durch die deutlich wird, weshalb es sich bei dieser Funktion um eine verwaltete Dateiübertragung handelt.

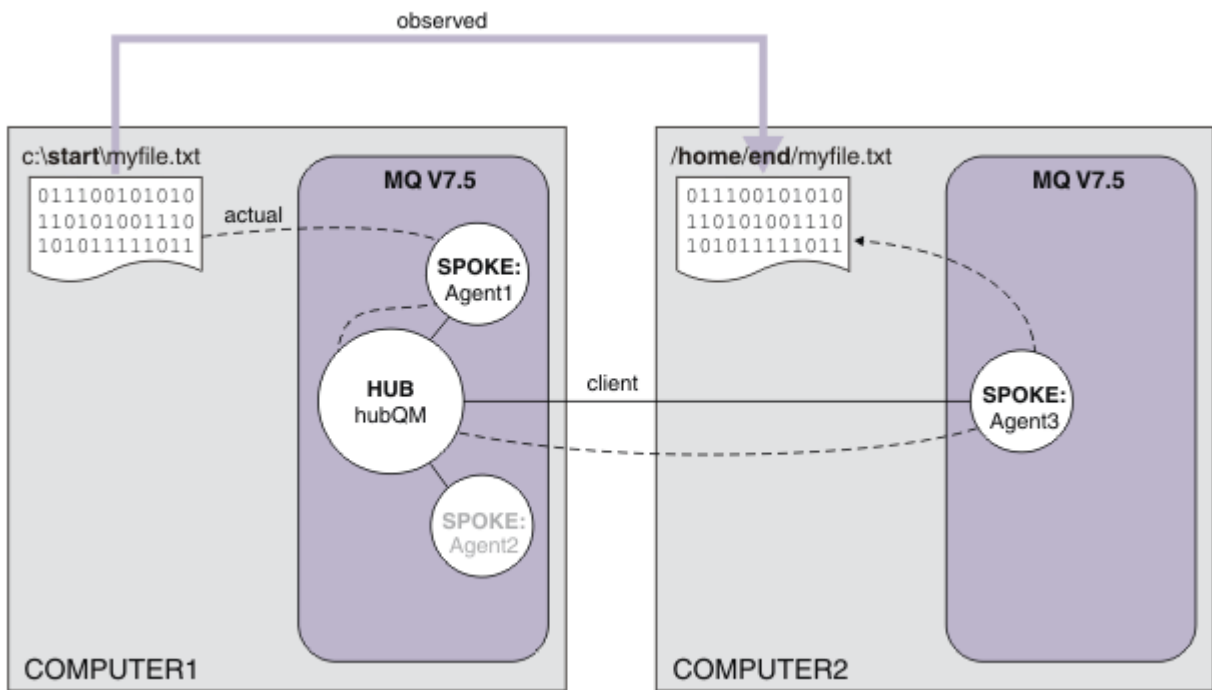
Es wird vorausgesetzt, dass Sie die Tasks im Abschnitt [„Basisdateiübertragung mithilfe der Scripts“](#) auf Seite 159 ausgeführt haben.

## Übersicht

In diesem Szenario wird die Hub-und Peripherietopologie, die im Szenario „Basisdateiübertragung mithilfe der Scripts“ auf Seite 159 konfiguriert ist, um einen zweiten Computer, Computer 2, erweitert. Auf Computer 2 sind der Managed File Transfer Agent und die Befehlszeilentools installiert. Auf diesem zweiten Computer ist keine lokale IBM WebSphere MQ -Serverinstallation erforderlich. Ein derartiges Modell wird vielfach in Hub-and-Spoke-Lösungen eingesetzt, bei denen mehrere Managed File Transfer Agents über Clientverbindungen mit einem zentralen IBM WebSphere MQ-Warteschlangenmanager interagieren, ohne dass in jedem Peripheriesystem IBM WebSphere MQ-Serverinstallationen und zugehörige Lizenzen erforderlich sind. Sonstige Topologien werden ebenfalls unterstützt; Ihre eigene Topologie hängt von Ihren speziellen Anforderungen und der Lizenzierung oder Berechtigung ab. AGENT2 wird zu Gunsten von AGENT2 gesperrt, damit dieser Übertragungen auf diesem Computer alleine verarbeitet.



Dieses Diagramm zeigt eine Hub-and-Spoke-Topologie mit zwei Computern. AGENT1 und AGENT2 werden für die Verwendung von Bindungen zum (lokalen) Hub konfiguriert, während AGENT3 sich über eine Clientverbindung verbindet. AGENT2 ist inaktiviert, da er keine weitere Rolle im Szenario spielt. AGENT1 verarbeitet alle Dateiübertragungsaktivitäten auf Computer 1. Stellen Sie sicher, dass Sie das Sicherheitsmodell und seine Einschränkungen verstehen. Details finden Sie im Szenario „Basisdateiübertragung mithilfe der Scripts“ auf Seite 159. Nach ihrer Konfiguration wird diese Topologie für die Übertragung einer Datei von Computer 1 an Computer 2 verwendet.



Dieses Diagramm zeigt die Dateiübertragungsrouten, die veranschaulicht wird. Auch in diesem Fall findet die zu Grunde liegende Dateiübertragung über zuverlässige und aktive IBM WebSphere MQ-Verbindungen statt. In diesem Beispiel wird angenommen, dass Computer 1 ein Windows-Computer und Computer 2 ein Linux -Computer ist. Sie können auch andere Plattformen und Architekturen nutzen. Sie finden eine vollständige Liste der unterstützten Plattformen unter <https://www.ibm.com/support/docview.wss?uid=swg27006467#7.1>.

## Lösung planen

Beschreibt die Übertragung von Dateien an einen Computer oder von einem Computer, Dateiübertragungsagenten (File Transfer Agents), Voraussetzungen, Lizenzen, Vorbereitung des Computers und Konfiguration von IBM WebSphere MQ für das Dateiübertragungsszenario mit zwei Computern unter Verwendung von Scripts.

## Voraussetzungen

Sie benötigen die folgenden Elemente:

- Computer 1, eine funktionierende Konfiguration aus „Basisdateiübertragung mithilfe der Scripts“ auf Seite 159.
- Computer 2, ein zweiter Testcomputer, der die Hardware- und Betriebssystemvoraussetzungen für IBM WebSphere MQ Version 7.5 erfüllt. Weitere Informationen finden Sie im Abschnitt <https://www.ibm.com/support/docview.wss?uid=swg27006467>.
- In: IBM WebSphere MQ Version 7.5. Sie können eine Testversion von <https://www.ibm.com/developerworks/downloads/ws/wmq/herunterladen>.
- Kenntnis der IP-Adressen von Computer 1 und Computer 2 (Format nnn . nnn . nnn . nnn).

## Installieren Sie IBM WebSphere MQ Version 7.5.

Installieren Sie den IBM WebSphere MQ Version 7.5-Server mit folgenden Komponenten: Server, IBM WebSphere MQ Explorer, IBM WebSphere MQ Advanced Managed File Transfer Agent und IBM WebSphere MQ Managed File Transfer Command Line Tools. Ausführliche Informationen finden Sie im Abschnitt [Zu installierende Komponenten auswählen](#).

Überlegen Sie sich, wie Sie IBM WebSphere MQ verwalten möchten. Zur Verwaltung von IBM WebSphere MQ haben Sie folgende Möglichkeiten:

- Einrichtung einer entsprechenden Umgebung mit dem Befehl **setmqenv**. Weitere Informationen finden Sie im Abschnitt [setmqenv](#).
- Aufruf vollständig qualifizierter IBM WebSphere MQ-Verwaltungsbefehle.

## Benutzer und Gruppen vorbereiten

Bei diesem Sicherheitsmodell werden die folgenden Gruppen und Benutzer vorausgesetzt:

Gruppen

- mqm

Wird automatisch bei der IBM WebSphere MQ-Installation erstellt. Mitglieder dieser Gruppe können IBM WebSphere MQ und die zugehörigen Ressourcen verwalten.

- FTUSERS

Diese Gruppe müssen Sie erstellen. Mitglieder dieser Gruppe können Dateiübertragungen einleiten.

- FTAGENTS

Diese Gruppe müssen Sie erstellen. Mitglieder dieser Gruppe können Übertragungsagenten starten und stoppen. Diese dienen als Endpunkte, welche die Dateiübertragungen in Ihrem Netz verarbeiten.

Benutzer

- mqmAdmin

IBM WebSphere MQ-Administrator, also ein Mitglied der Gruppe 'mqm' bzw. unter Windows ein Mitglied der Administratorgruppe.

- ftuser

Sie müssen diesen Benutzer erstellen und als Mitglied zur Gruppe FTUSERS hinzufügen. Nehmen Sie diesen Benutzer nicht in die Gruppe 'mqm' auf, um das Risiko von unerwünschten Angriffen auf den Warteschlangenmanager bezüglich der Verwaltungssicherheit zu vermeiden.

- ftagent

Sie müssen diesen Benutzer erstellen und als Mitglied zur Gruppe FTAGENTS hinzufügen. Nehmen Sie diesen Benutzer nicht in die Gruppe 'mqm' auf, um das Risiko von unerwünschten Angriffen auf den Warteschlangenmanager bezüglich der Verwaltungssicherheit zu vermeiden.

## Neuen Agenten AGENT3 als Spoke auf Computer 2 hinzufügen

Bereiten Sie Computer 2 auf die Unterstützung der erweiterten Topologie für das IBM WebSphere MQ-Da-  
teübertragungsszenario mit zwei Computern unter Verwendung von Scripts vor.

### Informationen zu diesem Vorgang

Verwenden Sie das Beispielscript addSpoke auf Computer 1 erneut und definieren Sie einen Agentenspo-  
ke, der über eine Clientverbindung angeschlossen wird. Bei Abschluss des Befehls **addSpoke** wird eine  
Gruppe mit Befehlen angezeigt, die über die IBM WebSphere MQ-Installation auf Computer 2 ausgeführt  
werden.

### Vorgehensweise

1. Führen Sie auf Computer 1 als Benutzer mqmAdmin den Befehl **addspoke** im Verzeichnis IBM Web-  
Sphere MQ bin <MQ\_INSTALL\_ROOT>\binaus.

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT3  
hubQmgr=hubQM connectionMode=CLIENT agentIPAddress=<IP address of computer 2>  
hubIPAddress=<IP address of computer 1> hubPort=1414
```



2. Führen Sie auf Computer 2 als Benutzer mqmAdmin die Befehle aus, die durch den Befehl **addSpoke** auf Computer 1 ausgegeben wurden. Beispiel:
  - a) `fteSetupCoordination -coordinationQMgr hubQM-coordinationQMgrHost <IP address of computer 1> -coordinationQMgrPort 1414 -coordinationQMgrChannel FTE.USER.SVRCONN -f`
  - b) `fteSetupCommands -p hubQM -connectionQMgr hubQM -connectionQMgrHost <IP address of computer 1> -connectionQMgrPort 1414 -connectionQMgrChannel FTE.USER.SVRCONN -f`
  - c) `fteCreateAgent -p hubQM -agentName AGENT3 -agentQMgr hubQM -agentQMgrHost <IP address of computer 1> -agentQMgrPort 1414 -agentQMgrChannel FTE.AGENT.SVRCONN -f`

Beachten Sie, dass Sie die generierten MQSC-Scripts nicht ausführen müssen.

3. Starten Sie auf Computer 2 den Agenten AGENT3 als Benutzer ftagent.

```
fteStartAgent -p hubQM AGENT3
```

4. Prüfen Sie auf Computer 2 als Benutzer ftagent, ob die drei Agenten verfügbar sind (starten Sie gegebenenfalls die Agenten auf Computer 1 unter dem Benutzer ftagent).

```
ftelistAgents -p hubQM
```

Sie sehen die folgende Ausgabe:

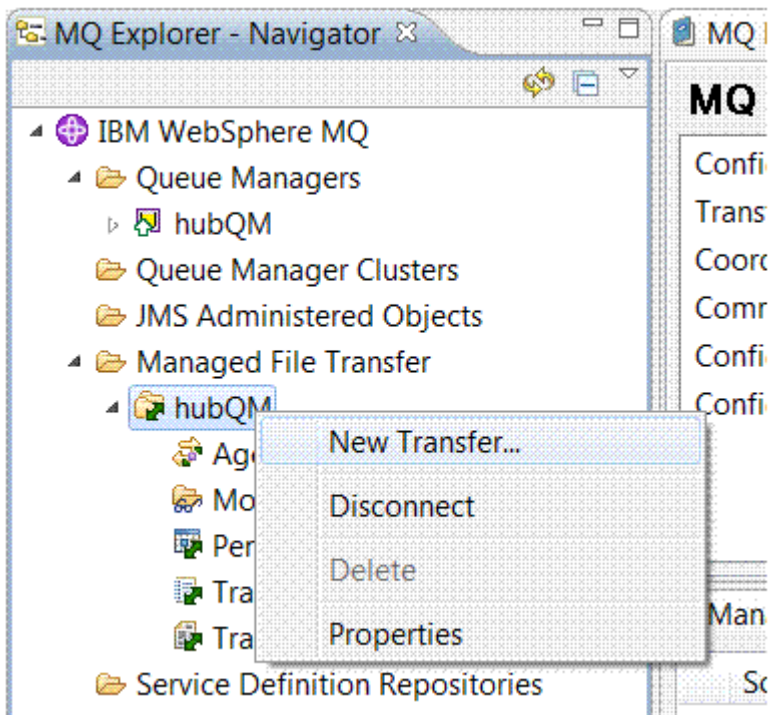
Agent Name:	Queue Manager Name:	Status:
AGENT1	hubQM	READY
AGENT2	hubQM	READY
AGENT2	hubQM	READY

## Lösung implementieren

Starten Sie die Demonstration für das Dateiübertragungsszenario mit zwei Computern unter Verwendung der Scripts, indem Sie eine Datei mit IBM WebSphere MQ übertragen. Überwachen Sie den Übertragungsfortschritt und überprüfen Sie seinen Erfolg durch die Untersuchung des Dateisystems.

### Vorgehensweise

1. Geben Sie die Quelldatei und das Zielverzeichnis an:
  - a) Erstellen Sie auf Computer 1 eine zu übertragende Beispieldatei, beispielsweise `C:\start\myfile.txt`.
  - b) Geben Sie auf Computer 2 ein vorhandenes Zielverzeichnis für die Übertragung dieser Datei an, beispielsweise `C:\end\`. Stellen Sie sicher, dass der Benutzer ftagent über Schreibzugriff für das Verzeichnis verfügt.
2. Starten Sie IBM WebSphere MQ Explorer auf Computer 1 als Benutzer mqmAdmin. Beachten Sie, dass Sie wie auch bei dem vorherigen Szenario als IBM WebSphere MQ-Administrator handeln und das Szenario auf Dateiübertragungen konzentrieren müssen, nicht auf die Konfiguration von IBM WebSphere MQ Explorer. Starten Sie das Programm über das Menü "Start" (bzw. über eine funktionale Entsprechung) oder führen Sie den Befehl **MQExplorer** aus. Sie finden ausführliche Informationen hierzu unter [WebSphere MQ Explorer starten](#).
3. Klicken Sie im Abschnitt **Verwaltete Dateiübertragung** im IBM WebSphere MQ -Navigator mit der rechten Maustaste auf hubQM und wählen Sie **Neue Übertragung** aus, damit der Assistent für neue Übertragung gestartet wird.



4. Wählen Sie über das Menü AGENT1 als Quellenagenten und AGENT3 als Zielagenten aus:

Source agent  
 Name: AGENT1  
 Type: Standard

---

Destination agent  
 Name: AGENT3  
 Type: Standard

5. Klicken Sie auf **Weiter** und anschließend auf **Hinzufügen ...**. Geben Sie als Quelle den Pfad zu der Datei ein, die Sie übertragen möchten, z. B. C:\start\myfile.txt. Stellen Sie sicher, dass das Kontrollkästchen **Nach erfolgreicher Übertragung Quelldatei löschen** aktiviert ist.

Source  
 Agent: AGENT1  
 Type: Standard agent  
 Type: File

File name: C:\start\myFile.txt Browse...

Remove source file if the transfer is successful

6. Geben Sie das Zielverzeichnis ein. Da sich in diesem Szenario das Ziel auf einer Linux-Plattform befindet, müssen Sie die entsprechende Schreibweise /home/end/ verwenden.

Destination

Agent: AGENT3

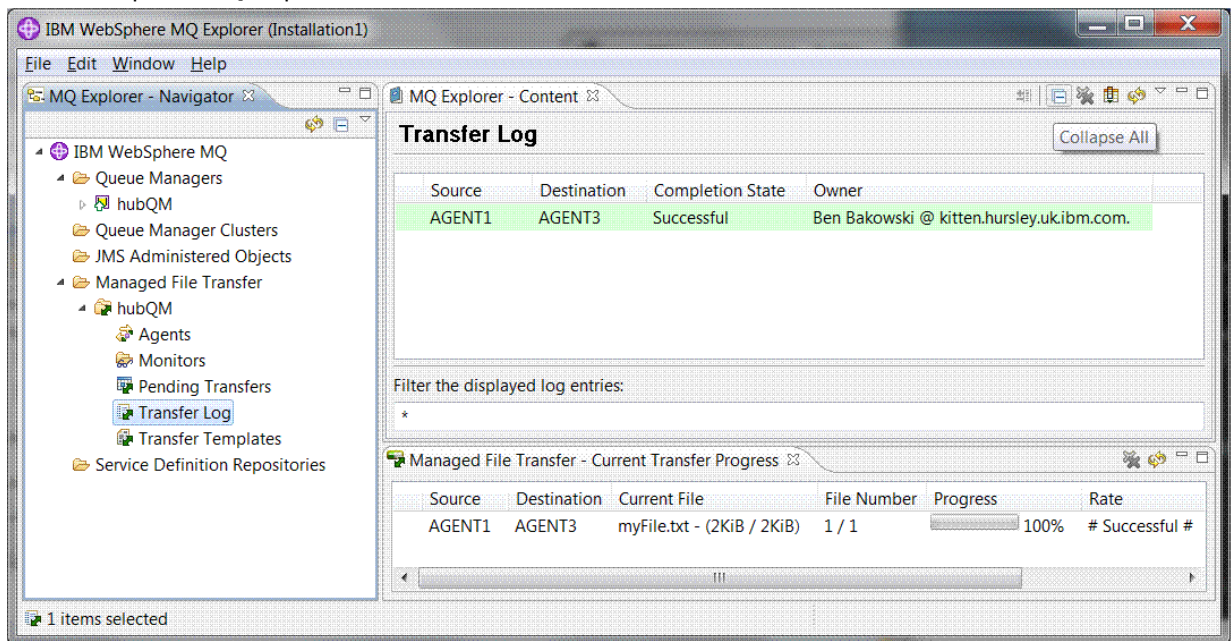
Type: Standard agent

Type:

Directory:

Overwrite files if present

- Klicken Sie auf **OK** und dann auf **Fertigstellen**, damit die Übertragung beginnt.
- Sie können den Übertragungsfortschritt auf der Registerkarte **Aktueller Übertragungsfortschritt** in IBM WebSphere MQ Explorer überwachen.



- Sie können das Dateisystem auch manuell überprüfen, um sich zu vergewissern, dass die neue Datei vorhanden ist, beispielsweise `/home/end/myfile.txt`.

### **Datei über die Befehlszeile übertragen**

Die Flexibilität von Dateiübertragungen lässt sich sehr gut anhand der Initiierung einer Übertragung über die Befehlszeile veranschaulichen. Im vorliegenden Szenario wird dies zwar nicht abgedeckt, Sie können aber mithilfe dieser Prinzipien und der Ant-Scriptingtechnologie wesentlich leistungsstärkere Dateiübertragungsszenarios definieren und implementieren.

### **Vorgehensweise**

- Löschen Sie die übertragene Datei aus der vorherigen Demonstration, beispielsweise `/home/end/myfile.txt`, und stellen Sie sie wieder in ihrem ursprünglichen Verzeichnis her.
- Verwenden Sie zur Verwendung von `ftuser` den Befehl **fteCreateTransfer** aus, um die Übertragung Ihrer Datei von `C:\start\myfile.txt` auf Computer 1 auf `/home/end/myfile.txt` auf Computer 2 einzuleiten:

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT3 -dm hubQM -w -dd "/home/end/" "C:\start\myfile.txt"
```

3. Vergewissern Sie sich, dass die Dateiübertragung erfolgreich verlaufen ist, indem Sie das Dateiübertragungsprotokoll in IBM WebSphere MQ Explorer überprüfen oder das Dateisystem manuell untersuchen.

## Nächste Schritte

Weitere Informationen zur IBM WebSphere MQ -Sicherheit finden Sie unter [Sicherheit](#).

Möglicherweise möchten Sie weitere Funktionen der Dateiübertragungsfunktionalität über externe Medien erkunden.

- Sie finden ausführliche Informationen zum Hinzufügen der Prüffunktion für die Bereitstellung des verwalteten Aspekts der verwalteten Dateiübertragung unter [„Prüffunktion zur verwalteten Dateiübertragung hinzufügen“](#) auf Seite 188.
- Weitere Informationen zu:
  - Triggering: Verschieben neuer Dateien bei deren Anzeige.
  - Triggering: Konfiguration der Anzeige einer einzelnen Datei für die Einleitung einer Übertragung mehrerer Dateien.
  - Scripterstellung für Übertragungen mit Apache Ant.

Weitere Informationen finden Sie im Abschnitt [https://www.ibm.com/developerworks/websphere/library/techarticles/1003\\_phillips/1003\\_phillips.html](https://www.ibm.com/developerworks/websphere/library/techarticles/1003_phillips/1003_phillips.html).

- Ihre eigenen Sicherheitsanforderungen erfordern möglicherweise ein anderes Zugriffsmodell als das im vorliegenden Szenario verwendete Modell. Sie finden weitere Informationen zu Best Practices für einen höheren Schutz Ihrer Umgebung unter [https://www.ibm.com/developerworks/websphere/library/techarticles/0902\\_wyatt/0902\\_wyatt.html](https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html).

## Dateiübertragung zwischen zwei Computern im Detail

Erweitert das Basisszenario der Dateiübertragung zu einer IBM WebSphere MQ-Nachrichtenübertragungstopologie mit mehreren Computern.

Nach der einfachen Demonstration der Funktion 'Managed File Transfer' für eine verwaltete Dateiübertragung (siehe [„Basisdateiübertragung im Detail“](#) auf Seite 167) sind Sie nun mit den grundlegenden Prinzipien von verwalteten Dateiübertragungen vertraut. Da Sie erkannt haben, dass diese Funktion auf einem einzelnen Computer nur von geringem Nutzen ist, erweitern Sie das Szenario nun, um zu erkunden, wie Dateiübertragungen in eine IBM WebSphere MQ-Nachrichtenübertragungstopologie mit mehreren Computern integriert werden können.

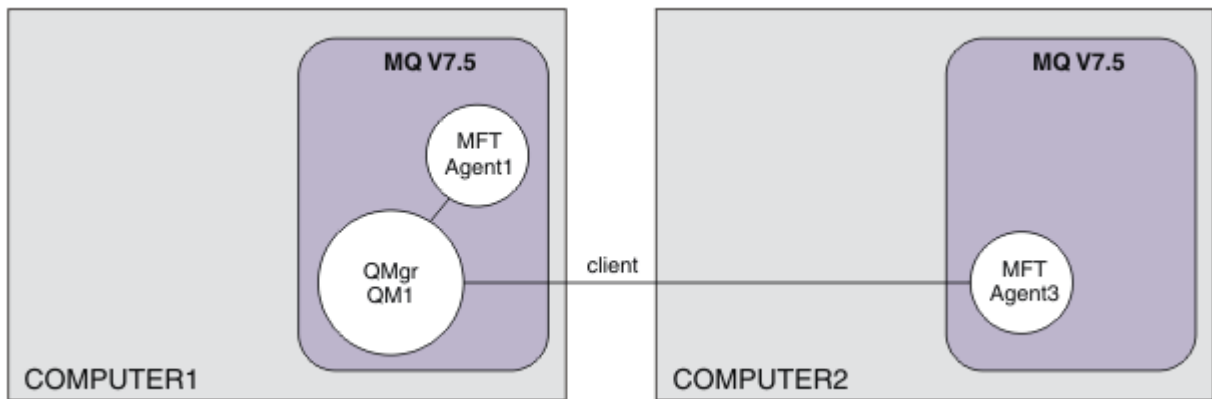
Beginnen Sie also, die Vorteile zu nutzen, die durch die zu Grunde liegende IBM WebSphere MQ-Technologie ermöglicht werden: eine zuverlässige und einmalige Zustellung von Dateien. Um dies zu erreichen, wird die Topologie um einen zweiten Computer erweitert, der an Dateiübertragungen beteiligt ist. Im vorliegenden Szenario installieren und konfigurieren Sie den separat erhältlichen Managed File Transfer Agent und erhalten einen Einblick in die Sicherheitsaspekte, die bei einer Dateiübertragungstopologie mit mehreren Computern wichtig sind. Am Ende dieses Szenarios wird eine Dateiübertragung von einem Computer zu einem anderen veranschaulicht. Auf diesem Vorgang basiert das nächste Szenario, in dem Sie Prüffunktionen hinzufügen können, durch die deutlich wird, weshalb es sich bei dieser Funktion um eine verwaltete Dateiübertragung handelt.

Es wird vorausgesetzt, dass Sie die Tasks im Abschnitt [„Basisdateiübertragung im Detail“](#) auf Seite 167 ausgeführt haben.

## Übersicht

In diesem Szenario fahren Sie mit dem vorhandenen Windows-Computer fort, der im Basisszenario der Dateiübertragung konfiguriert wurde. Agent2 ist inaktiviert, da Sie den einzelnen Agenten als Endpunkt der Dateiübertragung auf diesem ersten Computer verwenden. Auf einem zweiten Computer installieren Sie den Managed File Transfer Agent; eine lokale IBM WebSphere MQ-Serverinstallation ist auf diesem zweiten Computer keine Vorbedingung. Ein derartiges Modell wird vielfach in Hub-and-Spoke-Lösungen

eingesetzt, bei denen mehrere Managed File Transfer Agents über Clientverbindungen mit einem zentralen IBM WebSphere MQ-Warteschlangenmanager interagieren, ohne dass in jedem Peripheriesystem IBM WebSphere MQ-Serverinstallationen und zugehörige Lizenzen erforderlich sind. Sonstige Topologien werden ebenfalls unterstützt; Ihre eigene Topologie hängt von Ihren speziellen Anforderungen und der Lizenzierung oder Berechtigung ab.

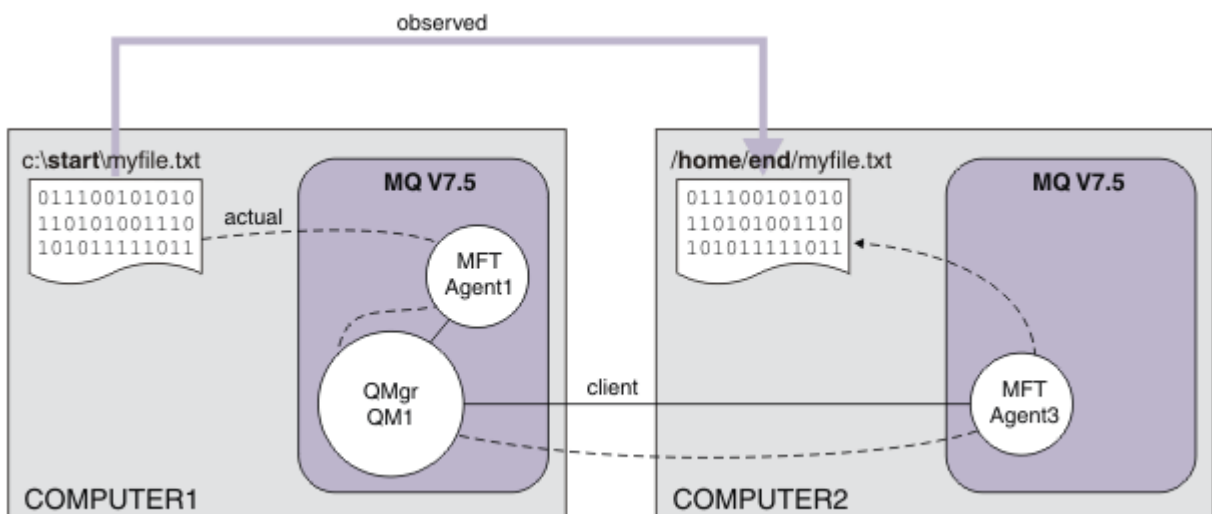


In einem Dateiübertragungsnetz wird ein einzelner Warteschlangenmanager ausgewählt, der als zentrale Stelle im Netz für die Übertragung von Prüf- und Dateiübertragungsinformationen agiert. Im vorherigen Szenario „Basisdateiübertragung im Detail“ auf Seite 167 wurde ein einzelner Warteschlangenmanager implizit als Koordinations-Warteschlangenmanager verwendet. In diesem Szenario verwenden Sie weiterhin den Warteschlangenmanager QM1 auf Computer 1 als Koordinations-Warteschlangenmanager und ein Teil der Konfiguration von Computer 2 legt die Umgebung so fest, dass dieser Warteschlangenmanager verwendet wird.

Da es in dieser Topologie nicht erforderlich ist, dass Dateiübertragungen von Computer 2 aus eingeleitet werden können, müssen Sie die optionale Komponente 'Managed File Transfer Command Line Tools' auf diesem zweiten Computer nicht installieren.

Sie müssen sich der Tatsache bewusst sein, dass zwar einige grundlegende Sicherheitsfunktionen eingesetzt werden, die Sicherheit der Dateiübertragungstopologie jedoch möglicherweise nicht Ihrem Standard entspricht. Eine Beschreibung zum Schutz von IBM WebSphere MQ und insbesondere von Dateiübertragungen finden Sie unter [https://www.ibm.com/developerworks/websphere/library/techarticles/0902\\_wyatt/0902\\_wyatt.html](https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html).

Bei der Installation auf dem zweiten Computer erstellen Sie den eigentlichen Agenten Agent3 und veranschaulichen dann die Lösung, indem Sie eine Datei von Computer 1 auf Computer 2 verschieben.



Die zugrunde liegende Dateiübertragung erfolgt über zuverlässige und leistungsfähige IBM WebSphere MQ -Verbindungen. Die nachfolgenden Tasks demonstrieren die Konfiguration und ordnungsgemäße Sicherung dieser Verbindungen.

In diesem Beispiel wird angenommen, dass Computer 1 ein Windows -Computer und Computer 2 ein Linux -Computer ist. Sie können auch andere Plattformen und Architekturen nutzen. Sie finden eine vollständige Liste der unterstützten Plattformen unter <https://www.ibm.com/support/docview.wss?uid=swg27006467#7.1>.

## Lösung planen

Hier finden Sie Informationen zur Planung der Dateiübertragungslösung mit zwei Computern, einschließlich einer Beschreibung der entsprechenden Infrastruktur und der Gruppen und Benutzer, die erstellt werden müssen.

## Vorbereitende Schritte

Sie benötigen die folgenden Elemente:

- Eine funktionierende Konfiguration aus dem Basisszenario der Dateiübertragung. Ausführliche Informationen hierzu finden Sie unter „Basisdateiübertragung im Detail“ auf Seite 167.
- In: IBM WebSphere MQ Version 7.5. Sie können eine Testversion von <https://www.ibm.com/developerworks/downloads/ws/wmq/herunterladen>.
- Ein zweiter Testcomputer, der die Hardware- und Betriebssystemvoraussetzungen für IBM WebSphere MQ Version 7.5 erfüllt. Ausführliche Informationen hierzu finden Sie unter <https://www.ibm.com/support/docview.wss?uid=swg27006467>.

In diesem Szenario interagiert der zweite Computer über einen Clientverbindungskanal mit dem Warteschlangenmanager. Durch die Verwendung von Kanalauthentifizierungsdatensätzen wird Folgendes sichergestellt:

- Die eingehende Verbindung wird als vom neuen Computer stammend authentifiziert, die einen weiteren verwalteten Managed File Transfer Agent bereitstellt.
- Die eingehende Anforderung wird einem Benutzer zugeordnet, der über den erforderlichen Zugriff auf die Ressourcen einer verwalteten Dateiübertragung verfügt.

Das Sandboxing wird nicht berücksichtigt. Dies ist ein Verfahren, bei dem die Bearbeitung bestimmter Dateisystembereiche (beispielsweise IBM WebSphere MQ-Konfigurationsdateien) durch Dateiübertragungen eingeschränkt wird.

## Vorgehensweise

Auf **beiden** Computern:

1. Erstellen Sie die Gruppen FTEUSERS und FTEAGENTS. Sie finden ausführliche Informationen hierzu in den Anweisungen zu Ihrem Betriebssystem.
2. Erstellen Sie den Benutzer `fteuser1` und fügen Sie ihn der Gruppe FTEAGENTS hinzu.
3. Erstellen Sie den Benutzer `fteagent` und fügen Sie ihn der Gruppe FTEUSERS hinzu.

Der Benutzer `fteuser1` leitet Dateiübertragungen ein, während der Benutzer `fteagent` Agentenprozesse startet und stoppt. Da diese Benutzer nicht der Gruppe 'mqm' (oder der Gruppe der Windows-Administratoren) angehören, bedeutet dies einen gewissen Schutz der Topologie vor unerwünschten Administrationsangriffen. Durch die Definition bestimmter Benutzer auf Agentenbasis (beispielsweise `fteagent1`, `fteagent2`) können Sie einen differenzierten Sicherheitszugriff bereitstellen.

## Computer 1 für die Unterstützung der erweiterten Topologie ändern

Bereiten Sie Computer 1 auf die Unterstützung der erweiterten Topologie für das IBM WebSphere MQ-Dateiübertragungsszenario mit zwei Computern vor.

## Informationen zu diesem Vorgang

In dieser Task löschen Sie Agent2, da er nicht mehr benötigt wird. Anschließend erstellen und starten Sie ein Empfangsprogramm für die Annahme einer Clientverbindung von dem Agenten, der auf Computer 2 erstellt wurde, und erstellen so ein einfaches IBM WebSphere MQ-Netz, das die beiden Computer umfasst. Dabei wird vorausgesetzt, dass auf Computer 1 das Betriebssystem Windows ausgeführt wird. Falls Sie bei der Ausführung dieses Szenarios eine andere Plattform verwenden, müssen Sie die entsprechenden plattformspezifischen Befehle einsetzen.

## Vorgehensweise

1. Stoppen Sie den Agenten Agent2.

```
fteStopAgent AGENT2
```

Sie finden ausführliche Informationen zum Befehl **fteStopAgent** unter [fteStopAgent \(Stoppen eines WebSphere MQ Managed File Transfer-Agenten\)](#).

2. Löschen Sie den Agenten Agent2.

```
fteDeleteAgent AGENT2  
runmqsc QM1 < <output>
```

Sie finden ausführliche Informationen zum Befehl **fteDeleteAgent** unter [fteDeleteAgent \(Löschen eines WebSphere MQ Managed File Transfer-Agenten\)](#).

Konfigurieren Sie die IBM WebSphere MQ-Sicherheit so, dass ein neuer Dateiübertragungsagent, der auf Computer 2 konfiguriert wurde, mit dem Koordinations-Warteschlangenmanager QM1 interagieren kann. Dieser neue Agent verbindet sich über den vorhandenen Kanal SYSTEM.DEF.SVRCONN mit QM1. Ihre eigene Sicherheitsstruktur kann davon abweichen. Unter [Weitere Schritte](#) finden Sie weitere Details zur Abschottung dieser Topologie.

3. Starten Sie die MQSC-Schnittstelle für QM1.

```
runmqsc QM1
```

4. Erstellen Sie zwei Kanäle für die Verarbeitung eingehender Anforderungen, die von Benutzern und Agenten stammen.

```
DEFINE CHANNEL(FTE.USER.SVRCONN) CHLTYPE(SVRCONN)  
DEFINE CHANNEL(FTE.AGENT.SVRCONN) CHLTYPE(SVRCONN)
```

5. Erstellen Sie einen Kanalauthentifizierungsdatensatz, damit eine Verbindung von Computer 2 zu QM1 möglich ist; ordnen Sie dabei den erstellten Benutzer zu.

```
SET CHLAUTH('FTE.USER.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('<IP address of computer2>')  
USERSRC (MAP) MCAUSER('fteuser' DESCR('Rule to allow file transfer users to communicate'))  
ACTION(ADD)  
  
SET CHLAUTH('FTE.AGENT.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('<IP address of computer2>')  
USERSRC(MAP) MCAUSER('fteagent') DESCR('Rule to allow file transfer agent processes to communicate') ACTION(ADD)
```

Weitere Informationen finden Sie im Abschnitt [Kanalauthentifizierungsdatensätze](#).

Dieses Szenario zielt nicht darauf ab, die Topologie zu sperren und abzuschotten, sondern das Ziel besteht in der Veranschaulichung einer Basisdateiübertragung. Das implementierte Sicherheitsmodell reicht für diese Demonstration zwar aus, Sie müssen sich jedoch Ihrer eigenen Sicherheitsbedrohungen bewusst sein und gegebenenfalls geeignete Maßnahmen ergreifen. Unter [Weitere Schritte](#) werden mögliche Optionen erläutert.

6. Ermitteln Sie einen freien Port, der für die Netzkommunikation mit IBM WebSphere MQ verwendet werden kann. Definieren Sie das Empfangsprogramm LISTENER1 für die Verwendung dieses freien Ports (beispielsweise 1414).

```
DEFINE LISTENER(LISTENER1) TRPTYPE(TCP) CONTROL(QMGR) PORT(1414)
```

7. Starten Sie das Empfangsprogramm LISTENER1.

```
START LISTENER(LISTENER1)
```

8. Stoppen Sie die MQSC-Schnittstelle für QM1.

```
end
```

9. Überprüfen Sie, ob die Gruppen FTEAGENTS und FTEUSERS über den entsprechenden Zugriff auf IBM WebSphere MQ -Objekte verfügen, um Dateiübertragungsaktionen für einen Agenten AGENT3 auszuführen, der auf Computer 2 erstellt werden soll. Sie können diese Konfiguration an Ihre eigenen Sicherheitsanforderungen anpassen.

- a. `setmqaut -m QM1 -t qmgr -g FTEAGENTS +connect +inq`
- b. `setmqaut -m QM1 -t qmgr -g FTEUSERS +connect`
- c. `setmqaut -m QM1 -n "SYSTEM.FTE" -t q -g FTEAGENTS +get +put`
- d. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT1" -t q -g FTEUSERS +put`
- e. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT1" -t q -g FTEAGENTS +setid +get +put`
- f. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT3" -t q -g FTEUSERS +put`
- g. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT3" -t q -g FTEAGENTS +setid +get +put`
- h. `setmqaut -m QM1 -n "SYSTEM.FTE.DATA.AGENT1" -t q -g FTEAGENTS +get +put`
- i. `setmqaut -m QM1 -n "SYSTEM.FTE.DATA.AGENT3" -t q -g FTEAGENTS +get +put`
- j. `setmqaut -m QM1 -n "SYSTEM.FTE.EVENT.AGENT1" -t q -g FTEAGENTS +get +put`
- k. `setmqaut -m QM1 -n "SYSTEM.FTE.EVENT.AGENT3" -t q -g FTEAGENTS +get +put`
- l. `setmqaut -m QM1 -n "SYSTEM.FTE.REPLY.AGENT1" -t q -g FTEAGENTS +get +put`
- m. `setmqaut -m QM1 -n "SYSTEM.FTE.REPLY.AGENT3" -t q -g FTEAGENTS +get +put`
- n. `setmqaut -m QM1 -n "SYSTEM.FTE.STATE.AGENT1" -t q -g FTEAGENTS +get +put +inq`
- o. `setmqaut -m QM1 -n "SYSTEM.FTE.STATE.AGENT3" -t q -g FTEAGENTS +get +put +inq`
- p. `setmqaut -m QM1 -n "SYSTEM.FTE" -t topic -g FTEUSERS +sub`
- q. `setmqaut -m QM1 -n "SYSTEM.FTE" -t topic -g FTEAGENTS +pub +sub`
- r. `setmqaut -m QM1 -n "SYSTEM.DEFAULT.MODEL.QUEUE" -t q -g FTEUSERS +dsp +browse +get +put`
- s. `setmqaut -m QM1 -n "SYSTEM.DEFAULT.MODEL.QUEUE" -t q -g FTEAGENTS +dsp +browse +get +put`

Weitere Informationen zum Befehl **setmqaut** finden Sie im Abschnitt [see setmqaut](#).

Sie finden ausführliche Informationen zur Erteilung einer Berechtigung für Gruppen unter [Gruppenberechtigungen für spezielle WebSphere MQ Managed File Transfer-Ressourcen](#).

## Computer 2 für Dateiübertragungen vorbereiten

Beschreibt die Vorbereitung von Computer 2 für Dateiübertragungen für das Dateiübertragungsszenario mit IBM WebSphere MQ zwei Computern.



## Informationen zu diesem Vorgang

Diese Task setzt voraus, dass auf Computer 2 das Betriebssystem Linux ausgeführt wird. Falls Sie bei der Ausführung dieses Szenarios eine andere Plattform verwenden, müssen Sie die entsprechenden plattformspezifischen Befehle einsetzen.

## Vorgehensweise

1. Installieren Sie auf beiden Computern IBM WebSphere MQ sowie die Komponente 'Managed File Transfer Agent' und alle anderen vorausgesetzten Komponenten (zum Beispiel 'Managed File Transfer Command Line Tools'). Ausführliche Informationen finden Sie im Abschnitt [Zu installierende Komponenten auswählen](#).

Dieser Schritt setzt voraus, dass Sie einen bereinigten Computer ohne vorherige Installationen von IBM WebSphere MQ oder IBM WebSphere MQ File Transfer Edition verwenden. Ist dies nicht der Fall, müssen Sie herausfinden, ob eine Koexistenz unterstützt wird. Außerdem müssen Sie das Installationsverfahren oder die Konfiguration der Umgebung entsprechend anpassen. Ausführliche Informationen hierzu finden Sie im Abschnitt [Mehrere Installationen](#).

Wenn Sie die Komponente 'Managed File Transfer Command Line Tools' installieren, können Sie Übertragungen nicht nur für Computer 1 definieren und von dort aus einleiten, sondern auch von Computer 2 aus (siehe vorliegendes Szenario).

2. Konfigurieren Sie als Benutzer, der Mitglied der Gruppe 'mqm' ist, Dateiübertragungen, bei denen QM1 auf Computer 1 als Koordinations-Warteschlangenmanager verwendet wird. Geben Sie folgende Befehle ein:

```
fteSetupCoordination -coordinationQMgr QM1 -coordinationQMgrHost <computer1_hostname>
-coordinationQMgrPort 1414 -coordinationQMgrChannel FTE.USER.SVRCONN
```

Sie müssen das generierte MQSC-Skript nicht ausführen, da es bereits bei der Konfiguration von Computer 1 ausgeführt wurde.

```
fteSetupCommands -connectionQMgr QM1 -connectionQMgrHost <computer1_hostname>
-connectionQMgrPort 1414 -connectionQMgrChannel FTE.USER.SVRCONN
```

Weitere Informationen finden Sie unter:

- [Übersicht über die WebSphere MQ Managed File Transfer-Topologie](#)
  - [fteSetupCoordination \(Details zur Koordinationseinrichtung\)](#)
  - [fteSetupCommands \(Datei 'command.properties' erstellen\)](#)
3. Listen Sie die Agenten auf, die bei QM1 registriert sind, um sicherzustellen, dass Ihre Konfiguration der Clientkonnektivität korrekt ist. Geben Sie den folgenden Befehl ein:

```
ftelistAgents
```

Sie sehen die folgende Ausgabe:

Agent Name:	Queue Manager Name:	Status:
AGENT1	QM1	Ready

4. Erstellen Sie den zweiten Dateiübertragungsagenten AGENT3:

```
fteCreateAgent -agentName AGENT3 -agentQMgr QM1 -agentQMgrHost <computer1_hostname>
-agentQMgrPort 1414 -agentQMgrChannel FTE.AGENT.SVRCONN
```

Wechseln Sie zu Computer 1 und geben Sie folgenden Befehl ein:

```
runmqsc QM1 < <AGENT3_create.mqsc>
```

5. Wechseln Sie zu Computer 2 und starten Sie als Benutzer `fteagent` Ihren neuen Dateiübertragungsagenten AGENT3.

```
fteStartAgent AGENT3
```

- Optional: Wechseln Sie zu Computer 1 und starten Sie AGENT1 als Benutzer fteagent und nicht als privilegierter IBM WebSphere MQ -Administrator erneut, der im ersten Szenario verwendet wurde.
- Prüfen Sie, ob Ihre Konfiguration der Clientkonnektivität korrekt ist, indem Sie die Agenten auflisten, die bei QM1 registriert sind.

```
ftelistAgents
```

Sie sehen die folgende Ausgabe:

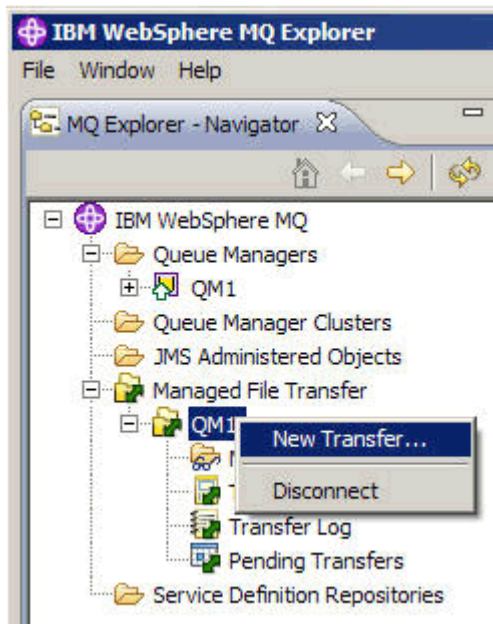
Agent Name:	Queue Manager Name:	Status:
AGENT1	QM1	READY
AGENT3	QM1	READY

## Lösung implementieren

Die Implementierung der Lösung in diesem Szenario umfasst die Verwendung von IBM WebSphere MQ Explorer, um eine Dateiübertragung zu definieren und einzuleiten. Sie können den Übertragungsfortschritt überwachen und seinen Erfolg durch die Untersuchung des Dateisystems überprüfen.

### Vorgehensweise

- Geben Sie die Quellendatei und das Zielverzeichnis an:
  - Erstellen Sie auf Computer 1 eine zu übertragende Beispieldatei, beispielsweise `C:\start\my-file.txt`.
  - Geben Sie auf Computer 2 ein vorhandenes Zielverzeichnis für die Übertragung dieser Datei an, beispielsweise `/home/end/`. Stellen Sie sicher, dass der Benutzer, der den Agenten gestartet hat, über Schreibzugriff für dieses Verzeichnis verfügt.
- Starten Sie IBM WebSphere MQ Explorer auf Computer 1. Starten Sie das Programm über das Menü "Start" (bzw. über eine funktionale Entsprechung) oder führen Sie den Befehl **MQExplorer** aus. Weitere Details finden Sie im Abschnitt [IBM WebSphere MQ Explorer starten](#).
- Klicken Sie im IBM WebSphere MQ Explorer-Navigator auf **Verwaltete Dateiübertragung**, klicken Sie mit der rechten Maustaste auf **QM1** und wählen Sie **Neue Übertragung** aus, um den Assistenten für neue Übertragung zu starten.



- Wählen Sie **AGENT1** als Quellenagent im Abschnitt **From** (Von) aus:

From:

Agent: AGENT1

Type: File

File:

Include subdirectories

5. Geben Sie den Pfad zu der zuvor erstellten Datei ein, beispielsweise C:\start\myfile.txt.

From:

Agent: AGENT1

Type: File

File: C:\Users\Ben Bakowski\Desktop\FTETEST\START\myDemoTransfer.txt

Include subdirectories

6. Wählen Sie im Abschnitt **An** den Agenten **AGENT2** als Zielagenten aus.

7. Geben Sie das zuvor bestimmte Zielverzeichnis ein. Beispiel: /home/end/.

To:

Agent: AGENT2

Type: File

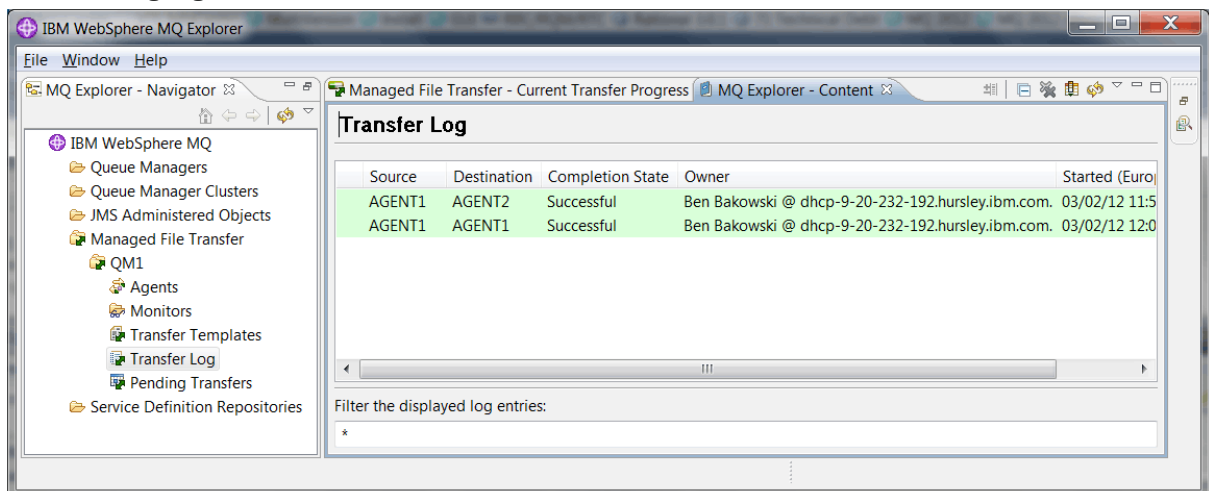
Directory: /home/ben/FTETEST/END/

File name: myDemoTransfer.txt

Overwrite files on the destination file system that have the same name

8. Klicken Sie auf **Jetzt fertigstellen**, damit die Übertragung beginnt.

9. Auf der Registerkarte **Aktueller Übertragungsfortschritt** in IBM WebSphere MQ Explorer können Sie den Übertragungsfortschritt überwachen:



10. Sie können das Dateisystem von Computer 2 auch manuell überprüfen, um sich zu vergewissern, dass die neue Datei vorhanden ist, beispielsweise /home/end/myfile.txt.

## Datei über die Befehlszeile übertragen

In diesem Abschnitt, in dem die Verwendung der Befehlszeile für die Einleitung einer Übertragung gezeigt wird, erhalten Sie einen Einblick in die Flexibilität von Dateiübertragungen. Im vorliegenden Szenario wird dies zwar nicht abgedeckt, Sie können aber mithilfe dieser Prinzipien und der Ant-Scriptingtechnologie wesentlich leistungstärkere Dateiübertragungsszenarios definieren und implementieren.

### Vorgehensweise

1. Löschen Sie die übertragene Datei aus der vorherigen Demonstration, beispielsweise `/home/end/myfile.txt`.
2. Geben Sie als Benutzer `fteuser` den Befehl **fteCreateTransfer** ein, um die Übertragung Ihrer Datei von `C:\start\myfile.txt` auf Computer 1 auf `/home/end/myfile.txt` auf Computer 2 einzuleiten:

```
fteCreateTransfer -sa AGENT1 -sm QM1 -da AGENT3 -dm QM1 -w -dd "/home/end/" "C:\start\myfile.txt"
```

3. Vergewissern Sie sich, dass die Dateiübertragung erfolgreich verlaufen ist, indem Sie das Dateiübertragungsprotokoll in IBM WebSphere MQ Explorer überprüfen oder das Dateisystem manuell untersuchen.

### Nächste Schritte

Weitere Informationen zur IBM WebSphere MQ -Sicherheit finden Sie unter [Sicherheit](#).

Möglicherweise möchten Sie weitere Funktionen der Dateiübertragungsfunktionalität über externe Medien erkunden.

- Sie finden ausführliche Informationen zum Hinzufügen der Prüffunktion für die Bereitstellung des verwalteten Aspekts der verwalteten Dateiübertragung unter „[Prüffunktion zur verwalteten Dateiübertragung hinzufügen](#)“ auf Seite 188.
- Weitere Informationen zu:
  - Triggering: Verschieben neuer Dateien bei deren Anzeige.
  - Triggering: Konfiguration der Anzeige einer einzelnen Datei für die Einleitung einer Übertragung mehrerer Dateien.
  - Scripterstellung für Übertragungen mit Apache Ant.

Weitere Informationen finden Sie im Abschnitt [https://www.ibm.com/developerworks/websphere/library/techarticles/1003\\_phillips/1003\\_phillips.html](https://www.ibm.com/developerworks/websphere/library/techarticles/1003_phillips/1003_phillips.html).

- Ihre eigenen Sicherheitsanforderungen erfordern möglicherweise ein anderes Zugriffsmodell als das im vorliegenden Szenario verwendete Modell. Sie finden weitere Informationen zu Best Practices für einen höheren Schutz Ihrer Umgebung unter [https://www.ibm.com/developerworks/websphere/library/techarticles/0902\\_wyatt/0902\\_wyatt.html](https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html).

## Prüffunktion zur verwalteten Dateiübertragung hinzufügen

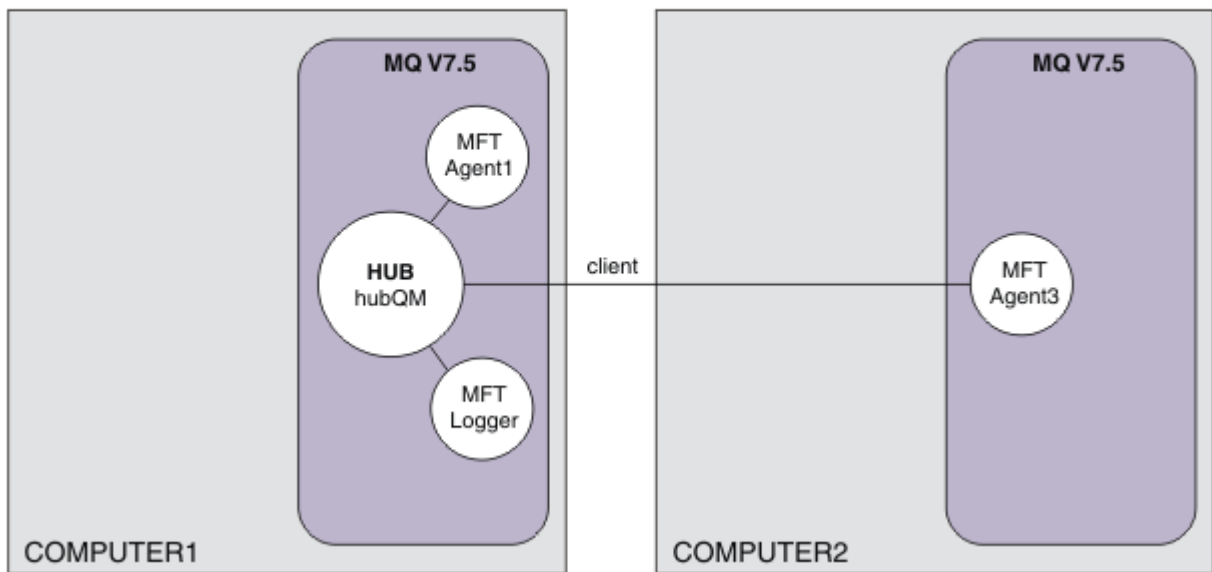
Im vorliegenden Szenario erfahren Sie, wie Sie eine Protokollfunktion konfigurieren und mit dieser Funktion ein Auditprotokoll bereitstellen.

Sie haben durch die beiden vorherigen Szenarios „[Basisdateiübertragung im Detail](#)“ auf Seite 167 und „[Dateiübertragung zwischen zwei Computern im Detail](#)“ auf Seite 180 eine Demonstration der Funktion 'Managed File Transfer' durchlaufen und sind mit der Konfiguration einer Dateiübertragungstopologie vertraut. Sie verstehen, wie IBM WebSphere MQ Version 7.5 durch die Bereitstellung von Funktionen zur Protokollierung und Prüfung von Dateiübertragungen den verwalteten Aspekt von Managed File Transfer bereitstellt.

Es wird vorausgesetzt, dass Sie das zweite Szenario der verwalteten Dateiübertragung unter Einhaltung der Anweisungen im Szenario „[Dateiübertragung zwischen zwei Computern im Detail](#)“ auf Seite 180 ausgeführt haben.

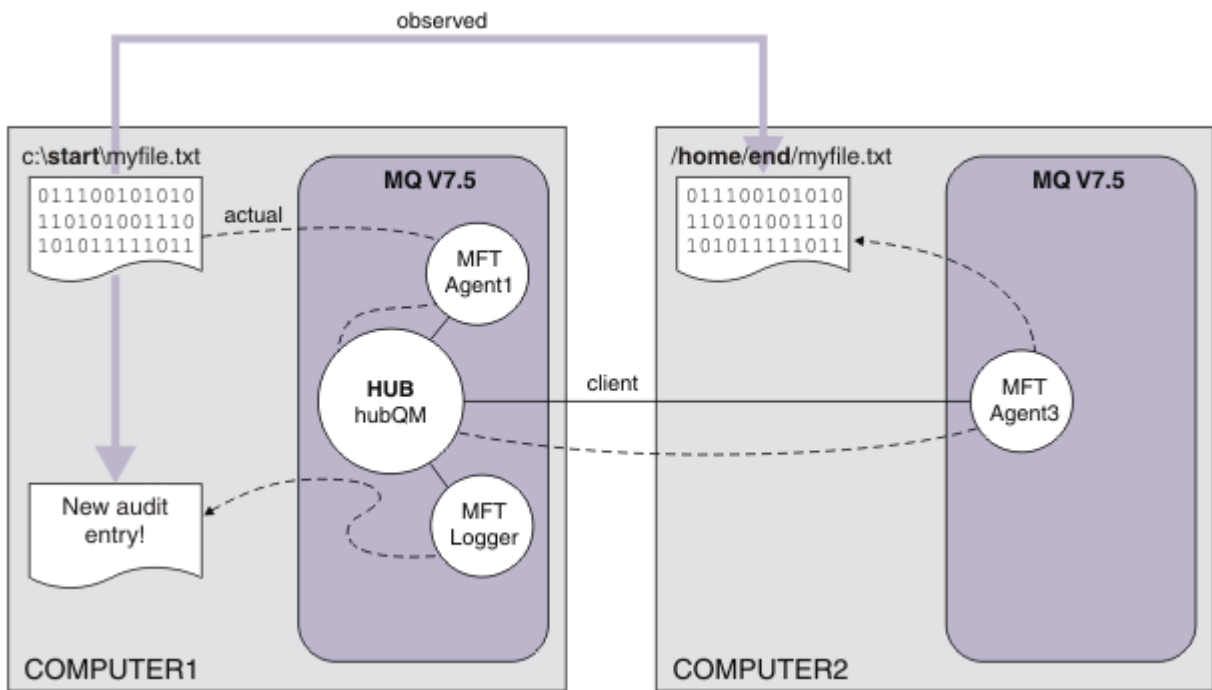
## Übersicht

In diesem Szenario fahren Sie mit der vorhandenen Windows- und Linux -Topologie fort, die Sie in „Dateiübertragung zwischen zwei Computern im Detail“ auf Seite 180 konfiguriert haben, und aktivieren die Dateiprotokollierungsfunktion. Für die Dateiprotokollierungsfunktion müssen keine anderen Komponenten oder Produkte installiert werden. Daher müssen Sie bei diesem Szenario keine Lizenzierungsaspekte oder Änderungen der Berechtigungen im Vergleich mit dem Dateiübertragungsszenario mit zwei Computern berücksichtigen.



Sie können auch eine Datenbankprotokollfunktion implementieren, die sich in einer Produktionsumgebung möglicherweise besser eignet, da sie beispielsweise für eine Skalierbarkeit und einen Ausweichbetrieb eingesetzt werden kann. Wenn Sie dieses Szenario jedoch einfach halten möchten und keine weiteren Produkte installieren möchten, verwenden Sie die vollständig unterstützte Dateiprotokollfunktion. Es werden keine weiteren Sicherheitsaspekte berücksichtigt. Da diese Funktion möglicherweise Prüfinformationen bereitstellt, kann es sinnvoll sein, im entsprechenden Kontext die Sicherheit zu erhöhen.

In diesem Szenario wird eine Dateiübertragung eingeleitet und die Einzeldaten werden in einer Protokolldatei erfasst.



## Lösung planen

An dieser Stelle werden die Vorbedingungen und Lizenzvoraussetzungen für die ergänzende IBM WebSphere MQ-Prüffunktion in einem Szenario der verwalteten Dateiübertragung beschrieben.

In diesem Szenario erstellen Sie eine Protokollfunktion für die Prüfung von Dateiübertragungen. Da die Protokollfunktion am ehesten einem Agentenprozess entspricht, können Sie den vorhandenen Benutzer `ftagent` zum Starten und Stoppen der Protokollfunktion verwenden. Sie können Ihren eigenen Benutzer oder Ihre eigene Gruppe für die Verwaltung der Protokollfunktion erstellen.

## Voraussetzungen und Lizenzen

Sie benötigen eine funktionierende Konfiguration aus dem Szenario „Dateiübertragung zwischen zwei Computern im Detail“ auf Seite 180.

## Lösung implementieren

Hier finden Sie eine Beschreibung der Änderung von Computer 1 zur Konfiguration einer Dateiprotokollfunktion für die ergänzende IBM WebSphere MQ-Prüffunktion in einem Szenario der verwalteten Dateiübertragung.

## Vorgehensweise

1. Prüfen Sie, ob die Gruppe `ftagent` bei der Ausführung des Protokollierungsprozesses über die entsprechenden Berechtigungen für IBM WebSphere MQ-Objekte verfügt. Geben Sie folgende Befehle ein:

```
setmqaut -m hubQM -n "SYSTEM.FTE.LOG.RJCT.MYFILELOGGER" -t q -g FTAGENTS +put
setmqaut -m hubQM -n "SYSTEM.FTE.LOG.CMD.MYFILELOGGER" -t q -g FTAGENTS +get
```

Sie finden ausführliche Informationen unter [Berechtigungen für die Datenbankprotokollfunktion](#).

2. Erstellen Sie als IBM WebSphere MQ-Administrator eine Dateiprotokollfunktion und verwenden Sie 'hubQM' als Warteschlangenmanager der Protokollfunktion.

```
fteCreateLogger -loggerType FILE -loggerQMgr hubQM -fileLoggerMode LINEAR -fileSize 5MB myFileLogger
runmqsc hubQM < <MYFILELOGGER_create.mqsc>
```

Sie finden weitere Details unter [fteCreateLogger \(Erstellung einer WebSphere MQ Managed File Transfer-Protokollfunktion\)](#). Sie können eine Datenbankprotokollfunktion auch während der Produktion einsetzen.

3. Starten Sie die Protokollfunktion als Benutzer ftagent.

```
fteStartLogger MYFILELOGGER
```

4. Vergewissern Sie sich, dass die Protokollfunktion gestartet wurde.

Zur Überprüfung des Starts der Protokollfunktion müssen Sie das Dateisystem untersuchen. Nachdem Sie die Dateiprotokollfunktion mit den obigen Befehlen konfiguriert haben, finden Sie die Protokolle in <MQ INSTALL>/mqft/logs/hubQM/loggers/MYFILELOGGER/logs. Bestätigen Sie, dass output0.log die Nachricht "BFGDB0023I: The logger has completed startup activities and is now running." enthält.

5. Löschen Sie die übertragene Datei aus der vorherigen Demonstration, beispielsweise /home/end/myfile.txt.
6. Starten Sie als Benutzer ftuser mit dem Befehl fteCreateTransfer ([Erstellung einer neuen Dateiübertragung](#)) auf Computer 1 die Übertragung Ihrer Datei aus C:\start\myfile.txt auf Computer 1 an /home/end/myfile.txt auf Computer 2. Geben Sie den folgenden Befehl ein:

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT3 -dm hubQM -w -dd "/home/end/" "C:\start\myfile.txt"
```

7. Vergewissern Sie sich, dass die Protokollfunktion diese Übertragung erfasst und machen Sie sich mit dem Inhalt des Protokolleintrags vertraut.

- a. Datei öffnen <MQ INSTALL>/mqft/logs/hubQM/loggers/MYFILELOGGER/MYFILELOGGER-XXXXXXXXX.log
- b. Der Protokolleintrag enthält die von Ihnen eingeleitete Übertragung, und zwar einschließlich der Positionen der Dateiquelle und des Dateiziels, des Datums, der Uhrzeit und der ID der anfordernden Stelle. Beispiel:

```
2012-03-23T16:42:21;414d5120514d31202020202020202020202020207a556b4f2000aa03;[TSTR]; ;
AGENT1;hubQM;STANDARD;AGENT3;hubQM;User;;;com.ibm.wmqfte.SourceAgent=AGENT1,
com.ibm.wmqfte.DestinationAgent=AGENT3, com.ibm.wmqfte.MqmdUser=User,
com.ibm.wmqfte.OriginatingUser=User, com.ibm.wmqfte.OriginatingHost=
dhcp-9-10-123-123.hursley.ibm.com., com.ibm.wmqfte.TransferId=
414d5120514d31202020202020202020202020207a556b4f2000aa03, com.ibm.wmqfte.Priority=0;

2012-03-23T16:42:21;414d5120514d31202020202020202020202020207a556b4f2000aa03;[TPRO];0 ;
C:\start\myfile.txt;51447;file;leave ;;;;/home/end/myfile.txt;51447;file;
;;;;;;

2012-03-23T16:42:21;414d5120514d31202020202020202020202020207a556b4f2000aa03;[TCOM];0 ;
AGENT1;hubQM;STANDARD;AGENT3;hubQM;STANDARD;User;;BFGRP0032I: The file transfer
request has successfully completed.;com.ibm.wmqfte.SourceAgent=AGENT1,
com.ibm.wmqfte.DestinationAgent=AGENT3, com.ibm.wmqfte.MqmdUser=User,
com.ibm.wmqfte.OriginatingUser=User, com.ibm.wmqfte.OriginatingHost=
dhcp-9-20-123-123.hursley.ibm.com.,
com.ibm.wmqfte.TransferId=414d5120514d31202020202020202020202020207a556b4f2000aa03,
com.ibm.wmqfte.Priority=0;
```

## Glossar

Dieses Glossar enthält Begriffe und Definitionen für IBM WebSphere MQ.

Folgende Querverweise werden in diesem Glossar verwendet:

- 'Siehe' verweist von einem Terminus zu einem bevorzugt zu verwendenden Synonym oder von einem Akronym oder einer Abkürzung zu der Definition der vollständigen Langform des Terminus.

- 'Siehe auch' verweist auf einen verwandten oder gegensätzlichen Begriff.

„A“ auf Seite 192 „B“ auf Seite 195 „C“ auf Seite 196 „D“ auf Seite 201 „E“ auf Seite 203 „F“ auf Seite 204 „G“ auf Seite 206 „H“ auf Seite 206 „I“ auf Seite 207 „J“ auf Seite 209 „K“ auf Seite 210 „L“ auf Seite 210 „M“ auf Seite 212 „N“ auf Seite 216 „O“ auf Seite 217 „P“ auf Seite 219 „Q“ auf Seite 222 „R“ auf Seite 223 „S“ auf Seite 226 „T“ auf Seite 231 „U“ auf Seite 234 „V“ auf Seite 235 „W“ auf Seite 235 „X“ auf Seite 236

## A

### **Ursachencode für abnormale Beendigung**

Ein 4 Bytes langer hexadezimaler Code, der einen Fehler in einem Programm, das unter dem Betriebssystem z/OS ausgeführt wird, eindeutig bezeichnet.

### **Abstrakte Klasse**

In der objektorientierten Programmierung eine Klasse, die ein Konzept darstellt. Von einer abstrakten Klasse abgeleitete Klassen stellen Implementierungen des Konzepts dar. Ein Objekt kann nicht aus einer abstrakten Klasse erstellt werden, d. h., es kann nicht instanziiert werden. Siehe auch [Übergeordnete Klasse](#).

### **Zugriffssteuerung**

In der IT-Sicherheit der Prozess, durch den sichergestellt wird, dass Benutzer nur auf solche Ressourcen eines Computersystems zugreifen können, für die sie eine Berechtigung besitzen.

### **Zugriffssteuerungsliste (ACL)**

In der IT-Sicherheit eine Liste, die einem Objekt zugeordnet ist und in der alle Subjekte, die auf das Objekt zugreifen können, sowie deren Zugriffsberechtigungen festgelegt sind.

### **Verantwortlichkeit**

Die Eigenschaft, für seine Aktionen verantwortlich zu sein.

### **ACL (Access Control List)**

Siehe [Zugriffssteuerungsliste](#).

### **aktives Protokoll**

Eine Datei fester Größe, in der Wiederherstellungsereignisse in dem Moment aufgezeichnet werden, in dem sie stattfinden. Wenn das aktive Protokoll voll ist, wird der Inhalt des aktiven Protokolls in das Archivprotokoll kopiert.

### **Aktive Warteschlangenmanager-Instanz**

Die Instanz eines aktiven Multi-Instanz-Warteschlangenmanagers, der Anforderungen verarbeitet. Es gibt nur eine aktive Instanz eines Multi-Instanz-Warteschlangenmanagers.

### **Adapter**

Eine intermediäre Softwarekomponente, die die Kommunikation zwischen zwei anderen Softwarekomponenten ermöglicht.

### **Adressraum (ASID)**

Der Adressbereich, der einem Computerprogramm oder Prozess zur Verfügung steht. Adressraum kann sich auf eine physische Speichereinheit, einen virtuellen Speicher oder beides beziehen. Siehe auch [zugehöriger Adressraum](#) und [Pufferpool](#).

### **Verwaltungstasche**

In WebSphere MQ Administration Interface (MQAI) ein Typ von Datenbehälter, der für die Verwaltung von WebSphere MQ erstellt wird und für den implizit gilt, dass er die Reihenfolge der Datenelemente ändern, Listen erstellen und Selektoren in einer Nachricht überprüfen kann.

### **Topic-Verwaltungsobjekt**

Ein Objekt, das es ermöglicht, Topics bestimmte, nicht standardmäßige Attribute zuzuweisen.

### **Administratorbefehl**

Ein Befehl zum Verwalten von WebSphere MQ-Objekten, z. B. Warteschlangen, Prozesse und Namenslisten.

### **Advanced Program-to-Program Communication (APPC)**

Eine Implementierung des SNA LU 6.2-Protokolls, das die Kommunikation und die gemeinsame Verarbeitung von Programmen in einem Systemverbund ermöglicht.



**Erweiterter Telemetrieclient**

Siehe [Telemetrieclient \(erweitert\)](#).

**Affinität**

Eine Zuordnung von Objekten, zwischen denen eine Beziehung oder gegenseitige Abhängigkeit besteht.

**Alert**

Eine Nachricht oder eine andere Meldung, die auf ein Ereignis oder ein bevorstehendes Ereignis hinweist.

**Alertmonitor**

In WebSphere MQ for z/OS eine Komponente des CICS-Adapters zur Verarbeitung ungeplanter Ereignisse, die als Ergebnis von Verbindungsanforderungen an WebSphere MQ for z/OS auftreten.

**Aliaswarteschlange**

Ein WebSphere-MQ-Objekt, dessen Name ein Alias für eine Basiswarteschlange oder ein Basistopic ist, die bzw. das für den lokalen Warteschlangenmanager definiert ist. Wenn eine Anwendung oder ein Warteschlangenmanager eine Aliaswarteschlange verwendet, wird der Aliasname aufgelöst und die angeforderte Operation für das zugeordnete Basisobjekt ausgeführt.

**Aliaswarteschlangenobjekt**

Ein WebSphere-MQ-Objekt, dessen Name ein Alias für eine Basiswarteschlange ist, die für den lokalen Warteschlangenmanager definiert ist. Wenn eine Anwendung oder ein Warteschlangenmanager eine Aliaswarteschlange verwendet, wird der Aliasname aufgelöst und die angeforderte Operation für die zugeordnete Basiswarteschlange ausgeführt.

**Zugehöriger Adressraum**

Ein z/OS-Adressraum, der mit WebSphere MQ for z/OS verbunden ist.

**Adressraumverbindung**

Siehe [Zugehöriger Adressraum](#).

**Alternative Benutzerberechtigung**

Die Fähigkeit einer Benutzer-ID, eine andere Benutzer-ID für Sicherheitsprüfungen zu übergeben. Wenn eine Anwendung ein WebSphere-MQ-Objekt öffnet, kann sie im MQOPEN-, MQPUT1- oder MQSUB-Aufruf eine Benutzer-ID übergeben, die der Warteschlangenmanager anstelle der Benutzer-ID, die der Anwendung zugeordnet ist, für Berechtigungsprüfungen verwendet.

**Alternative Benutzersicherheit**

Unter z/OS sind dies die Berechtigungsprüfungen, die ausgeführt werden, wenn eine Anwendung beim Öffnen eines WebSphere MQ-Objekts eine alternative Benutzerberechtigung anfordert.

**APAR**

Siehe [Authorized Program Analysis Report](#).

**APF**

Siehe [Authorized Program Facility](#).

**API-Steuerübergabeexit**

Ein vom Benutzer geschriebenes Programm, dessen Konzept dem eines API-Exits gleicht. Es wird nur für CICS-Anwendungen in WebSphere MQ for z/OS unterstützt.

**API-Exit**

Ein vom Benutzer geschriebenes Programm, das die Funktion eines MQI-Aufrufs überwacht oder ändert. Für jeden von einer Anwendung abgesetzten MQI-Aufruf wird der API-Exit aufgerufen, bevor der Warteschlangenmanager mit der Verarbeitung des Aufrufs beginnt und nachdem er die Verarbeitung des Aufrufs abgeschlossen hat. Mit dem API-Exit können alle Parameter im MQI-Aufruf überprüft und geändert werden.

**APPC**

Siehe [Advanced Program-to-Program Communication](#).

**Anwendungsdefiniertes Format**

Anwendungsdaten in einer Nachricht, deren Bedeutung von der Benutzeranwendung definiert wird. Siehe auch [Integriertes Format](#).

**Anwendungsumgebung**

Die Umgebung, die die Software und die Server- bzw. Netzinfrastruktur enthält, die diese Software unterstützt.

**Sicherheit auf Anwendungsebene**

Die Sicherheitsservices, die aufgerufen werden, wenn eine Anwendung einen MQI-Aufruf ausgibt.

**Anwendungsprotokoll**

Auf Windows-Systemen ein Protokoll, in dem wichtige Anwendungsereignisse aufgezeichnet werden.

**Anwendungswarteschlange**

Eine lokale Warteschlange, die erfordert, dass Auslösenachrichten geschrieben werden, wenn Auslöser aktiviert und Auslösebedingungen erfüllt sind.

**Archivprotokoll**

Eine Datei in einer Speichereinheit, in die WebSphere MQ den Inhalt jeder aktiven Protokolldatei kopiert, wenn deren Größenbegrenzung erreicht wird. Siehe auch [Wiederherstellungsprotokoll](#).

**ARM**

Siehe [Automatic Restart Manager](#).

**ASID**

Siehe [Adressraum](#).

**Asymmetrische Verschlüsselung**

Ein Verschlüsselungssystem, in dem zwei Schlüssel verwendet werden: ein öffentlicher Schlüssel, der jedem zugänglich ist, und ein privater Schlüssel, der nur dem Empfänger oder Sender der Nachricht bekannt ist. Siehe auch [Symmetrische Verschlüsselung](#).

**Asynchrone Verarbeitung**

Ein Prozess, bei dem eine Anwendung mithilfe einer Gruppe von MQI-Aufrufen Nachrichten aus einer Gruppe von Warteschlangen verarbeitet. Nachrichten werden der Anwendung zugestellt, indem eine von der Anwendung angegebene Codeeinheit aufgerufen wird, wobei die Nachricht oder ein Token übergeben wird, das die Nachricht darstellt.

**Asynchrone Nachrichtenübertragung**

Ein Kommunikationsverfahren zwischen Programmen, bei dem ein Programm eine Nachricht in eine Nachrichtenwarteschlange stellt und dann seine Verarbeitungsschritte fortsetzt, ohne auf die Beantwortung der Nachricht zu warten. Siehe auch [Synchrone Nachrichtenübertragung](#).

**Asynchroner Put**

Das Einreihen einer Nachricht in eine Warteschlange durch eine Anwendung, ohne dass auf eine Antwort vom Warteschlangenmanager gewartet wird.

**Attribut**

1. In der objektorientierten Programmierung eine Eigenschaft eines Objekts oder einer Klasse, die eindeutig von allen anderen Eigenschaften unterschieden werden kann. Attribute beschreiben häufig Statusinformationen.
2. Ein Merkmal einer Entität, das die Entität beschreibt. Beispiel: Die Telefonnummer eines Mitarbeiters ist ein Mitarbeiterattribut. Siehe auch [Entität](#).

**Authentifizierung**

Ein Sicherheitsservice, der nachweist, dass ein Benutzer eines Computersystems wirklich die Person ist, die er zu sein vorgibt. Typische Mechanismen zum Implementieren dieses Service sind Kennwörter und digitale Signaturen.

**Authentifizierungsdatenobjekt**

Ein Objekt, das die Definitionen bereitstellt, die benötigt werden, um zur Unterstützung der SSL-Sicherheit (Secure Sockets Layer) Zertifikatswiderrufslisten mithilfe von LDAP-Servern zu überprüfen.

**Berechtigungsprüfung**

Siehe [Berechtigungsprüfung](#).

**Berechtigung**

Der Prozess, bei dem einem Benutzer, System oder Prozess entweder uneingeschränkter oder eingeschränkter Zugriff auf ein Objekt, eine Ressource oder eine Funktion erteilt wird.

**Berechtigungsprüfung**

Eine Sicherheitsprüfung, die durchgeführt wird, wenn ein Benutzer oder eine Anwendung versucht, auf eine Systemressource zuzugreifen, z. B., wenn ein Administrator versucht, einen Befehl zur Verwaltung von WebSphere MQ abzusetzen, oder wenn eine Anwendung versucht, eine Verbindung mit einem Warteschlangenmanager herzustellen.

**Berechtigungsdatei**

Eine Datei, die Sicherheitsdefinitionen für ein Objekt, eine Klasse von Objekten oder alle Klassen von Objekten enthält.

**Berechtigungs-service**

In WebSphere MQ auf UNIX- und Linux -Systemen und WebSphere MQ for Windows ein Service, der die Berechtigungsprüfung für Befehle und MQI-Aufrufe für die dem Befehl oder Aufruf zugeordnete Benutzer-ID bereitstellt.

**Authorized Program Analysis Report (APAR)**

Eine Anforderung zur Behebung eines Fehlers in einem unterstützten Release eines von IBM gelieferten Programms.

**Authorized Program Facility (APF)**

In einer z/OS-Umgebung eine Einrichtung zur Identifikation von Programmen, die berechtigt sind, eingeschränkte Funktionen zu verwenden.

**Automatic Restart Manager (ARM)**

Eine z/OS-Wiederherstellungsfunktion, die Stapeljobs und gestartete Tasks automatisch erneut starten kann, nachdem sie oder die Systeme, auf denen sie ausgeführt werden, unerwartet beendet wurden.

**B****Rücksetzung**

Eine Operation, bei der alle Änderungen an Ressourcen rückgängig gemacht werden, die während der aktuellen Arbeitseinheit vorgenommen wurden. Siehe auch [Commit](#).

**Tasche**

Siehe [Datenbehälter](#).

**Grenze**

Eine z/OS-Speichergrenze, die in 64-Bit-Systemen auf 2 GB gesetzt ist. Die Grenze trennt den Speicher unterhalb der 2-GB-Adresse vom Speicher oberhalb der 2-GB-Adresse. Der Bereich oberhalb der Grenze ist für Daten bestimmt; dort werden keine Programme ausgeführt.

**Basic Mapping Support (BMS)**

Eine Schnittstelle zwischen CICS und Anwendungsprogrammen, die Eingabe- und Ausgabeanzeigedaten formatiert und mehrseitige Ausgabenachrichten ohne Berücksichtigung von Steuerzeichen, die von verschiedenen Terminals verwendet werden, weiterleitet.

**behavior**

In der objektorientierten Programmierung die in eine Methode integrierte Funktionalität.

**BMS**

Siehe [Basic Mapping Support](#).

**Booch-Methode**

Eine objektorientierte Methode, die Benutzer bei der Entwicklung von Systemen auf Basis des objektorientierten Konzepts unterstützt.

**Bootstrap Data Set (BSDS)**

Eine VSAM-Datei, die einen Bestand aller aktiven und archivierten Protokolldateien, die WebSphere MQ for z/OS bekannt sind, und einen Umlaufbestand aller kürzlich ausgeführten Aktivitäten von WebSphere MQ for z/OS enthält. Das Bootstrap-Dataset ist für den Neustart des WebSphere MQ for z/OS-Subsystems erforderlich.

**Anzeigen**

Beim Message-Queuing das Kopieren einer Nachricht, ohne sie aus der Warteschlange zu entfernen. Siehe auch [Abrufen](#) und [Einreihen](#).

**Anzeigecursor**

Beim Message-Queuing ein Anzeiger, der beim Durchsuchen einer Warteschlange nach der nächsten Nachricht verwendet wird.

**BSDS**

Siehe [Bootstrap-Dataset](#).

**Pufferpool**

Ein Speicherbereich, in den Datenseiten gelesen und in dem sie während der Verarbeitung geändert und aufbewahrt werden. Siehe auch [Adressraum](#).

**Integriertes Format**

Anwendungsdaten in einer Nachricht, deren Bedeutung vom Warteschlangenmanager definiert wird. Siehe auch [Anwendungsdefiniertes Format](#).

**C****Zertifizierungsstelle**

Siehe [Zertifizierungsstelle](#).

**CAF**

Siehe [Client Attachment Feature](#).

**Callback (Rückruf)**

Die Routine eines Nachrichtenkonsumenten oder Ereignishandlers.

**CCDT (Client Channel Definition Table)**

Siehe [Definitionstabelle für Clientkanal](#).

**CCF (Channel Control Funktion)**

Siehe [Kanalsteuerfunktion](#).

**CCSID**

Siehe [ID des codierten Zeichensatzes](#).

**CDF (Channel Definition File)**

Siehe [Kanaldefinitionsdatei](#).

**Zertifizierungsstelle (CA)**

Ein vertrauenswürdiger Dritter (Organisation oder Unternehmen), der die digitalen Zertifikate als Antwort auf eine Zertifikatssignieranforderung ausstellt. Die Zertifizierungsstelle überprüft die Identität der Personen, denen das eindeutige Zertifikat erteilt wird. Siehe auch [Secure Sockets Layer](#).

**Zertifikatkette**

Eine Hierarchie von Zertifikaten, die durch Ihre Verschlüsselung miteinander verbunden sind. Sie beginnt mit dem persönlichen Zertifikat und endet mit dem Stammelement am Anfang der Kette.

**Ablauf des Zertifikats**

Ein digitales Zertifikat enthält einen Datumsbereich für die Gültigkeit des Zertifikats. Außerhalb des gültigen Datumsbereichs gilt das Zertifikat als "abgelaufen".

**Zertifikatswiderrufsliste (CRL)**

Eine Liste mit Zertifikaten, die vor ihrem festgelegten Ablaufdatum widerrufen wurden. Zertifikatswiderrufslisten werden von der Zertifizierungsstelle verwaltet und beim SSL-Handshake verwendet, um sicherzustellen, dass die beteiligten Zertifikate nicht widerrufen wurden.

**Zertifikatsspeicher**

Der Windows-Name für ein Schlüsselrepository.

**Zertifikatssignieranforderung (CSR)**

Eine Anforderung, die den öffentlichen Schlüssel und definierten Subjektnamen einer Einrichtung oder Organisation enthält. Sie wird an die Zertifizierungsstelle gesendet, damit diese eine digitale Signatur für die betreffende Einrichtung ausstellt.

**CF**

Siehe [Coupling-Facility](#).

**CFSTRUCT**

Ein WebSphere MQ-Objekt, das die Verwendung einer Coupling-Facility durch den Warteschlangenmanager beschreibt.

**Kanal**

Ein WebSphere MQ-Objekt, das eine Kommunikationsverbindung zwischen zwei Warteschlangenmanagern (Nachrichtenkanal) oder zwischen einem Client und einem Warteschlangenmanager (MQI-Kanal) definiert. Siehe auch [Nachrichtenkanal](#) und [MQI-Kanal](#).

**Kanalrückruf**

Ein Mechanismus, der sicherstellt, dass die Kanalverbindung zur richtigen Maschine hergestellt wird. Bei einem Kanalrückruf sendet ein Senderkanal mithilfe der Senderdefinition einen Rückruf an den ursprünglichen Requesterkanal.

**Kanalsteuerfunktion**

Ein Programm zur Übergabe von Nachrichten aus einer Übertragungswarteschlange an eine Kommunikationsverbindung und von einer Kommunikationsverbindung an eine lokale Warteschlange, einschließlich einer Bedienerkonsolenschnittstelle zur Definition und Steuerung von Kanälen.

**Kanaldefinitionsdatei**

Eine Datei mit Kommunikationskanaldefinitionen, die Übertragungswarteschlangen und Kommunikationsverbindungen einander zuordnen.

**channel event**

Ein Ereignis, das Bedingungen meldet, die bei Kanaloperationen erkannt werden, z. B. beim Starten oder Stoppen einer Kanalinstanz. Kanalereignisse werden in den Warteschlangenmanagern auf beiden Seiten des Kanals generiert.

**Kanalexitprogramm**

Ein vom Benutzer geschriebenes Programm, das an einer der definierten Stellen in der Verarbeitungsreihenfolge eines Nachrichtenkanalagenten aufgerufen wird.

**Kanalinitiator**

Eine Komponente der verteilten Steuerung von Warteschlangen in WebSphere MQ, die die Initialisierungswarteschlange überwacht und bei Erfüllung bestimmter Auslösebedingungen den Senderkanal startet.

**Kanallistener**

Eine Komponente der verteilten Steuerung von Warteschlangen in WebSphere MQ, die das Netz auf Startanforderungen überwacht und dann den Empfängerkanal startet.

**Prüfpunkt**

Eine Stelle in einem Programm, an der eine Prüfung durchgeführt wird oder an der Daten aufgezeichnet werden, damit das Programm im Falle einer Unterbrechung neu gestartet werden kann.

**CI (Control Interval)**

Siehe [Steuerintervall](#).

**CipherSpec**

Die Kombination aus Verschlüsselungsalgorithmus und Hashfunktion, die nach Abschluss der Authentifizierung auf eine SSL-Nachricht angewendet wird.

**Cipher-Suite**

Die Kombination aus Authentifizierung, Schlüsselaustauschalgorithmus und SSL-Verschlüsselungsspezifikation, die für sicheren Datenaustausch verwendet wird.

**ciphertext**

Daten, die verschlüsselt wurden. Verschlüsselter Text kann erst gelesen werden, nachdem er mit einem Schlüssel in unverschlüsselten Text umgewandelt (entschlüsselt) wurde. Siehe auch [Klartext](#).

**Umlaufprotokollierung**

In WebSphere MQ auf UNIX- und Linux -Systemen und WebSphere MQ for Windows der Prozess, bei dem alle Neustartdaten in einem Ring aus Protokolldateien aufbewahrt werden. Siehe auch [Lineare Protokollierung](#).

**CL**

Siehe [Command Language](#).

**Klasse**

Im objektorientierten Design oder in der objektorientierten Programmierung ein Modell oder eine Schablone, das bzw. die verwendet werden kann, um Objekte mit einer gemeinsamen Definition und gemeinsamen Eigenschaften, Operationen und Verhaltensweisen zu erstellen. Ein Objekt ist eine Instanz einer Klasse.

**Klassenhierarchie**

Die Beziehungen zwischen Klassen, die eine Einfachvererbung gemeinsam verwenden.

**Klassenbibliothek**

In der objektorientierten Programmierung eine Sammlung vordefinierter Klassen oder codierter Schablonen, die bei der Entwicklung einer Anwendung von einem Programmierer angegeben und verwendet werden können.

**Klartext**

Eine Zeichenfolge, die in einem lesbaren Format über ein Netz gesendet wird. Sie kann zum Zweck der Komprimierung codiert werden, aber ohne großen Aufwand wieder decodiert werden. Siehe auch [Verschlüsselter Text](#).

**Client**

Eine Laufzeitkomponente, die lokalen Benutzeranwendungen Zugriff auf Services zur Steuerung von Warteschlangen auf einem Server ermöglicht. Die von den Anwendungen verwendeten Warteschlangen befinden sich auf dem Server. Siehe auch [WebSphere MQ-Client](#), [WebSphere MQ-Java-Client](#) und [Vollständig verwalteter WebSphere MQ-.NET-Client](#).

**Clientanwendung**

Eine Anwendung, die auf einer Workstation ausgeführt wird und mit einem Client verbunden ist, über den sie Zugriff auf Warteschlangensteuerservices erhält, die auf einem Server zur Verfügung stehen.

**Client Attachment Feature (CAF)**

Eine Option, die den Anschluss von Clients an z/OS unterstützt.

**Definitionstabelle für Clientkanal (CCDT)**

Eine Datei, die mindestens eine Clientverbindungskanaldefinition enthält.

**Clientverbindungskanaltyp**

Der MQI-Kanaldefinitionstyp, der einem WebSphere MQI-Client zugeordnet ist. Siehe auch [Serververbindungskanaltyp](#).

**CLUSRCVR**

Siehe [Clusterempfängerkanal](#).

**CLUSSDR**

Siehe [Clustersenderkanal](#).

**Cluster**

In WebSphere MQ eine Gruppe von mindestens zwei Warteschlangenmanagern auf einem oder mehreren Computern, die automatisch miteinander verbunden werden und Warteschlangen und Themen zum Zweck des Lastausgleichs und der Redundanz gemeinsam nutzen.

**Clusterwarteschlange**

Eine lokale Warteschlange, die von einem Clusterwarteschlangenmanager gehostet wird und als Ziel für Nachrichten definiert ist, die von einer Anwendung eingereicht werden, die mit einem beliebigen Warteschlangenmanager im Cluster verbunden ist. Alle Anwendungen, die Nachrichten abrufen, müssen lokal verbunden sein.

**Clusterwarteschlangenmanager**

Ein Warteschlangenmanager, der zu einem Cluster gehört. Ein Warteschlangenmanager kann mehreren Clustern angehören.

**Clusterempfängerkanal (CLUSRCVR)**

Ein Kanal, über den Clusterwarteschlangenmanager Nachrichten von anderen Warteschlangenmanagern im Cluster sowie Clusterinformationen von den Repository-Warteschlangenmanagern empfangen können.

**Clustersenderkanal (CLUSSDR)**

Ein Kanal, über den Clusterwarteschlangenmanager Nachrichten an andere Warteschlangenmanager im Cluster sowie Clusterinformationen an die Repository-Warteschlangenmanager senden können.

**Cluster-Topic**

Ein Verwaltungsthema (Topic), das in einem Clusterwarteschlangenmanager definiert ist und anderen Warteschlangenmanagern im Cluster zur Verfügung gestellt wird.

**Clusterübertragungswarteschlange**

Eine Übertragungswarteschlange, die alle Nachrichten eines Warteschlangenmanagers enthält, die für einen anderen Warteschlangenmanager in demselben Cluster bestimmt sind. Die Warteschlange heißt SYSTEM.CLUSTER.TRANSMIT.QUEUE.

**CMS-Schlüsseldatenbank**

Eine CMS-Schlüsseldatenbank ist das Format der Datenbank, die von Windows-Systemen, UNIX-Systemen, Linux und den Clients dieser Plattformen unterstützt wird. Dateien mit der Erweiterung .kdb haben das CMS-Format. Die .kdb-Dateien enthalten die Zertifikate und die Schlüssel.

**ID des codierten Zeichensatzes**

Eine 16-Bit-Zahl, die einen bestimmten Satz an Schemakennungen für Codeumsetzung, Zeichensatzkennungen, Kennungen für Codepages sowie andere Informationen enthält, die die Darstellung der codierten Grafikzeichen eindeutig kennzeichnen.

**Koexistenz**

Der Zustand, in dem mindestens zwei unterschiedliche Versionen von WebSphere MQ auf demselben Computer ausgeführt werden.

**Befehl**

Eine Anweisung, mit der eine Aktion eingeleitet oder ein Service gestartet wird. Ein Befehl besteht aus der Abkürzung des Befehlsnamens und ggf. zugehörigen Parametern und Attributen.

**Befehlsbehälter**

In WebSphere MQ Administration Interface ein Typ von Behälter, der für die Verwaltung von WebSphere-MQ-Objekten erstellt wird, aber weder die Änderung der Reihenfolge der Datenelemente noch die Erstellung von Listen innerhalb einer Nachricht zulässt.

**command event**

Eine Benachrichtigung über die erfolgreiche Ausführung eines WebSphere MQ-Scriptbefehls oder PCF-Befehls.

**Command Language (CL)**

In WebSphere MQ for iSeries eine Sprache, mit deren Hilfe Befehle entweder über die Befehlszeile oder durch Schreiben eines CL-Programms ausgegeben werden können.

**Befehlspräfix**

1. Eine ein- bis achtstellige Befehlskennung. Das Befehlspräfix kennzeichnet den Befehl als einen Befehl, der zu einer Anwendung oder einem Subsystem gehört, nicht zu z/OS.
2. In WebSphere MQ for z/OS ist dies eine Zeichenfolge, die den Warteschlangenmanager angibt, an den WebSphere MQ for z/OS-Befehle übertragen und von dem WebSphere MQ for z/OS-Bedienernachrichten empfangen werden.

**Befehlsserver**

Die WebSphere MQ-Komponente, die Befehle aus der Eingabewarteschlange für Systembefehle liest, überprüft und gültige Befehle an den Befehlsprozessor weiterleitet.

**Commit**

Das Anwenden aller Änderungen, die während der aktuellen Arbeitseinheit mit Wiederherstellung (UR) oder aktuellen Arbeitseinheit (UOW) vorgenommen wurden. Nach Abschluss der Operation kann eine neue UR oder UOW beginnen.

**Allgemeiner Name (Common Name, CN)**

Die Komponente in einem DN-Attribut (Distinguished Name) eines X.509-Zertifikats, die dem Namen entspricht, der dem Eigner des Zertifikats normalerweise zugewiesen ist. Bei Personen ist der CN üblicherweise deren tatsächlicher Name. Bei Web-Servern ist der CN der vollständig qualifizierte

Host- oder Domänenname des Servers. WebSphere MQ stellt an dieses Feld keine besonderen Anforderungen, aber viele Administratoren verwenden den Namen des Warteschlangenmanagers.

Siehe auch [Registrierter Name](#)

### **Beendigungscode**

Ein Rückkehrcode, der anzeigt, wie ein MQI-Aufruf beendet wurde.

### **Vertraulichkeit**

Der Sicherheitsservice, der sensible Informationen vor nicht autorisierter Offenlegung schützt. Ein typischer Mechanismus zum Implementieren dieses Service ist die Verschlüsselung.

### **Konfigurationsereignis**

Benachrichtigungen über die Attribute eines Objekts. Die Benachrichtigungen werden beim Erstellen, Ändern und Löschen eines Objekts und auch auf ausdrückliche Anforderung hin generiert.

### **Verbindungsaffinität**

Ein Kanalattribut, das die Clientkanaldefinition angibt, die von Clientanwendungen für die Verbindung zum Warteschlangenmanager verwendet wird, falls mehrere Verbindungen verfügbar sind.

### **Verbindungsfactory**

Eine Gruppe von Konfigurationswerten zum Erstellen von Verbindungen, über die eine Java EE-Komponente auf eine Ressource zugreifen kann. Verbindungsfactorys stellen bedarfsgesteuerte Verbindungen zwischen einer Anwendung und einem unternehmensweiten Informationssystem (EIS) bereit und ermöglichen einem Anwendungsserver, das EIS in einer verteilten Transaktion zu registrieren.

### **Verbindungskennung**

Die Kennung oder das Token, mit der bzw. dem ein Programm auf den Warteschlangenmanager zugreift, mit dem es verbunden ist.

### **Konstruktor**

In der objektorientierten Programmierung eine spezielle Methode zur Initialisierung eines Objekts.

### **Verarbeiten**

Das Entfernen einer Nachricht aus einer Warteschlange und die Rückgabe ihres Inhalts an die aufrufende Anwendung.

### **Nutzer**

Eine Anwendung, die Nachrichten empfängt und verarbeitet. Siehe auch [Nachrichtenkonsument](#).

### **Kontextsicherheit**

Unter z/OS sind dies die Berechtigungsprüfungen, die durchgeführt werden, wenn eine Anwendung eine Warteschlange öffnet und angibt, dass sie den Kontext in Nachrichten, die sie in die Warteschlange einreicht, festlegen wird oder dass sie den Kontext aus Nachrichten, die sie empfangen hat, an Nachrichten übergeben wird, die sie in die Warteschlange einreicht.

### **Steuerbefehl**

In WebSphere MQ auf UNIX- und Linux -Systemen und WebSphere MQ for Windows ein Befehl, der interaktiv über die Befehlszeile des Betriebssystems eingegeben werden kann. Einzige Voraussetzung für diese Befehle ist, dass das Produkt WebSphere MQ installiert ist. Darüber hinaus sind keine besonderen Dienstprogramme zur Ausführung dieser Befehle erforderlich.

### **Steuerintervall**

Ein Bereich mit fester Länge des Direktzugriffsspeichers, in dem VSAM Datensätze speichert und verteilten freien Speicherbereich erstellt. Das Steuerintervall ist die Informationseinheit, die VSAM an den bzw. aus dem Direktzugriffsspeicher überträgt. Ein Steuerintervall enthält immer eine integrale Anzahl physischer Datensätze.

### **Kontrolliertes Beenden**

Siehe [Gesteuerter Abschluss](#).

### **Korrelations-ID**

Ein Feld in einer Nachricht, das zur Identifizierung zusammengehöriger Nachrichten dient. Korrelations-IDs werden zum Beispiel verwendet, um Anforderungsnachrichten der entsprechenden Antwortnachricht zuzuordnen.



**Coupling Facility (CF)**

Eine spezielle logische Partition, die Zwischenspeicherung in Hochgeschwindigkeit, Listenverarbeitung und Sperrfunktionen in einem Sysplex bereitstellt.

**CPF**

Siehe Befehlspräfix.

**Zertifikatsanforderung**

Synonym für Zertifikatssignieranforderung.

**CRL**

Siehe Zertifikatswiderrufsliste.

**Cross-System Coupling-Facility (XCF)**

Eine Komponente, die Funktionen zur Unterstützung der Zusammenarbeit zwischen berechtigten Programmen, die in einem Sysplex ausgeführt werden, bereitstellt.

**Kryptografie**

Das Schützen von Informationen durch Umwandlung in ein nicht lesbares Format (Verschlüsselung), den so genannten verschlüsselten Text. Nur wer einen geheimen Schlüssel besitzt, kann die Nachricht entschlüsseln (in unverschlüsseltem Text umwandeln).

**D****DAE**

Siehe Dump Analysis and Elimination.

**daemon**

Ein automatisch ablaufendes Programm, das fortlaufende oder periodische Funktionen ausführt, z. B. zur Netzsteuerung.

**Datenbehälter**

Ein Behälter für Objekteigenschaften, die WebSphere MQ Administration Interface (MQAI) zur Verwaltung von Warteschlangenmanagern verwendet. Es gibt drei Typen von Datenbehältern: Benutzer (für Benutzerdaten), Verwaltung (für die Verwaltung mit angenommenen Optionen) und Befehl (für die Verwaltung ohne angenommene Optionen).

**Datenkonvertierungsschnittstelle**

Die WebSphere-MQ-Schnittstelle, der von Kunden oder Anbietern geschriebene Programme, die Anwendungsdaten in verschiedene Maschinencodierungen konvertieren, und CCSIDs entsprechen müssen. Ein Teil des WebSphere MQ-Frameworks.

**Datenkonvertierungsservice**

Ein Service, der Anwendungsdaten in den Zeichensatz und die Codierung konvertiert, die von Anwendungen auf anderen Plattformen gefordert werden.

**Datagramm**

Eine Form der asynchronen Nachrichtenübermittlung, bei der eine Anwendung eine Nachricht sendet, jedoch keine Antwort benötigt. Siehe auch Anforderung/Antwort.

**Datenintegrität**

Ein Sicherheitsservice, der erkennt, ob während der Übertragung eine unbefugte Datenänderung oder Manipulation von Daten vorgenommen wurde. Der Service erkennt nur, ob Daten geändert wurde, er stellt die Daten nicht in ihrem ursprünglichen Zustand wieder her, falls sie geändert wurden.

**Datenelement**

In WebSphere MQ Administration Interface ein Element, das in einem Datenbehälter enthalten ist. Dabei kann es sich um eine ganze Zahl oder eine Zeichenfolge und um ein Benutzerelement oder ein Systemelement handeln.

**DCE**

Siehe Distributed Computing Environment.

**DCE-Principal**

Eine Benutzer-ID, die die Umgebung für verteilte Datenverarbeitung verwendet.

**DCI (Data-Conversion Interface)**

Siehe Datenkonvertierungsschnittstelle.

## **DCM**

Siehe [Digital Certificate Manager](#).

## **Warteschlange für nicht zustellbare Nachrichten**

Eine Warteschlange, an die ein Warteschlangenmanager oder eine Anwendung Nachrichten sendet, die nicht an ihre eigentliche Zieladresse zugestellt werden können.

## **Warteschlange für dead-letter, Steuerroutine**

Ein Dienstprogramm zur Überwachung von Warteschlangen für nicht zustellbare Nachrichten und zur Verarbeitung von Nachrichten in der Warteschlange anhand einer benutzerdefinierten Regeltabelle. WebSphere MQ stellt eine Beispielsteyerroutine der Warteschlange für nicht zustellbare Nachrichten bereit.

## **Entschlüsselung**

Der Prozess, bei dem Daten, die in ein geheimes Format verschlüsselt wurden, entschlüsselt werden. Für die Entschlüsselung ist ein geheimer Schlüssel oder ein Kennwort erforderlich.

## **Standardobjekt**

Die Definition eines Objekts (z. B. einer Warteschlange), in der alle Attribute angegeben sind. Wenn ein Benutzer ein Objekt definiert, aber nicht alle möglichen Attribute für dieses Objekt angibt, verwendet der Warteschlangenmanager Standardattribute für die nicht angegebenen Attribute.

## **Verzögerte Verbindung**

Ein anstehendes Ereignis, das aktiviert wird, wenn ein CICS-Subsystem versucht, eine Verbindung mit WebSphere MQ for z/OS herzustellen, bevor es gestartet wurde.

## **Ableitung**

In der objektorientierten Programmierung die Verbesserung oder Erweiterung einer Klasse auf Basis einer anderen Klasse.

## **Ziel**

1. In JMS ein Objekt, das angibt, wohin Nachrichten gesendet und wie sie empfangen werden sollen.
2. Ein Endpunkt, an den Nachrichten gesendet werden, z. B. eine Warteschlange oder ein Thema.

## **Diffie-Hellman-Schlüsselaustausch**

Ein öffentlicher Schlüsselaustauschalgorithmus, der zur sicheren Erstellung eines geheimen Schlüssels über einen nicht sicheren Kanal verwendet wird.

## **Digitales Zertifikat**

Ein elektronisches Dokument, das zur Identifizierung einer Person, eines Systems, eines Servers, eines Unternehmens oder einer anderen Entität verwendet wird und der Entität einen öffentlichen Schlüssel zuordnet. Ein digitales Zertifikat wird von einer Zertifizierungsstelle ausgestellt und enthält eine digitale Signatur dieser Stelle.

## **Digital Certificate Manager (DCM)**

Ermöglicht auf IBM i-Systemen die Verwaltung von digitalen Zertifikaten und deren Verwendung in sicheren Anwendungen auf dem iSeries-Server. Digital Certificate Manager fordert digitale Zertifikate von Zertifizierungsstellen oder anderen Parteien an und verarbeitet sie.

## **Digitale Signatur**

Informationen, die mit einem privaten Schlüssel verschlüsselt und an eine Nachricht oder ein Objekt angefügt werden, um gegenüber dem Empfänger die Authentizität und Integrität der Nachricht oder des Objekts zu belegen. Die digitale Signatur belegt, dass die Nachricht oder das Objekt von der Entität signiert wurde, die Eigner des verwendeten privaten oder geheimen symmetrischen Schlüssels ist oder eine Zugriffsberechtigung für diesen Schlüssel besitzt.

## **Trennen**

Die Trennung der Verbindung zwischen einer Anwendung und einem Warteschlangenmanager.

## **Registrierter Name (Distinguished Name, DN)**

Eine Gruppe aus Name/Wert-Paaren (z. B. CN= Personennamen und C=Land), die eine Entität in einem digitalen Zertifikat eindeutig identifiziert. Der registrierte Name ist allerdings nur innerhalb des Namensbereichs einer angegebenen Zertifizierungsstelle eindeutig. Es ist absolut möglich, dass Zertifikate mit identischen registrierten Namen von verschiedenen Zertifizierungsstellen ausgestellt werden. Stellen Sie deshalb sicher, dass ein Schlüsselrepository so wenig vertrauenswürdige Rootzer-

tifizierungsstellenzertifikate wie möglich enthält, am besten nicht mehr als eins. Siehe auch [Zertifizierungsstelle](#), [Digitales Zertifikat](#) und [X509](#).

### **Verteilte Anwendung**

Beim Message-Queuing eine Gruppe von Anwendungsprogrammen, die einzeln jeweils mit verschiedenen Warteschlangenmanagern verbunden sein können, zusammen jedoch eine einzige Anwendung umfassen.

### **Distributed Computing Environment (DCE)**

Im Network-Computing eine Gruppe von Services und Tools, die die Erstellung, Verwendung und Wartung verteilter Anwendungen in heterogenen Betriebssystemen und Netzen unterstützen.

### **Verteiltes Warteschlangenmanagement**

Beim Message-Queuing die Einrichtung und Steuerung von Nachrichtenkanälen zu Warteschlangenmanagern auf anderen Systemen.

### **Distribution List**

Ein Liste mit Warteschlangen, in die eine Nachricht mithilfe einer einzelnen Anweisung eingereiht werden kann.

### **DLQ (Dead-Letter Queue)**

Siehe [Warteschlange für nicht zustellbare Nachrichten](#).

### **DN (Distinguished Name)**

Siehe [Registrierter Name](#).

### **Doppelprotokollierung**

Ein Verfahren zum Aufzeichnen von Aktivitäten in WebSphere MQ for z/OS, bei dem jede Änderung in zwei Dateien aufgezeichnet wird. Ist eine dieser Dateien bei einem erforderlichen Wiederanlauf nicht lesbar, kann auf die andere zurückgegriffen werden. Siehe auch [Einfache Protokollierung](#).

### **Dualmodus**

Siehe [Doppelte Protokollierung](#).

### **Dump Analysis and Elimination (DAE)**

Ein z/OS-Service, mit dessen Hilfe eine Installation unnötige SVC- und ABEND SYSUDUMP-Speicherauszüge unterdrücken kann, wenn diese mit bereits ausgegebenen Speicherausügen identisch sind.

### **Permanente Subskription**

Eine Subskription, die beibehalten wird, wenn die Verbindung zwischen einer Subskribentenanwendung und einem Warteschlangenmanager geschlossen wird. Wenn die Verbindung einer Subskribentenanwendung getrennt wird, bleibt die permanente Subskription erhalten und Veröffentlichungen werden weiterhin zugestellt. Wenn die Verbindung der Anwendung wiederhergestellt wird, kann die Anwendung dieselbe Subskription verwenden, indem sie den eindeutigen Subskriptionsnamen angibt. Siehe auch [Nicht permanente Subskription](#).

### **Dynamische Warteschlange**

Eine lokale Warteschlange, die beim Öffnen eines Modellwarteschlangenobjekts durch ein Programm erstellt wird.

## **E**

### **Abhören**

Ein Verstoß gegen die Kommunikationssicherheit, bei dem die Daten zwar unbeschädigt bleiben, aber ihre Vertraulichkeit beeinträchtigt wird. Siehe auch [Identitätsvortäuschung](#) und [Verfälschung](#).

### **Eclipse**

Eine Open-Source-Initiative, die unabhängigen Softwareanbietern (ISVs) und anderen Toolentwicklern eine Standardplattform zur Entwicklung kompatibler Plug-in-Tools für die Anwendungsentwicklung ermöglicht.

### **Kapselung**

In der objektorientierten Programmierung die Technik, mit deren Hilfe die inhärenten Details eines Objekts, einer Funktion oder einer Klasse vor Clientprogrammen verborgen werden.

## **Verschlüsselung**

In der IT-Sicherheit die Umsetzung von Daten in ein nicht lesbares Format, sodass die ursprünglichen Daten gar nicht oder nur durch einen Entschlüsselungsprozess erhalten werden können.

## **In Warteschlange stellen**

Das Einreihen einer Nachricht oder eines Elements in eine Warteschlange.

## **Entität**

Ein Benutzer, eine Gruppe oder eine Ressource, der/die in einem Sicherheitservice, z. B. RACF, definiert ist.

## **Umgebungsvariable**

Eine Variable, mit der angegeben wird, wie ein Betriebssystem oder ein anderes Programm ausgeführt wird bzw. welche Einheiten das Betriebssystem erkennt.

## **ESM**

Siehe [Externer Sicherheitsmanager](#).

## **ESTAE**

Siehe [Extended Specify Task Abnormal Exit](#).

## **Ereignisdaten**

In einer Ereignisnachricht der Teil der Nachrichtendaten, der Informationen zum Ereignis enthält, z. B. den Namen des Warteschlangenmanagers und der Anwendung, die das Ereignis ausgelöst hat. Siehe auch [Ereignisheader](#).

## **Ereignisheader**

In einer Ereignisnachricht der Teil der Nachrichtendaten, der den Ereignistyp des Ursachencodes für das Ereignis angibt. Siehe auch [Ereignisdaten](#).

## **Ereignisnachricht**

Eine Nachricht, die Informationen (z. B. die Ereigniskategorie, den Namen der Anwendung, die das Ereignis ausgelöst hat, oder Statistiken des Warteschlangenmanagers) enthält, die sich auf den Ursprung eines Instrumentierungsereignisses in einem Netz aus WebSphere-MQ-Systemen beziehen.

## **Ereigniswarteschlange**

Die Warteschlange, in die der Warteschlangenmanager beim Erkennen eines Ereignisses eine entsprechende Ereignisnachricht stellt. Für jede Ereigniskategorie (Warteschlangenmanager-, Leistungs-, Konfigurations-, Instrumentierungs- oder Kanalereignis) gibt es eine eigene Ereigniswarteschlange.

## **Ereignisanzeige**

Ein Tool auf Windows-Systemen, mit dem Protokolldateien überprüft und verwaltet werden können.

## **Listener für Ausnahmebedingungen**

Eine Instanz einer Klasse, die von einer Anwendung registriert werden kann und für die die Methode `onException()` aufgerufen wird, um eine JMS-Ausnahme asynchron an die Anwendung zu übergeben.

## **Exklusive Methode**

In der objektorientierten Programmierung eine Methode, die keine Polymorphie unterstützt, d. h. eine Methode mit einem bestimmten Effekt.

## **Extended Specify Task Abnormal Exit (ESTAE)**

Ein z/OS-Makro, das Wiederherstellungsfunktionen bereitstellt und die Steuerung an eine benutzerdefinierte Exitroutine übergibt, um eine abnormale Beendigung zu verarbeiten und zu diagnostizieren oder um eine Wiederholungsadresse anzugeben.

## **Externem Sicherheitsmanager (ESM)**

Ein Sicherheitsprodukt, das Sicherheitsprüfungen für Benutzer und Ressourcen durchführt. RACF ist ein Beispiel für einen externen Sicherheitsmanager.

# **F**

## **Failover**

Ein automatischer Vorgang, bei dem im Falle einer Software-, Hardware- oder Netzstörung auf ein redundantes oder Standby-System umgeschaltet wird.

## **FAP**

Siehe [Formate und Protokolle](#).

**FFDC (First-Failure Data Capture)**

Siehe [Erfassung von Fehlerdaten beim ersten Auftreten](#).

**FFST**

Siehe [First Failure Support Technology](#).

**FFST-Datei**

See [First Failure Support Technology-Datei](#).

**FIFO**

Siehe [First In/First Out](#).

**FIPS**

Federal Information Processing Standard (USA)

**Erfassung von Fehlerdaten beim ersten Auftreten (FFDC)**

1. Eine Hilfe zur Problemdiagnose, die Fehler ermittelt, Informationen zu diesen Fehlern sammelt und protokolliert und die Steuerung an die beteiligte Laufzeitssoftware zurückgibt.
2. Die IBM i-Implementierung der FFST-Architektur, die Problemerkennung, selektive Speicherauszüge von Diagnosedaten, Generierung von Symptomzeichenfolgen und Problemprotokolleinträge bereitstellt.

**First Failure Support Technology (FFST)**

Eine IBM Architektur, die einen einheitlichen Ansatz zur Fehlererkennung über defensive Programmierertechniken definiert. Diese Techniken ermöglichen eine proaktive Problemerkennung (passiv, bis Aktion erforderlich) und eine Beschreibung der Diagnosenachrichten, die zur Behebung eines Softwareproblems erforderlich sind.

**First Failure Support Technology-Datei (FFST-Datei)**

Eine Datei, die Informationen zur Erkennung und Diagnose von Softwareproblemen enthält. In WebSphere MQ haben FFST-Dateien den Dateityp FDC.

**First In/First Out (FIFO)**

Ein Warteschlangensteuerungsverfahren, bei dem als nächstes Element das Element abgerufen wird, das am längsten in der Warteschlange steht.

**Erzwungene Beendigung**

Die Beendigung eines CICS-Adapters, bei der die Verbindung zwischen Adapter und WebSphere MQ für z/OS sofort und ohne Rücksicht auf den Status der zu diesem Zeitpunkt aktiven Tasks getrennt wird. Siehe auch [Gesteuerter Abschluss](#).

**Format**

Beim Message-Queuing ein Begriff, der die Art von Anwendungsdaten in einer Nachricht angibt.

**Formate und Protokolle**

Beim Message-Queuing eine Definition, die festlegt, wie Warteschlangenmanager miteinander kommunizieren und wie Clients mit Warteschlangenmanagern des Servers kommunizieren.

**Framework**

In WebSphere MQ eine Sammlung von Programmierschnittstellen, die es Kunden oder Anbietern ermöglichen, Programme zu schreiben, die bestimmte, in WebSphere-MQ-Produkten bereitgestellte Funktionen erweitern oder ersetzen. Dabei handelt es sich um folgende Schnittstellen: Datenkonvertierungsschnittstelle, Nachrichtenkanalschnittstelle, Namensserviceschnittstelle, Schnittstelle für Sicherheitsaktivierung und Auslösemonitorschnittstelle.

**Friend-Klasse**

Eine Klasse, in der allen Elementfunktionen Zugriff auf die privaten und geschützten Elemente einer anderen Klasse erteilt wird. Sie wird in der Deklaration einer anderen Klasse angegeben und verwendet das Schlüsselwort 'friend' als Präfix für die Klasse.

**FRR (Functional Recovery Routine)**

Siehe [Funktionswiederherstellungsroutine](#).

### **Vollständiges Repository**

Vollständige Informationen zu allen Warteschlangenmanagern in einem Cluster. Diese Informationen werden als Repository oder manchmal auch als vollständiges Repository bezeichnet und werden normalerweise von zwei der Warteschlangenmanager im Cluster verwaltet. Siehe auch [Teilrepository](#).

### **function**

Eine benannte Gruppe von Anweisungen, die aufgerufen und ausgewertet werden kann und einen Wert an die aufrufende Anweisung zurückgeben kann.

### **Funktionswiederherstellungsroutine**

Ein z/OS-Manager zur Wiederherstellung und Beendigung, der es einer Wiederherstellungsroutine ermöglicht, im Falle einer Programmunterbrechung die Steuerung zu übernehmen.

## **G**

### **Gateway-Warteschlangenmanager**

Ein Clusterwarteschlangenmanager, mit dem Nachrichten von einer Anwendung an andere Warteschlangenmanager im Cluster weitergeleitet werden.

### **Allgemeine Tracefunktion**

Ein z/OS-Serviceprogramm, das wichtige Systemereignisse wie E/A-Interrupts, SVC-Interrupts, Programm-Interrupts und externe Interrupts aufzeichnet.

### **Generic Security Services-API**

Siehe [Generic Security Services-Anwendungsprogrammierschnittstelle](#).

### **Generic Security Services-Anwendungsprogrammierschnittstelle (Generic Security Services-API, GSS-API)**

Eine allgemeine Anwendungsprogrammierschnittstelle (API) für den Zugriff auf Sicherheitsservices.

### **Abrufen**

Beim Message-Queuing die Verwendung des MQGET-Aufrufs, um eine Nachricht aus einer Warteschlange zu entfernen und ihren Inhalt an die aufrufende Anwendung zurückzugeben. Siehe auch [Durchsuchen](#) und [Einreihen](#).

### **Global definiertes Objekt**

In z/OS ein Objekt, dessen Definition im gemeinsamen Repository gespeichert wird. Das Objekt steht allen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange zur Verfügung. Siehe auch [Lokal definiertes Objekt](#).

### **globaler Trace**

Eine Traceoption in WebSphere MQ for z/OS, bei der Tracedaten aus dem gesamten WebSphere MQ for z/OS-Subsystem stammen.

### **globale Transaktion**

Eine wiederherstellbare Arbeitseinheit, die von mindestens einem Ressourcenmanager in einer verteilten Transaktionsumgebung ausgeführt und von einem externen Transaktionsmanager koordiniert wird.

### **GSS-API**

Siehe [Generic Security Services-Anwendungsprogrammierschnittstelle](#).

### **GTF**

Siehe [Allgemeine Tracefunktion](#).

## **H**

### **Handshake**

Der Austausch von Nachrichten beim Start einer SSL-Sitzung (Secure Sockets Layer), der es dem Client ermöglicht, den Server mithilfe öffentlicher Schlüssel zu authentifizieren (und optional auch umgekehrt). Anschließend können Client und Server bei der Erstellung symmetrischer Schlüssel zur Verschlüsselung und Entschlüsselung sowie zur Erkennung von Verfälschungen zusammenarbeiten.

### **Permanent gespeicherte Nachricht**

Eine Nachricht, die in einen Zusatz(platten)speicher geschrieben wird, damit die Nachricht im Falle eines Systemausfalls nicht verloren geht.

**Header**

Siehe [Nachrichtenheader](#).

**heartbeat**

Ein Signal, das eine Entität an eine andere sendet, um anzuzeigen, dass sie noch aktiv ist.

**Austausch von Überwachungssignalen**

Ein Impuls, der von einem sendenden an einen empfangenden Nachrichtenkanalagenten übertragen wird, wenn keine Nachrichten zum Senden vorhanden sind. Der Impuls gibt den empfangenden Nachrichtenkanalagenten frei, der ansonsten im Wartestatus verbleiben würde, bis eine Nachricht eingeht oder das Intervall für Verbindungstrennung abläuft.

**Überwachungssignalintervall**

Die Zeit (in Sekunden) zwischen dem Austausch von Überwachungssignalen.

**Hierarchie**

In der Publish/Subscribe-Messaging-Topologie ein lokaler Warteschlangenmanager, der mit einem übergeordneten Warteschlangenmanager verbunden ist.

**HTTP**

Siehe [Hypertext Transfer Protocol](#).

**HTTP (Hypertext Transfer Protocol)**

Ein Internetprotokoll, das zum Übertragen und Anzeigen von Hypertext- und XML-Dokumenten im Web verwendet wird.

**I****Identitätskontext**

Informationen, die den Benutzer der Anwendung identifizieren, die die Nachricht zuerst in eine Warteschlange einreicht.

**Identifikation**

Der Sicherheitsservice, der die eindeutige Identifikation jedes Benutzers eines Computersystems ermöglicht. Ein typisches Verfahren zum Implementieren dieses Service besteht darin, jedem Benutzer eine Benutzer-ID zuzuordnen.

**Identitätskontext**

Informationen, die den Benutzer der Anwendung identifizieren, die die Nachricht zuerst in eine Warteschlange einreicht.

**IFCID**

Siehe [Instrumentation Facility Component Identifier](#).

**ILE**

Siehe [Integrated Language Environment](#).

**Sofortige Beendigung**

In WebSphere MQ das Beenden eines Warteschlangenmanagers, ohne eine Verbindungsunterbrechung der Anwendungen abzuwarten. Aktuelle MQI-Aufrufe können abgeschlossen werden, neue MQI-Aufrufe hingegen können bei Anforderung eines sofortigen Abschlusses nicht ausgeführt werden. Siehe auch [Präventiver Abschluss](#) und [Gesteuerter Abschluss](#).

**impersonation**

Ein Verstoß gegen die Kommunikationssicherheit, bei dem die Daten an eine Person übergeben werden, die sich als vermeintlicher Empfänger ausgibt, oder Daten von einer Person gesendet werden, die sich als jemand anders ausgibt. Siehe auch [Abhören](#), [Verfälschung](#).

**Eingehender Kanal**

Ein Kanal, der Nachrichten von einem anderen Warteschlangenmanager empfängt.

**Eingebundenes Format**

Siehe [Integriertes Format](#).

**Index**

In WebSphere MQ Administration Interface (MQAI) ein Verfahren zum Verweisen auf Datenelemente.

**Unbestätigte Arbeitseinheit mit Wiederherstellung**

Der Status einer Arbeitseinheit mit Wiederherstellung, für die ein Synchronisationspunkt angefordert, aber noch nicht bestätigt wurde.

**Unvollständig**

Der Status einer Ressource oder Arbeitseinheit mit Wiederherstellung, die die Vorbereitungsphase des Commitprozesses noch nicht abgeschlossen hat.

**Vererbung**

Ein Verfahren in der objektorientierten Programmierung, bei der vorhandene Klassen als Basis für die Erstellung weiterer Klassen verwendet werden. Durch Vererbung übernehmen spezifischere Elemente die Struktur und das Verhalten allgemeinerer Elemente.

**Initialisierungseingabedatei**

Eine von WebSphere MQ for z/OS beim Start verwendete Datei.

**Initialisierungswarteschlange**

Eine lokale Warteschlange, in die der Warteschlangenmanager Auslösenachrichten stellt.

**Initiator**

Bei der verteilten Warteschlangensteuerung ein Programm, das Netzverbindungen auf einem anderen System anfordert. Siehe auch [Responder](#).

**Eingabeparameter**

Ein Parameter eines MQI-Aufrufs, in dem beim Absetzen des Aufrufs Informationen angegeben werden.

**Einfügereihenfolge**

In WebSphere MQ Administration Interface (MQAI) die Reihenfolge, in der Datenelemente in einem Datenbehälter abgelegt werden.

**installierbarer Service**

In WebSphere MQ auf UNIX- und Linux -Systemen und WebSphere MQ for Windows wird zusätzliche Funktionalität als unabhängige Komponente bereitgestellt. Die Installation dieser Komponenten ist optional; stattdessen können auch eigene Komponenten oder Komponenten anderer Hersteller verwendet werden.

**Instanz**

Ein bestimmtes Vorkommen eines Objekts, das zu einer Klasse gehört. Siehe auch [Objekt](#).

**Instanzdaten**

In der objektorientierten Programmierung Statusinformationen, die einem Objekt zugeordnet sind.

**Instrumentierungsereignis**

Eine Methode zur Überwachung von Warteschlangenmanager-Ressourcendefinitionen, Ausführungsbedingungen und Kanalbedingungen in einem Netz aus WebSphere-MQ-Systemen.

**Instrumentation Facility Component Identifier (IFCID)**

In Db2 for z/OS ein Wert, mit dem ein Tracesatz eines Ereignisses benannt und identifiziert wird.

Als Parameter in den Befehlen 'START TRACE' und 'MODIFY TRACE' gibt dieser Wert an, dass für die entsprechenden Ereignisse ein Trace ausgeführt werden soll.

**Integrated Language Environment (ILE)**

Eine Gruppe von Konstrukten und Schnittstellen, die eine gemeinsame Laufzeitumgebung und zur Laufzeit bindungsfähige Anwendungsprogrammierschnittstellen (APIs) für alle ILE-konformen höheren Sprachen bereitstellt.

**Interactive Problem Control System (IPCS)**

Eine Komponente in MVS und z/OS, die Onlineproblemmanagement, interaktive Problemdiagnose, Onlinefehlerbehebung für plattenresidente Speicherauszüge nach einem Absturz, Fehlerverfolgung und Problemmeldung ermöglicht.

**Interactive System Productivity Facility (ISPF)**

Ein IBM Lizenzprogramm, das als Gesamtanzeigeditor und Dialogmanager eingesetzt wird. Das Programm wird zum Schreiben von Anwendungsprogrammen verwendet und ermöglicht die Erstellung standardmäßiger Bildschirmanzeigen und interaktiver Dialoge zwischen dem Anwendungsprogrammierer und dem Terminalbenutzer.



**Zwischenzertifikat**

Ein Unterzeichnerzertifikat, bei dem es sich nicht um das Stammzertifikat handelt.

**Interface**

In der objektorientierten Programmierung ein abstraktes Verhaltensmodell; eine Gruppe von Funktionen oder Methoden.

**Internetprotokoll (IP)**

Ein Protokoll, das Daten in einem Netz oder in miteinander verbundenen Netzen weiterleitet. Dieses Protokoll fungiert als Vermittler zwischen den höheren Protokollschichten und dem physischen Netz. Siehe auch [Transmission Control Protocol](#).

**Interprozesskommunikation (IPC)**

Der Prozess, bei dem Programme sich einander Nachrichten senden. Sockets, Semaphoren, Signale und interne Nachrichtenwarteschlangen sind typische Methoden der Interprozesskommunikation. Siehe auch [Client](#).

**Intersystem Communication (ISC)**

Eine CICS-Funktion, die die eingehende und abgehende Kommunikation mit anderen Computersystemen unterstützt.

**IP**

Siehe [Internet Protocol](#).

**IPC (Interprocess Communication)**

Siehe [Interprozesskommunikation](#).

**IPCS**

Siehe [Interactive Problem Control System](#).

**ISC**

Siehe [Intersystem Communication](#).

**ISPF**

Siehe [Interactive System Productivity Facility](#).

**J****JAAS**

Siehe [Java Authentication and Authorization Service](#).

**Java Authentication and Authorization Service (JAAS)**

In der Java EE-Technologie eine Standard-API zur Durchführung sicherheitsbasierter Operationen. Mit JAAS können Services Benutzer authentifizieren und berechtigen und es den Anwendungen gleichzeitig ermöglichen, weiterhin unabhängig von zugrunde liegenden Technologien zu sein.

**Java Message Service (JMS)**

Eine Anwendungsprogrammierschnittstelle, die Java-Funktionen für die Verarbeitung von Nachrichten bereitstellt. Siehe auch [Message Queue Interface](#).

**Java Runtime Environment (JRE)**

Eine Untergruppe eines Java-Entwicklerkits, das die zentralen ausführbaren Programme und Dateien enthält, die die Java-Standardplattform bilden. Die JRE umfasst die Java Virtual Machine (JVM), Kernklassen und Unterstützungsdateien.

**JMS**

Siehe [Java Message Service](#).

**JMSAdmin**

Ein Verwaltungstool, mit dem Administratoren die Eigenschaften von JMS-Objekten definieren und in einem JNDI-Namensbereich speichern können.

**Journal**

Eine Funktion von OS/400, die von WebSphere MQ for iSeries zur Steuerung von Aktualisierungen lokaler Objekte verwendet wird. Jede Warteschlangenmanagerbibliothek enthält ein Journal für den jeweiligen Warteschlangenmanager.

## JRE

Siehe [Java Runtime Environment](#).

## K

### Keepalive-Paket

Ein TCP/IP-Mechanismus, bei dem ein kleines Paket in vordefinierten Intervallen im Netz versendet wird, um festzustellen, ob das Socket noch ordnungsgemäß funktioniert.

### Kerberos

Ein Netzauthentifizierungsprotokoll, das auf symmetrischer Verschlüsselung basiert. Kerberos weist jedem Benutzer, der sich am Netz anmeldet, einen eindeutigen Schlüssel (ein so genanntes Ticket) zu. Das Ticket wird in Nachrichten eingebettet, die über das Netz gesendet werden. Der Empfänger einer Nachricht verwendet das Ticket, um den Sender zu authentifizieren.

### Schlüsselauthentifizierung

Siehe [Authentifizierung](#).

### Schlüsselrepository

Ein allgemeiner Begriff für einen Speicher für digitale Zertifikate und die zugehörigen Schlüssel. Es gibt verschiedene Typen von Schlüsselrepositories, z. B. Certificate Management System (CMS), Java Keystore (JKS), Java Cryptography Extension Keystore (JCEKS), Public Key Cryptography Standard 12 (PKCS12) Keystore und RACF-Schlüsselringe. Falls es wichtig ist, zwischen Schlüsselrepository-Typen zu unterscheiden, ist in der Dokumentation ausdrücklich der Name des Schlüsselrepository-Typs angegeben. In Kontexten, die für mehrere Schlüsselrepository-Typen gültig sind, wird der allgemeine Begriff Schlüsselrepository verwendet.

### Schlüsselring

In der IT-Sicherheit eine Datei, die öffentliche Schlüssel, private Schlüssel, Trusted Roots und Zertifikate enthält.

### Schlüsselspeicher

Der Ort für einen privaten Schlüssel und das entsprechende persönliche Zertifikat. Siehe auch [Truststore](#)

## L

### Last Will and Testament

Ein Objekt, das von einem Client mit einem Monitor registriert und vom Monitor verwendet wird, falls der Client unerwartet beendet wird.

### LDAP-

Siehe [Lightweight Directory Access Protocol](#).

### Lightweight Directory Access Protocol (LDAP)

Ein offenes Protokoll, das über TCP/IP Zugriff auf Verzeichnisse bereitstellt, die ein X.500-Modell unterstützen, und das nicht den Ressourcenbedarf des komplexeren X.500 Directory Access Protocol (DAP) beansprucht. Über LDAP kann beispielsweise in einem Internet- oder Intranetverzeichnis nach Personen, Organisationen und anderen Ressourcen gesucht werden.

### Lineare Protokollierung

In WebSphere MQ auf UNIX- und Linux -Systemen und WebSphere MQ for Windows der Prozess der Aufbewahrung von Neustartdaten in einer Dateifolge. Je nach Bedarf werden neue Dateien erstellt. Der Speicherbereich, in den die Daten geschrieben werden, wird nicht wiederverwendet. Siehe auch [Umlaufprotokollierung](#).

### Sicherheit auf Verbindungsebene

Die Sicherheitsservices, die von einem Nachrichtenkanalagenten (MCA), vom Kommunikationssystem oder von einer Kombination dieser beiden direkt oder indirekt aufgerufen werden.

### Empfangsprogramm

Ein Programm, das eingehende Anforderungen erkennt und die zugeordneten Kanäle startet.

**Lokale Definition einer fernen Warteschlange**

Ein WebSphere MQ-Objekt, das einem lokalen Warteschlangenmanager zugeordnet ist und in dem die Attribute einer Warteschlange definiert sind, die einem anderen Warteschlangenmanager zugeordnet ist. Darüber hinaus wird dieses Objekt zur Aliasnamensumsetzung für Warteschlangenmanager und Warteschlangen für Antwortnachrichten verwendet.

**Ländereinstellung**

Eine Einstellung, die die Sprache oder Ländergruppe kennzeichnet und Formatierungskonventionen wie Sortierfolge, Wechsel zwischen Groß- und Kleinschreibung, Zeichenklassifikation, Nachrichtensprache, Datums- und Uhrzeitdarstellung und numerische Darstellung bestimmt.

**Lokal definiertes Objekt**

In z/OS ein Objekt, dessen Definition in der Seitengruppe 0 gespeichert ist. Auf diese Definition kann nur der Warteschlangenmanager zugreifen, der sie erstellt hat. Siehe auch [Global definiertes Objekt](#).

**Lokale Warteschlange**

Eine Warteschlange, die zum lokalen Warteschlangenmanager gehört. Eine lokale Warteschlange kann eine Liste der Nachrichten enthalten, die zur Verarbeitung anstehen. Siehe auch [Ferne Warteschlange](#).

**Lokaler Warteschlangenmanager**

Der Warteschlangenmanager, mit dem das Programm verbunden ist und der Message-Queuing-Services für das Programm bereitstellt. Siehe auch [Ferner Warteschlangenmanager](#).

**log**

In WebSphere MQ eine Datei, in der die von Warteschlangenmanagern beim Empfangen, Übertragen und Zustellen von Nachrichten ausgeführten Aktivitäten aufgezeichnet werden, um den Warteschlangenmanagern eine Wiederherstellung im Falle eines Fehlers zu ermöglichen.

**Protokollsteuerdatei**

In WebSphere MQ auf UNIX- und Linux-Systemen und WebSphere MQ for Windows die Datei mit Informationen, die zur Überwachung der Verwendung von Protokolldateien benötigt werden (z. B. Größe und Position der Protokolldateien sowie Name der nächsten verfügbaren Datei).

**Protokolldatei**

In WebSphere MQ auf UNIX-Systemen unter Linux und WebSphere MQ for Windows eine Datei, in der alle wichtigen Änderungen an den von einem Warteschlangenmanager gesteuerten Daten aufgezeichnet werden. Wenn die primären Protokolldateien voll sind, ordnet WebSphere MQ sekundäre Protokolldateien zu.

**Logische Einheit**

Ein Zugriffspunkt, über den ein Benutzer oder Anwendungsprogramm auf das SNA-Netz zugreift, um mit einem anderen Benutzer oder Anwendungsprogramm zu kommunizieren.

**Logische Einheit 6.2**

Eine logische SNA-Einheit, die die allgemeine Kommunikation zwischen Programmen in einer Umgebung für verteilte Verarbeitung unterstützt.

**ID der logischen Arbeitseinheit**

Ein Name, der einen Thread in einem Netz eindeutig identifiziert. Dieser Name besteht aus dem vollständig qualifizierten Netznamen einer logischen Einheit, der Instanznummer einer logischen Arbeitseinheit und der Folgenummer einer logischen Arbeitseinheit.

**Protokolleintrag**

Eine Gruppe von Daten, die in einer Protokolldatei als einzelne Einheit behandelt wird.

**Protokollsatzfolgenummer (LRSN, Log Record Sequence Number)**

Eine eindeutige Kennung für einen Protokolleintrag, der einem Benutzer, der Daten gemeinsam nutzt, zugeordnet ist. Db2 for z/OS verwendet die Protokollsatzfolgenummer zur Wiederherstellung in der Umgebung für gemeinsame Datennutzung.

**LRSN**

Siehe [Protokollsatzfolgenummer](#).

**LU (Logical Unit)**

Siehe [Logische Einheit](#).

## LU 6.2

Siehe [Logische Einheit 6.2](#).

### LU 6.2-Dialog

In SNA eine logische Verbindung zwischen zwei Transaktionsprogrammen über eine LU-6.2-Sitzung, die es ihnen ermöglicht, miteinander zu kommunizieren.

### Sicherheit auf LU 6.2-Dialogebene

In SNA ein Sicherheitsprotokoll auf Dialogebene, das es einem Partnertransaktionsprogramm ermöglicht, das Transaktionsprogramm zu authentifizieren, das den Dialog eingeleitet hat. Die Sicherheit auf LU 6.2-Dialogebene wird auch als Endbenutzerprüfung bezeichnet.

### LU 6.2-Sitzung

In SNA eine Sitzung zwischen zwei logischen Einheiten (LUs) des Typs 6.2.

### LU-Name

Der Name, mit dem VTAM auf einen Knoten im Netz verweist.

### LUID (Logical Unit of Work Identifier)

Siehe [ID der logischen Arbeitseinheit](#).

## M

### Verwaltetes Ziel

Eine Warteschlange, die vom Warteschlangenmanager für eine Anwendung, die sich für die Verwendung einer verwalteten Subskription entscheidet, als Ziel bereitgestellt wird, an das veröffentlichte Nachrichten gesendet werden sollen. Siehe auch [Verwaltete Subskription](#).

### Verwaltetes Handle

Eine Kennung, die vom MQSUB-Aufruf zurückgegeben wird, wenn ein Warteschlangenmanager für die Verwaltung des Speichers für Nachrichten, die an die Subskription gesendet werden, angegeben wird.

### Verwaltete Subskription

Eine Subskription, für die der Warteschlangenmanager eine Subskribentenwarteschlange zum Empfangen von Veröffentlichungen erstellt, weil für die Anwendung keine bestimmte Warteschlange verwendet werden muss. Siehe auch [Verwaltetes Ziel](#).

### Marshalling

Siehe [Serielle Verarbeitung](#).

### Nachrichtenkanalagent

Siehe [Nachrichtenkanalagent](#).

### MCI (Message Channel Interface)

Siehe [Nachrichtenkanalschnittstelle](#).

### Medienimage

In WebSphere MQ auf UNIX- und Linux -Systemen und WebSphere MQ for Windows die Folge von Protokollsätzen, die ein Image eines Objekts enthalten. Das Objekt kann auf der Basis dieses Images erneut erstellt werden.

### Nachricht

1. Eine Mitteilung, die von einer Person oder einem Programm an eine andere Person oder ein anderes Programm gesendet wird.
2. In der Systemprogrammierung eine Informationen für den Terminalbediener oder Systemadministrator.

### Nachrichtenaffinität

Die Beziehung zwischen Dialognachrichten, die zwischen zwei Anwendungen ausgetauscht werden, wobei die Nachrichten von einem bestimmten Warteschlangenmanager oder in einer bestimmten Reihenfolge verarbeitet werden müssen.

### Nachrichtenkanal

Bei der Steuerung der Warteschlangen für verteilte Nachrichten ein Mechanismus für das Verschieben von Nachrichten von einem Warteschlangenmanager zu einem anderen. Ein Nachrichtenkanal besteht

aus zwei Nachrichtenkanalagenten (einem Sender auf der einen und einem Empfänger auf der anderen Seite) und einer Kommunikationsverbindung. Siehe auch [Kanal](#).

### **Nachrichtenkanalagent (MCA)**

Ein Programm, das vorbereitete Nachrichten aus einer Übertragungswarteschlange an eine Kommunikationsverbindung oder von einer Kommunikationsverbindung an eine Zielwarteschlange überträgt. Siehe auch [Message Queue Interface](#).

### **Nachrichtenkanalschnittstelle**

Die WebSphere-MQ-Schnittstelle, der von Kunden oder Anbietern geschriebene Programme, die Nachrichten zwischen einem WebSphere-MQ-Warteschlangenmanager und einem anderen Messaging-System übertragen, entsprechen müssen. Ein Teil des WebSphere MQ-Frameworks. Siehe auch [Message Queue Interface](#).

### **Nachrichtenkonsument**

1. Ein Programm oder eine Funktion, das bzw. die Nachrichten abrufen und verarbeitet. Siehe auch [Konsument](#).
2. In JMS ein Objekt, das in einer Sitzung erstellt wird, um Nachrichten von einem Ziel zu empfangen.

### **Nachrichtenkontext**

Informationen zum Sender einer Nachricht, die in Feldern des Nachrichtendeskriptors enthalten sind. Es gibt zwei Kategorien von Kontextinformationen: Identitätskontext und Ursprungskontext.

### **Nachrichtendeskriptor**

Steuerinformationen, die das Nachrichtenformat und die Darstellung beschreiben und in einer WebSphere MQ-Nachricht übertragen werden. Das Format des Nachrichtendeskriptors wird durch die MQMD-Struktur definiert.

### **Nachrichtenexit**

Ein Kanalexitprogramm, mit dem der Inhalt einer Nachricht geändert wird. Nachrichtenexits arbeiten in der Regel paarweise, einer auf jeder Seite eines Kanals. Auf der Sendeseite eines Kanals wird ein Nachrichtenexit aufgerufen, nachdem der Nachrichtenkanalagent eine Nachricht aus einer Übertragungswarteschlange abgerufen hat. Auf der Empfangsseite eines Kanals wird ein Nachrichtenexit aufgerufen, bevor der Nachrichtenkanalagent eine Nachricht in seine Zielwarteschlange einreicht.

### **Nachrichtenflusssteuerung**

Eine Task für verteiltes Warteschlangenmanagement zum Konfigurieren und Verwalten von Nachrichtenrouten zwischen Warteschlangenmanagern.

### **Message Format Service (MFS)**

Eine IMS-Bearbeitungsfunktion, die es Anwendungsprogrammen ermöglicht, mit einfachen logischen Nachrichten statt mit geräteabhängigen Daten zu arbeiten. Dies vereinfacht den Anwendungsentwicklungsprozess.

### **Nachrichtengruppe**

Eine logische Gruppe zusammengehöriger Nachrichten. Die Beziehung wird von der Anwendung definiert, die die Nachrichten einreicht, und stellt sicher, dass die Nachrichten in der Reihenfolge abgerufen werden, in der sie eingereicht wurden, sofern sowohl Produzent als auch Konsument die Gruppierung berücksichtigen.

### **Nachrichtenhandle**

Ein Verweis auf eine Nachricht. Das Handle kann verwendet werden, um den Zugriff auf die Nachrichteneigenschaften der Nachricht zu erhalten.

### **Nachrichtenheader**

Der Teil einer Nachricht, der Steuerinformationen enthält, z. B. eine eindeutige Nachrichten-ID, Sender und Empfänger der Nachricht, Nachrichtenpriorität und Nachrichtentyp.

### **Nachrichteneingabedeskriptor**

Der MFS-Steuerblock (Message Format Service), der das Format der Daten beschreibt, die dem Anwendungsprogramm präsentiert werden. Siehe auch [Nachrichtenausgabedeskriptor](#).

### **Nachrichtenlistener**

Ein Objekt, das als asynchroner Nachrichtenkonsument agiert.

**Nachrichtenausgabedeskriptor**

Der MFS-Steuerblock (Message Format Service), der das Format der Ausgabedaten beschreibt, die vom Anwendungsprogramm erstellt werden. Siehe auch [Nachrichteneingabedeskriptor](#).

**Nachrichtenpriorität**

In WebSphere MQ ein Nachrichtenattribut, das sich auf die Abrufreihenfolge von Nachrichten aus einer Warteschlange auswirken kann und angibt, ob ein Auslöserereignis generiert wird.

**Nachrichtenproduzent**

In JMS ist dies ein Objekt, das von einer Sitzung erstellt wird, um Nachrichten an ein Ziel zu senden.

**Nachrichteneigenschaft**

Daten, die einer Nachricht in Form von Name/Wert-Paaren zugeordnet sind. Nachrichteneigenschaften können als Nachrichtenselektoren zum Filtern von Veröffentlichungen oder zum selektiven Abrufen aus Warteschlangen verwendet werden. Über Nachrichteneigenschaften können Geschäftsdaten oder Statusinformationen zur Verarbeitung eingeschlossen werden, ohne den Nachrichtenhauptteil ändern zu müssen.

**Message Queue Interface (MQI)**

Die Programmierschnittstelle, die von WebSphere MQ-Warteschlangenmanagern bereitgestellt wird. Über die Programmierschnittstelle können Anwendungsprogramme auf Message-Queuing-Services zugreifen. Siehe auch [Java Message Service](#), [Nachrichtenkanalagent](#), [Nachrichtenkanalschnittstelle](#).

**Message Queue Management (MQM)**

In WebSphere MQ for HP Integrity NonStop Server eine Funktion, die Zugriff auf PCF-Befehlsformate und Steuerbefehle zum Verwalten von Warteschlangenmanagern, Warteschlangen und Kanälen bereitstellt.

**Message-Queuing**

Eine Programmiertechnik, bei der jedes Programm in einer Anwendung mit den anderen Programmen kommuniziert, indem es Nachrichten in Warteschlangen einreicht.

**Nachrichtenwiederholung**

Eine Option für einen Nachrichtenkanalagenten, der nicht in der Lage ist, eine Nachricht einzureihen. Der Nachrichtenkanalagent kann eine vordefinierte Zeit warten und anschließend versuchen, die Nachricht erneut einzureihen.

**Nachrichtensegment**

Eines von mehreren Segmenten einer Nachricht, die so lang ist, dass sie entweder von der Anwendung oder vom Warteschlangenmanager nicht verarbeitet werden kann.

**Nachrichtenselektor**

In der Anwendungsprogrammierung ist dies eine Zeichenfolge variabler Länge, mit der sich eine Anwendung für den Abruf nur solcher Nachrichten registriert, deren Eigenschaften die als Auswahlzeichenfolge dargestellte SQL-Abfrage erfüllen. Die Syntax eines Nachrichtenselektors basiert auf einer Untergruppe der SQL92-Syntax für Bedingungsausdrücke.

**Nachrichtenfolgenummerierung**

Eine Programmiertechnik, bei der Nachrichten während der Übertragung über eine Kommunikationsverbindung eindeutige Nummern zugeordnet werden. Der empfangende Prozess kann anhand dieser Nummern feststellen, ob alle Nachrichten empfangen wurden, diese in ihrer ursprünglichen Reihenfolge in eine Warteschlange stellen und doppelt vorhandene Nachrichten löschen.

**Nachrichten-Token**

Eine eindeutige Kennung einer Nachricht in einem aktiven Warteschlangenmanager.

**Methode**

Im objektorientierten Design oder in der objektorientierten Programmierung die Software, die das von einer Operation definierte Verhalten implementiert.

**MFS**

Siehe [Message Format Service](#).

**MGAS**

Siehe [Mostly Global Address Space](#).

**Microsoft Cluster Server (MSCS)**

Eine Technologie, um hohe Verfügbarkeit zu erreichen, indem Windows-Computer zu MSCS-Clustern zusammengefasst werden. Wenn einer der Computer im Cluster auf ein Problem aus einem bestimmten Problembereich stößt, beendet MSCS die fehlerhafte Anwendung auf ordnungsgemäße Art, überträgt ihre Statusdaten an einen anderen Computer im Cluster und startet die Anwendung dort erneut.

**Microsoft Transaction Server (MTS)**

Eine Funktion, die Windows-Benutzer bei der Ausführung von Geschäftslogikanwendungen in einem Server der Mittelschicht unterstützt. MTS teilt Arbeit in Aktivitäten ein, die kurze unabhängige Teile eines Geschäftsablaufs darstellen.

**MID (Message Input Descriptor)**

Siehe [Nachrichteneingabedeskriptor](#).

**MOD (Message Output Descriptor)**

Siehe [Nachrichtenausgabedeskriptor](#).

**Modellwarteschlangenobjekt**

Eine Gruppe von Warteschlangenattributen, die beim Erstellen einer dynamischen Warteschlange durch ein Programm als Schablone dienen.

**Mostly Global Address Space (MGAS)**

Ein flexibles Modell für virtuelle Adressräume, das in Systemen wie HP-UX verwendet wird und den größten Teil des Adressraums für gemeinsam genutzte Anwendungen reserviert. Dies kann die Leistung von Prozessen verbessern, die große Datenmengen gemeinsam nutzen. Siehe auch [Mostly Private Address Space](#).

**Mostly Private Address Space (MPAS)**

Ein flexibles Modell für virtuelle Adressräume, das in Systemen wie HP-UX verwendet wird und Prozessen größere Adressraumblocke zuweisen kann. Dies kann die Leistung von Prozessen verbessern, die viel Datenspeicherbereich erfordern. Siehe auch [Mostly Global Address Space](#).

**MPAS**

Siehe [Mostly Private Address Space](#).

**MQAI**

Siehe [WebSphere MQ Administration Interface](#).

**MQI**

Siehe [Message Queue Interface](#).

**MQI-Kanal**

Eine Verbindung zwischen einem WebSphere MQ-Client und einem Warteschlangenmanager auf einem Serversystem. Ein MQI-Kanal überträgt nur MQI-Aufrufe und -Antworten in bidirektionaler Weise. Siehe auch [Kanal](#).

**MQM**

Siehe [Message Queue Management](#).

**MQSC**

Siehe [WebSphere MQ-Scriptbefehle](#).

**MQSeries**

Der frühere Name von WebSphere MQ.

**MQ Telemetry Transport**

MQ Telemetry Transport (MQTT) ist ein offenes, einfaches Publish/Subscribe-Protokoll, das über TCP/IP übertragen wird und mit dem eine große Anzahl von Einheiten wie Servomechanismen, Aktuatoren, Smart Phones, Fahrzeuge, Home-, Gesundheits- und ferne Sensoren sowie Steuereinheiten verbunden werden kann. MQTT wird in Umgebungen eingesetzt, in denen die Netzbandbreite möglicherweise begrenzt ist oder zum Beispiel die Speicher- oder Prozessorkapazität des Geräts eingeschränkt ist.

**MQTT**

Siehe [MQ Telemetry Transport](#).

## **MQTT-Client**

Eine MQTT-Clientanwendung stellt eine Verbindung zu MQTT-fähigen Servern wie beispielsweise WebSphere MQ Telemetry-Kanälen her. Sie können eigene Clients schreiben, die das veröffentlichte Protokoll verwenden sollen, oder einen der mit der WebSphere MQ Telemetry-Installation bereitgestellten Clients verwenden. Ein typischer Client ist für die Erfassung von Informationen aus einem Telemetriegerät und die Veröffentlichung der Informationen an den Server verantwortlich. Er kann außerdem Themen abonnieren, Nachrichten empfangen und mit diesen Informationen das Telemetriegerät steuern. Einige Clients werden mit WebSphere MQ Telemetry bereitgestellt; siehe [Telemetrieclients](#) und [Erweiterte Telemetrieclients](#).

## **MQTT-Server**

Ein MQTT-Server ist für die Serverseite des MQTT-Protokolls zuständig. Er erlaubt in der Regel vielen MQTT-Clients, gleichzeitig eine Verbindung mit ihm herzustellen, und stellt einen Hub für die Verteilung von Nachrichten für die MQTT-Clients bereit. Ein WebSphere MQ-Warteschlangenmanager mit Telemetrieservice (MQXR) ist ein MQTT-Server.

## **MSCS**

Siehe [Microsoft Cluster Server](#).

## **MTS**

Siehe [Microsoft Transaction Server](#).

## **Über Multihopping ansteuern**

Die Weiterleitung über einen oder mehrere zwischengeschaltete Warteschlangenmanager, wenn keine direkte Kommunikationsverbindung zwischen dem Quellen-Warteschlangenmanager und dem Ziel-Warteschlangenmanager besteht.

## **Multi-Instanz-Warteschlangenmanager**

Ein Warteschlangenmanager, der so konfiguriert ist, dass die Daten des Warteschlangenmanagers auch von anderen Warteschlangenmanagerinstanzen genutzt werden. Eine Instanz des aktiven Multi-Instanz-Warteschlangenmanagers ist aktiv, während andere Instanzen im Standby-Betrieb bereitgehalten werden, um jederzeit den Betrieb von der aktiven Instanz übernehmen zu können. Siehe auch [Einzel-Instanz-Warteschlangenmanager](#).

# **N**

## **Namensliste**

Ein WebSphere MQ-Objekt, das eine Liste mit Objektnamen, z. B. Warteschlangennamen, enthält.

## **Namensservice**

In WebSphere MQ auf UNIX- und Linux-Systemen und WebSphere MQ for Windows die Funktion, die bestimmt, welcher Warteschlangenmanager Eigner einer angegebenen Warteschlange ist

## **Namensserviceschnittstelle (NSI)**

Die WebSphere-MQ-Schnittstelle, der von Kunden oder Anbietern geschriebene Programme, die das Eigentumsrecht an Warteschlangennamen auflösen, entsprechen müssen. Ein Teil des WebSphere MQ-Frameworks.

## **Namensumsetzung**

In WebSphere MQ auf UNIX- und Linux-Systemen und WebSphere MQ for Windows ein interner Prozess, der einen Warteschlangenmanagernamen so ändert, dass er für das verwendete System eindeutig und gültig ist. Extern bleibt der Name des Warteschlangenmanagers unverändert.

## **verschachtelter Behälter**

In WebSphere MQ Administration Interface (MQAI) ein Systembehälter, der in einen anderen Datenbehälter eingefügt ist.

## **nesting**

In WebSphere MQ Administration Interface (MQAI) ein Verfahren zum Gruppieren von Informationen, die von WebSphere MQ zurückgegeben werden.

## **NetBIOS (Network Basic Input/Output System)**

Eine Standardschnittstelle für Netze und Personal Computer, die in lokalen Netzen zur Bereitstellung von Nachrichten-, Druckserver- und Dateiserverfunktionen verwendet wird. Anwendungsprogramme,



die NetBIOS verwenden, müssen sich nicht mit den Details von Protokollen für die LAN-Datenübertragungssteuerung beschäftigen.

### **Network Basic Input/Output System**

Siehe [NetBIOS](#).

### **New Technology File System (NTFS)**

Eines der systemeigenen Dateisysteme in Windows-Betriebsumgebungen.

### **Knoten**

In Microsoft Cluster Server (MSCS) ist jeder Computer im Cluster ein Knoten.

### **Nicht permanente Subskription**

Eine Subskription, die nur so lange existiert, wie die Verbindung der abonnierenden Anwendung zum Warteschlangenmanager geöffnet ist. Die Subskription wird entfernt, wenn die abonnierende Anwendung absichtlich oder durch eine Verbindungsunterbrechung vom Warteschlangenmanager getrennt wird. Siehe auch [Permanente Subskription](#).

### **Nicht persistente Nachricht**

Eine Nachricht, die nach dem Neustart eines Warteschlangenmanagers nicht mehr vorhanden ist. Siehe auch [Persistente Nachricht](#).

### **NSI (Name Service Interface)**

Siehe [Namensserviceschnittstelle](#).

### **NTFS**

Siehe [New Technology File System](#).

### **NUL**

Siehe [Nullzeichen](#).

### **Nullzeichen (NUL)**

Ein Steuerzeichen mit dem Wert X'00', das die Abwesenheit eines angezeigten oder gedruckten Zeichens darstellt.

## **O**

### **OAM**

Siehe [Objektberechtigungsmanager](#).

### **Objekt**

1. In WebSphere MQ ein Warteschlangenmanager, eine Warteschlange, eine Prozessdefinition, ein Kanal, eine Namensliste, ein Authentifizierungsdatenobjekt, ein Topic-Verwaltungsobjekt, ein Empfangsprogramm, ein Serviceobjekt oder (nur unter z/OS) ein CF-Strukturobjekt oder eine Speicherklasse.
2. Im objektorientierten Design oder in der objektorientierten Programmierung die konkrete Realisierung (Instanz) einer Klasse, die aus Daten und den diesen Daten zugeordneten Operationen besteht. Ein Objekt enthält die Instanzdaten, die durch die Klasse definiert werden, während die Klasse die Operationen besitzt, die den Daten zugeordnet sind.

### **Objektberechtigungsmanager (OAM)**

In WebSphere MQ auf UNIX- und Linux-Systemen WebSphere MQ für IBM und WebSphere MQ for Windows der Standardberechtigungs-service für Befehls- und Objektverwaltung. Der Objektberechtigungsmanager kann durch einen vom Kunden bereitgestellten Sicherheits-service ersetzt oder in Kombination mit diesem ausgeführt werden.

### **Objektdeskriptor**

Eine Datenstruktur, die ein bestimmtes WebSphere-MQ-Objekt beschreibt. Der Deskriptor enthält auch den Namen des Objekts und den Objekttyp.

### **Objektkennung**

Die Kennung oder das Token, über die oder das ein Programm auf das WebSphere MQ-Objekt zugreift, mit dem es arbeitet.

### **Objektorientierte Programmierung**

Eine Programmiermethode, die auf den Konzepten der Datenabstraktion und Vererbung basiert. Anders als die prozedurorientierten Programmier-techniken konzentriert sich die objektorientierte

Programmierung nicht darauf, wie etwas erreicht wird, sondern darauf, welche Datenobjekte zum Problem gehören und wie sie verarbeitet werden.

### **OCSP**

Online Certificate Status Protocol. Eine Methode zur Überprüfung, ob ein Zertifikat widerrufen wurde.

### **Auslagerung**

In WebSphere MQ for z/OS ein automatischer Prozess, bei dem das aktive Protokoll eines Warteschlangenmanagers in das Archivprotokoll übertragen wird.

### **Unidirektionale Authentifizierung**

Bei dieser Authentifizierungsmethode legt der Warteschlangenmanager das Zertifikat dem Client vor, aber umgekehrt führt der Warteschlangenmanager keine Authentifizierung des Clients durch.

### **Öffnen**

Das Einrichten des Zugriffs auf ein Objekt, z. B. eine Warteschlange oder ein Topic.

### **Open Systems Interconnection (OSI)**

Die Verbindung zwischen offenen Systemen gemäß den ISO-Normen (International Organization for Standardization) für den Informationsaustausch.

### **Open Transaction Manager Access (OTMA)**

Eine Komponente von IMS, die ein transaktionsorientiertes, verbindungsunabhängiges Client/Server-Protokoll in einer MVS-Sysplexumgebung implementiert. Die Domäne des Protokolls ist auf die Domäne der Cross-System Coupling Facility (XCF) von z/OS beschränkt. OTMA verbindet Clients mit Servern, sodass der Client ein großes Netz (oder eine große Anzahl von Sitzungen) unterstützt und dabei eine hohe Leistung beibehält.

### **OPM**

Siehe [ursprüngliches Programmmodell](#).

### **Original Program Model (OPM)**

Die Gruppe von Funktionen, die vor der Einführung des ILE-Modells (Integrated Language Environment) für die Kompilierung von Quellcode und die Erstellung von Programmen in einer höheren Programmiersprache verwendet wurden.

### **Open Service Gateway-Initiative**

Ein Konsortium aus über 20 Unternehmen, darunter IBM, das Spezifikationen für den Entwurf offener Standards zur Verwaltung von Sprache, Daten und Multimedia in Mobil- und Festnetzen erstellt.

### **OSI**

Siehe [Open Systems Interconnection](#).

### **OSI-Verzeichnisstandard**

Der als X.500 bekannte Standard, der einen umfassenden Verzeichnisservice, einschließlich eines Informationsmodells, Namensbereichs, Funktionsmodells und Authentifizierungs-Frameworks, definiert. X.500 definiert außerdem das Verzeichniszugriffsprotokoll, über das Clients auf das Verzeichnis zugreifen. Lightweight Directory Access Protocol (LDAP) entlastet die Verzeichnisclients teilweise von X.500-Zugriffen, wodurch das Verzeichnis einer größeren Zahl von Maschinen und Anwendungen zur Verfügung steht.

### **OTMA**

Siehe [Open Transaction Manager Access](#).

### **Kanal für abgehende Nachrichten**

Ein Kanal, der Nachrichten aus einer Übertragungswarteschlange abrufen und an einen anderen Warteschlangenmanager sendet.

### **Ausgabeprotokollpuffer**

In WebSphere MQ for z/OS ein Puffer, in dem Wiederherstellungsprotokolleinträge abgelegt werden, bevor sie in das Archivprotokoll geschrieben werden.

### **Ausgabeparameter**

Ein Parameter eines MQI-Aufrufs, in dem der Warteschlangenmanager Informationen über die erfolgreiche bzw. nicht erfolgreiche Ausführung des Aufrufs zurückgibt.

## Überladung

In der objektorientierten Programmierung die Fähigkeit eines Operators oder einer Methode, je nach Kontext eine andere Bedeutung anzunehmen. In C++ kann ein Benutzer beispielsweise Funktionen und die meisten Standardoperatoren neu definieren, wenn die Funktionen und Operatoren mit Klassentypen verwendet werden. Der Methodenname bzw. Operator bleibt derselbe, aber die Methodenparameter unterscheiden sich in Typ und/oder Anzahl. Dieser Unterschied wird kollektiv als Signatur der Funktion bzw. des Operators bezeichnet und jede Signatur erfordert eine separate Implementierung.

## P

### Seitengruppe

Eine VSAM-Datei, die in WebSphere MQ for z/OS beim Verschieben von Daten (z. B. Warteschlangen und Nachrichten) aus Puffern im Hauptspeicher in einen permanenten Sicherungsspeicher (DASD) verwendet wird.

### Übergeordnete Klasse

Eine Klasse, von der eine andere Klasse Instanzmethoden, Attribute und Instanzvariablen übernimmt. Siehe auch [Abstrakte Klasse](#).

### Teilrepository

Eine Teilmenge der Informationen zu Warteschlangenmanagern in einem Cluster. Ein Teilrepository wird von allen Clusterwarteschlangenmanagern verwaltet, die kein vollständiges Repository besitzen. Siehe auch [Vollständiges Repository](#).

### Partner-Warteschlangenmanager

Siehe [Ferner Warteschlangenmanager](#).

### PasSTicket

Bei RACF-gesicherten Anmeldungen ein dynamisch generierter, zufälliger Kennwortsatz zur einmaligen Verwendung, mit dem sich eine Workstation oder ein anderer Client beim Host anmelden kann, statt ein RACF-Kennwort über das Netz zu senden.

### PCF

Siehe [Programmable Command Format](#).

### Anstehendes Ereignis

Ein nicht geplantes Ereignis, das als Ergebnis einer Verbindungsanforderung von einem CICS-Adapter auftritt.

### Unveränderte Weiterleitung

Bei der Fehlerbehebung der Weg von einer Wiederherstellungsroutine über einen vordefinierten Steuerungspfad zu einer Wiederherstellungsroutine der höheren Ebene.

### Leistungsereignis

Eine Ereigniskategorie, die anzeigt, dass eine Einschränkung festgestellt wurde.

### Leistungstrace

Eine WebSphere MQ-Traceoption, bei der die Tracedaten zur Leistungsanalyse und Optimierung verwendet werden.

### Permanente dynamische Warteschlange

Eine dynamische Warteschlange, die beim Schließen nur dann gelöscht wird, wenn das Löschen explizit angefordert wird. Permanente dynamische Warteschlangen werden nach dem Ausfall eines Warteschlangenmanagers wiederhergestellt, sodass sie permanente Nachrichten enthalten können. Siehe auch [Temporäre dynamische Warteschlange](#).

### Persistente Nachricht

Eine Nachricht, die auch nach dem Wiederanlauf eines Warteschlangenmanagers noch vorhanden ist. Siehe auch [Nicht persistente Nachricht](#).

### Persönlichem Zertifikat

Ein Zertifikat, für das Sie den zugehörigen privaten Schlüssel besitzen. Ist Warteschlangenmanagern oder Anwendungen zugeordnet.

**PGM**

Siehe [Pragmatic General Multicast](#).

**Prozess-ID**

Siehe [Prozess-ID](#).

**ping**

Der Befehl, mit dem ein ICMP-Echoanforderungspaket (Internet Control Message Protocol) an einen Gateway, Router oder Host gesendet wird mit der Erwartung, dass eine Antwort empfangen wird.

**PKCS**

Public Key Cryptography Standards. Eine Gruppe von Verschlüsselungsstandards, wobei

- 7 für Nachrichten,
- 11 für Hardwaresicherheitsmodule und
- 12 für das im Schlüsselrepository verwendete Dateiformat gilt.

**PKI**

Siehe [Public Key Infrastructure](#).

**Unverschlüsselter Text**

Siehe [Klartext](#).

**Wiederherstellungspunkt**

In WebSphere MQ for z/OS eine Gruppe von Sicherungskopien von WebSphere MQ for z/OS-Seitengruppen sowie der entsprechenden Protokolldateien, die zur Wiederherstellung dieser Seitengruppen erforderlich sind. Die Sicherungskopien stellen den potenziellen Wiederanlaufpunkt für den Fall von Seitengruppenverlusten dar (z. B. bei einem E/A-Fehler für die Seitengruppe).

**Falsch formatierte Nachricht**

In einer Warteschlange eine nicht ordnungsgemäß formatierte Nachricht, die von der empfangenden Anwendung nicht verarbeitet werden kann. Die Nachricht kann mehrfach an die Eingabewarteschlange zugestellt und mehrfach von der Anwendung zurückgesetzt werden.

**Polymorphie**

Ein Merkmal der objektorientierten Programmierung, das in Abhängigkeit von der Klasse, von der eine Methode implementiert wird, eine unterschiedliche Ausführung dieser Methode ermöglicht. Durch Polymorphie ist es möglich, dass eine Unterklasse eine übernommene Methode überschreibt, ohne dass dies Auswirkungen auf die Methode der übergeordneten Klasse hat. Darüber hinaus ermöglicht Polymorphie einem Client den Zugriff auf zwei oder mehr Implementierungen eines Objekts über eine einzelne Schnittstelle.

**Pragmatic General Multicast (PGM)**

Ein zuverlässiges Multicastransportprotokoll, über das mehreren Empfängern gleichzeitig eine zuverlässige Folge von Paketen zustellt werden kann.

**Präventiver Abschluss**

In WebSphere MQ die Beendigung eines Warteschlangenmanagers, ohne dass auf die Unterbrechung der Verbindungen zu Anwendungen oder den Abschluss aktueller MQI-Aufrufe gewartet wird. Siehe auch [Sofortige Beendigung](#) und [Gesteuerter Abschluss](#).

**Bevorzugter Computer**

Der primäre Computer, der von einer Anwendung verwendet wird, die unter Steuerung von Microsoft Cluster Server aktiv ist. Nach einem Failover auf einen anderen Computer überwacht MSCS den bevorzugten Computer, bis dieser wiederhergestellt ist. Sobald der Computer wieder ordnungsgemäß funktioniert, wird die Anwendung wieder auf diesen Computer verschoben.

**Prinzipal**

Eine Entität, die auf sichere Weise mit einer anderen Entität kommunizieren kann. Ein Principal wird durch seinen zugehörigen Sicherheitskontext identifiziert, der seine Zugriffsrechte definiert.

**Privat definiertes Objekt**

Siehe [Lokal definiertes Objekt](#).

**Private Methoden und Instanzdaten**

In der objektorientierten Programmierung Methoden und Instanzdaten, die nur für die Implementierung derselben Klasse zugänglich sind.

**Prozessdefinitionsobjekt**

Ein WebSphere MQ-Objekt, das die Definition einer WebSphere MQ-Anwendung enthält. Ein Warteschlangenmanager verwendet die Definition beispielsweise bei der Arbeit mit Auslösenachrichten.

**Prozess-ID (PID)**

Die eindeutige Kennung (ID) für einen Prozess. Eine Prozess-ID ist eine positive Ganzzahl, die erst nach Ende der Prozesslebensdauer wiederverwendet werden kann.

**Produzent**

Eine Anwendung, die Nachrichten erstellt und sendet. Siehe auch [Publisher](#), [Nachrichtenproduzent](#).

**Programmable Command Format (PCF)**

Ein WebSphere MQ-Nachrichtentyp, der von folgenden Anwendungen verwendet wird: Benutzerverwaltungsanwendungen verwenden ihn, um PCF-Befehle in die Eingabewarteschlange für Systembefehle eines bestimmten Warteschlangenmanagers einzureihen und die Ergebnisse eines PCF-Befehls von einem bestimmten Warteschlangenmanager abzurufen; Warteschlangenmanager verwenden ihn als Benachrichtigung darüber, dass ein Ereignis aufgetreten ist. Siehe auch [WebSphere MQ-Scriptbefehle](#).

**Vorläufige Programmkorrektur (Program Temporary Fix, PTF)**

In System i-, System p- und System z-Produkten ein Paket, das einzelne oder mehrere Programmkorrekturen enthält, die allen lizenzierten Kunden zur Verfügung gestellt werden. Eine vorläufige Programmkorrektur behebt Mängel und kann funktionale Erweiterungen enthalten.

**Eigenschaft**

Ein Merkmal eines Objekts, das dieses Objekt beschreibt. Eine Eigenschaft kann geändert werden. Eigenschaften können unter anderem den Namen, Typ, Wert oder das Verhalten eines Objekts beschreiben.

**Geschützte Methoden und Instanzdaten**

In der objektorientierten Programmierung Methoden und Instanzdaten, die nur für die Implementierungen derselben oder abgeleiteter Klassen und für Friend-Klassen zugänglich sind.

**Proxy-Subskription**

Eine Proxy-Subskription ist eine Subskription, die von einem Warteschlangenmanager für Themen eingerichtet wird, die auf einem anderen Warteschlangenmanager veröffentlicht werden. Eine Proxy-Subskription fließt zwischen Warteschlangenmanagern für jede einzelne Themenzeichenfolge, für die eine Subskription eingerichtet wurde. Sie müssen Proxy-Subskriptionen nicht explizit erstellen; das macht der Warteschlangenmanager automatisch für Sie.

**PTF**

Siehe [Vorläufige Programmkorrektur](#).

**öffentlicher Schlüssel**

Der allen bekannte Schlüssel. Dieser Schlüssel ist üblicherweise in ein digitales Zertifikat eingebettet, das den Eigner des öffentlichen Schlüssels angibt.

**Public-Key-Verschlüsselung**

Ein Verschlüsselungssystem, bei dem zwei Schlüssel verwendet werden: ein öffentlicher Schlüssel, der allen bekannt ist, und ein privater oder geheimer Schlüssel, den nur der Empfänger der Nachricht kennt. Die Verbindung zwischen dem öffentlichen und dem privaten Schlüssel besteht darin, dass alles, was mit dem öffentlichen Schlüssel verschlüsselt wird, nur mit zugehörigen privaten Schlüssel entschlüsselt werden kann.

**Public Key Infrastructure (PKI)**

Ein System aus digitalen Zertifikaten, Zertifizierungsstellen und anderen Registrierungsstellen, die die Gültigkeit jeder an einer Netztransaktion beteiligten Partei überprüfen und authentifizieren.

**Öffentliche Methoden und Instanzdaten**

In der objektorientierten Programmierung Methoden und Instanzdaten, die für alle Klassen zugänglich sind.

**veröffentlichen**

Die Bereitstellung von Informationen zu einem bestimmten Thema für einen Warteschlangenmanager in einem Publish/Subscribe-System.

### **Veröffentlichungsanwendung**

Eine Anwendung, die Informationen zu einem angegebenen Thema für einen Broker in einem Publish/Subscribe-System bereitstellt.

### **Publish/Subscribe**

Ein Typ der Nachrichtenübermittlungsinteraktion, bei der Informationen, die von veröffentlichenden Anwendungen bereitgestellt werden, von einer Infrastruktur für alle subscribierenden Anwendungen bereitgestellt werden, die Interesse an diesem Informationstyp bekundet haben.

### **Publish/Subscribe-Cluster**

Eine Gruppe von Warteschlangenmanagern, die vollständig miteinander verbunden sind und Teil eines Netzes aus mehreren Warteschlangenmanagern für Publish/Subscribe-Anwendungen bilden.

### **put**

Beim Message-Queuing die Verwendung von MQPUT- oder MQPUT1-Aufrufen, um Nachrichten in eine Warteschlange zu stellen. Siehe auch [Durchsuchen](#) und [Abrufen](#).

## **Q**

### **Warteschlange**

Ein Objekt, das Nachrichten für Message-Queuing-Anwendungen enthält. Eigner und Verwalter von Warteschlangen ist ein Warteschlangenmanager.

### **Warteschlangenindex**

In WebSphere MQ for z/OS eine Liste mit Nachrichten-IDs oder eine Liste mit Korrelations-IDs, mit deren Hilfe MQGET-Operationen in der Warteschlange beschleunigt werden können.

### **Warteschlangenmanager**

Eine Komponente eines Message-Queuing-Systems, die Services zur Steuerung von Warteschlangen für Anwendungen bereitstellt.

### **Warteschlangenmanager-Ereignis**

Ein Ereignis, das Folgendes anzeigt: Im Zusammenhang mit den Ressourcen, die von einem Warteschlangenmanager verwendet werden, ist eine Fehlerbedingung aufgetreten. Beispielsweise ist eine Warteschlange nicht verfügbar oder es hat eine erhebliche Änderung im Warteschlangenmanager gegeben. Beispiel: Ein Warteschlangenmanager wurde gestoppt oder gestartet.

### **Warteschlangenmanager-Gruppe**

In einer Clientkanaldefinitionstabelle die Gruppe von Warteschlangenmanagern, zu denen ein Client eine Verbindung herzustellen versucht, wenn eine Verbindung zu einem Server aufgebaut wird.

### **Sicherheit auf WS-Managerebene**

In WebSphere MQ for z/OS die Berechtigungsprüfungen, die abhängig von RACF-Profilen, die für einen Warteschlangenmanager spezifisch sind, durchgeführt werden.

### **Warteschlangenmanager-Set**

Eine Gruppierung von Warteschlangenmanagern in WebSphere MQ Explorer, die es einem Benutzer ermöglicht, Aktionen für alle Warteschlangenmanager in der Gruppe auszuführen.

### **Gruppe mit gemeinsamer Warteschlange**

In WebSphere MQ for z/OS eine Gruppe von Warteschlangenmanagern im selben Sysplex, die Zugriff auf eine einzelne Gruppe von Objektdefinitionen im gemeinsam genutzten Repository sowie auf eine einzelne Gruppe gemeinsamer Warteschlangen in der Coupling Facility hat. Siehe auch [Gemeinsam genutzte Warteschlangen](#).

### **Sicherheit auf Ebene der Gruppe mit gemeinsamer Warteschlange**

In WebSphere MQ for z/OS Berechtigungsprüfungen, die abhängig von RACF-Profilen durchgeführt werden, die von allen Warteschlangenmanagern in einer Gruppe mit gemeinsamer Warteschlange gemeinsam genutzt werden.

### **Quiesce**

Das Beenden eines Prozesses oder Herunterfahren eines Systems, nachdem die normale Beendigung der aktiven Operationen ermöglicht wurde.

## **Gesteuerter Abschluss**

1. Die Beendigung eines CICS-Adapters, bei der die Verbindung zwischen dem Adapter und WebSphere MQ erst nach Abschluss aller aktiven Tasks getrennt wird. Siehe auch [Erzwungene Beendigung](#).
2. In WebSphere MQ ist dies der Abschluss eines Warteschlangenmanagers, bei dem die Verbindungen aller angeschlossenen Anwendungen beendet werden. Siehe auch [Sofortiger Abschluss](#), [Präventiver Abschluss](#).

## **stilllegen**

In WebSphere MQ der Status eines Warteschlangenmanagers, bevor er gestoppt wird. In diesem Status können alle Programme ordnungsgemäß beendet, aber keine neuen Programme gestartet werden.

## **Quorumplatte**

Der Datenträger, auf den Microsoft Cluster Server exklusiv zugreift, um das Clusterwiederherstellungsprotokoll zu speichern und um zu ermitteln, ob ein Server aktiv oder inaktiv ist. Nur ein einziger Server kann Eigner der Quorumplatte sein. Die Server im Cluster können das Eigentumsrecht untereinander aushandeln.

# **R**

## **RACF**

Siehe [Resource Access Control Facility](#).

## **RAID**

Siehe [Redundant Array of Independent Disks](#).

## **RBA**

Siehe [Relative Byteadresse](#).

## **RC**

Siehe [Rückkehrcode](#).

## **Vorauslesen**

Eine Option, mit der Nachrichten an einen Client gesendet werden können, bevor sie von einer Anwendung angefordert werden.

## **Ursachencode**

Ein Rückkehrcode, der die Ursache für das Fehlschlagen oder teilweise Fehlschlagen eines MQI-Aufrufs beschreibt.

## **Empfangsexit**

Ein Kanalexitprogramm, das aufgerufen wird, unmittelbar nachdem der Nachrichtenkanalagent (MCA) eine Dateneinheit über eine Kommunikationsverbindung empfangen und wieder die Steuerung der Übertragung übernommen hat. Siehe auch [Sendeexit](#).

## **Empfängerkanal**

Beim Message-Queuing ein Kanal, der einem Senderkanal antwortet, Nachrichten von einer Kommunikationsverbindung annimmt und diese in eine lokale Warteschlange einreicht.

## **Wiederherstellungsprotokoll**

In WebSphere MQ for z/OS Dateien, die Informationen enthalten, die zur Wiederherstellung von Nachrichten, Warteschlangen und des WebSphere-MQ-Subsystems erforderlich sind. Siehe auch [Archivprotokoll](#).

## **Recovery Termination Manager (RTM)**

Ein Programm, das die normale und abnormale Beendigung von Tasks handhabt, indem es die Steuerung an eine Wiederherstellungsroutine übergibt, die der Beendigungsfunktion zugeordnet ist.

## **Redundant Array of Independent Disks (RAID)**

Eine Gruppe aus zwei oder mehr physischen Plattenlaufwerken, die für den Host ein oder mehrere logische Plattenlaufwerke darstellen. Falls eine physische Einheit ausfällt, können die Daten dank der Datenredundanz von den übrigen Plattenlaufwerken im selben Array gelesen oder neu generiert werden.

**Referenznachricht**

Eine Nachricht, die auf ein zu übertragendes Datenelement verweist. Die Referenznachricht wird von Nachrichtenexitprogrammen verarbeitet, die die Daten an die Nachricht anhängen und wieder abhängen, sodass die Daten übertragen werden können, ohne dass sie in irgendwelchen Warteschlangen gespeichert werden müssen.

**Registrierungsdatenbank**

Ein Repository, das Zugriffs- und Konfigurationsinformationen für Benutzer, Systeme und Software enthält.

**Registrierungseditor**

In Windows das Programm, das dem Benutzer die Änderung der Registrierungsdatenbank ermöglicht.

**Gruppe von Registrierungsschlüsseln**

Auf Windows-Systemen die Struktur der in der Registrierungsdatenbank gespeicherten Daten.

**Relative Byteadresse (RBA)**

Die relative Adresse eines Datensatzes oder Steuerintervalls ab dem Beginn des Speicherbereichs, der dem zugehörigen Datensatz bzw. der zugehörigen Datei zugeordnet ist.

**Reliable Multicast Messaging (RMM)**

Eine durchsatzstarke Transportstruktur mit niedriger Latenzzeit, die für eine Eins-zu-viele-Datenbereitstellung oder einen Viele-zu-viele-Datenaustausch in der Publish/Subscribe-Funktion einer nachrichtenorientierten Middleware entwickelt wurde. RMM nutzt die IP-Multicast-Infrastruktur, um einen skalierbaren Ressourcenschutz und eine zeitgerechte Informationsverteilung sicherzustellen.

**Ferne Warteschlange**

Eine Warteschlange, die zu einem fernen Warteschlangenmanager gehört. Programme können Nachrichten in ferne Warteschlangen einreihen, aber keine Nachrichten daraus abrufen. Siehe auch [Lokale Warteschlange](#).

**Ferner Warteschlangenmanager**

Ein Warteschlangenmanager, mit dem ein Programm nicht verbunden ist, auch wenn er auf demselben System wie das Programm aktiv ist. Siehe auch [Lokaler Warteschlangenmanager](#).

**Fernes Warteschlangenobjekt**

Ein WebSphere-MQ-Objekt, das zu einem lokalen Warteschlangenmanager gehört. Dieses Objekt definiert die Attribute einer Warteschlange, deren Eigner ein anderer Warteschlangenmanager ist. Darüber hinaus wird dieses Objekt zur Aliasnamensumsetzung für Warteschlangenmanager und Warteschlangen für Antwortnachrichten verwendet.

**Ferne Warteschlangensteuerung**

Beim Message-Queuing die Bereitstellung von Services, die es Anwendungen ermöglichen, Nachrichten in Warteschlangen einzureihen, die zu anderen Warteschlangenmanagern gehören.

**Antwortnachricht**

Ein Nachrichtentyp, der für Antworten auf Anforderungsnachrichten verwendet wird. Siehe auch [Berichtsnachricht](#) und [Anforderungsnachricht](#).

**Empfangswarteschlange für Antworten**

Der Name einer Warteschlange, an die auf Anforderung des Programms, das einen MQPUT-Aufruf ausgegeben hat, eine Antwort- oder Berichtsnachricht gesendet werden soll.

**Berichtsnachricht**

Ein Nachrichtentyp, der Informationen zu einer anderen Nachricht enthält. Eine Berichtsnachricht kann die Information enthalten, dass eine Nachricht zugestellt wurde, ihr Ziel erreicht hat, abgelaufen ist oder aus irgendeinem Grund nicht verarbeitet werden konnte. Siehe auch [Antwortnachricht](#), [Anforderungsnachricht](#).

**Repository**

Eine Sammlung von Informationen über die Warteschlangenmanager, die zu einem Cluster gehören. Zu diesen Informationen gehören die Namen der Warteschlangenmanager, ihre Standorte, ihre Kanäle und die zugehörigen Warteschlangen.

**Repository-Warteschlangenmanager**

Ein Warteschlangenmanager, der das vollständige Repository mit Informationen zu einem Cluster verwaltet.



**Requesterkanal**

Beim Message-Queuing ein Kanal, der lokal gestartet werden kann, um den Betrieb eines Serverkanals anzustoßen. Siehe auch [Serverkanal](#).

**Anforderungsnachricht**

Ein Nachrichtentyp, mit dem eine Antwort von einem anderen Programm angefordert wird. Siehe auch [Antwortnachricht](#), [Berichtsnachricht](#).

**Anforderung/Antwort**

Ein Typ von Messaging-Anwendung, bei dem mit einer Anforderungsnachricht eine Antwort von einer anderen Anwendung angefordert wird. Siehe auch [Datagramm](#).

**RESLEVEL**

In WebSphere MQ for z/OS eine Option, die die Anzahl der Benutzer-IDs steuert, die zur Sicherheit von API-Ressourcen überprüft werden.

**Auflösungspfad**

Die Warteschlangen, die geöffnet werden, wenn eine Anwendung als Eingabe in einem MQOPEN-Aufruf einen Aliasnamen oder eine ferne Warteschlange angibt.

**Ressource**

Eine Einrichtung eines Computersystems oder Betriebssystems, die für einen Job, eine Task oder ein aktives Programm erforderlich ist. Zu den Ressourcen gehören Hauptspeicher, Ein-/Ausgabeeinheiten, Verarbeitungseinheit, Datenbestände, Dateien, Bibliotheken, Ordner und Steuer- oder Verarbeitungsprogramme.

**Resource Access Control Facility (RACF)**

Ein IBM Lizenzprogramm für die Zugriffssteuerung, mit dem Benutzer gegenüber dem System identifiziert, Systembenutzer überprüft, Zugriffsberechtigungen für geschützte Ressourcen erteilt sowie unberechtigte Zugriffsversuche auf das System und Zugriffe auf geschützte Ressourcen protokolliert werden.

**Ressourcenadapter**

Eine Implementierung von Java Enterprise Edition Connector Architecture, die JMS-Anwendungen und nachrichtenbasierten Beans, die auf einem Anwendungsserver aktiv sind, den Zugriff auf die Ressourcen eines WebSphere MQ-Warteschlangenmanagers ermöglicht.

**Ressourcenmanager**

Eine Anwendung, ein Programm oder eine Transaktion, die den Zugriff auf gemeinsam genutzte Ressourcen (z. B. Hauptspeicherpuffer und Dateien) verwaltet und steuert. WebSphere MQ, CICS und IMS sind Ressourcenmanager.

**Resource Recovery Services (RRS)**

Eine z/OS-Komponente, die Änderungen unter den teilnehmenden Ressourcenmanagern mithilfe eines Synchronisationspunktmanagers koordiniert.

**Responder**

Bei der verteilten Steuerung von Warteschlangen ein Programm, das auf Netzverbindungsanforderungen anderer Systeme antwortet. Siehe auch [Initiator](#).

**Resynchronisation**

In WebSphere MQ eine Option, mit der ein Kanal angewiesen werden kann, zu starten und alle unbestätigten Statusnachrichten aufzulösen, ohne jedoch die Nachrichtenübertragung erneut zu starten.

**Rückkehrcode (RC)**

Ein Wert, der von einem Programm zurückgegeben wird, um das Ergebnis seiner Verarbeitung anzugeben. Beispiele für Rückkehrcodes sind Beendigungs- und Ursachencodes.

**Return-to-Sender**

Eine Option, die einem Nachrichtenkanalagenten zur Verfügung steht, der eine Nachricht nicht zustellen kann. Der Nachrichtenkanalagent kann die Nachricht an den Sender zurückschicken.

**Rivest-Shamir-Adleman-Algorithmus (RSA)**

Eine Verschlüsselungstechnik mit öffentlichem Schlüssel, die von RSA Data Security Inc. entwickelt wurde und in der IBM Implementierung von SSL verwendet wird.

**RMM**

Siehe [Reliable Multicast Messaging](#).

**rollback**

Siehe [Backout](#).

**Stammzertifikat**

Das Zertifikat am Anfang der Zertifikatskette. Wenn es sich dabei um ein selbst signiertes Zertifikat handelt, wird es nur zum Signieren anderer Zertifikate verwendet. Siehe auch [Selbst signiertes Zertifikat](#).

**RRS**

Siehe [Resource Recovery Services](#).

**RSA**

Siehe [Rivest-Shamir-Adleman-Algorithmus](#).

**RTM**

Siehe [Recovery Termination Manager](#).

**Regeltabelle**

Eine Steuerdatei, die eine oder mehrere Regeln enthält, die die Steuerroutine der Warteschlange für nicht zustellbare Nachrichten auf Nachrichten in der Warteschlange für nicht zustellbare Nachrichten anwendet.

**S****Scalable Parallel 2 (SP2)**

Das parallele UNIX-System von IBM: effektiv parallele AIX-Systeme in einem Hochgeschwindigkeitsnetz.

**Software-Development-Kit (SDK)**

Siehe [Software-Development-Kit](#).

**SDWA**

Siehe [Arbeitsbereich für Systemdiagnose](#).

**SECMEC (Security Mechanism)**

Siehe [Sicherheitsmechanismus](#).

**Secure Sockets Layer (SSL)**

Ein Sicherheitsprotokoll, das Datenschutz für die Kommunikation bereitstellt. Mit SSL können Client/Server-Anwendungen auf eine Weise kommunizieren, die das Abhören und Manipulieren von Daten sowie das Fälschen von Nachrichten verhindern soll. Siehe auch [Zertifizierungsstelle](#).

**Schnittstelle für Sicherheitsaktivierung**

Die WebSphere-MQ-Schnittstelle, der von Kunden oder Anbietern geschriebene Programme, die Berechtigungen prüfen, eine Benutzer-ID übergeben oder Authentifizierungen durchführen, entsprechen müssen. Ein Teil des WebSphere MQ-Frameworks.

**Sicherheitsexit**

Ein Kanalexitprogramm, das unmittelbar nach Abschluss der Anfangsdatenvereinbarung während des Kanalstarts aufgerufen wird. Sicherheitsexits arbeiten in der Regel paarweise und können sowohl in Nachrichtenkanälen als auch in MQI-Kanälen aufgerufen werden. Die Hauptfunktion des Sicherheitsexits besteht darin, dem Nachrichtenkanalagenten auf einer Seite eines Kanals die Authentifizierung seines jeweiligen Partners auf der anderen Seite zu ermöglichen.

**Sicherheits-ID (SID)**

Auf Windows-Systemen eine Ergänzung der Benutzer-ID, die die vollständigen Benutzerkontodetails in der Datenbank des Windows-Sicherheitskontenmanagers, in der der Benutzer definiert ist, identifiziert.

**Sicherheitsmechanismus**

Ein technisches Tool bzw. eine Technik zum Implementieren eines Sicherheitsservice. Ein Mechanismus kann eigenständig oder in Verbindung mit anderen arbeiten, um einen bestimmten Service bereitzustellen. Beispiele für Sicherheitsmechanismen sind Zugriffssteuerungslisten, Verschlüsselung und digitale Signaturen.

**Sicherheitsnachricht**

Eine der Nachrichten, die von Sicherheitsexits gesendet werden, die auf beiden Seiten eines Kanals aufgerufen werden, um miteinander zu kommunizieren. Das Format einer Sicherheitsnachricht ist nicht definiert und wird vom Benutzer festgelegt.

**Sicherheitservice**

Ein Service in einem Computersystem, der die Ressourcen des Systems schützt. Die Zugriffssteuerung ist ein Beispiel für einen Sicherheitservice.

**Security Support Provider Interface (SSI)**

Die Schnittstelle, über die Netzanwendungen einen von mehreren Sicherheits-Providern (SSP, Security Support Provider) aufrufen, um authentifizierte Verbindungen herzustellen und Daten über diese Verbindungen sicher auszutauschen. Sie ist auf Windows-Systemen verfügbar.

**Selbst signiertes Zertifikat**

Die digitale Signatur im Zertifikat wird mithilfe des privaten Schlüssels, der zu dem öffentlichen Schlüssel im Zertifikat gehört, erzeugt.

**Segmentierung**

Die Aufteilung einer Nachricht, die für einen Warteschlangenmanager, eine Warteschlange oder eine Anwendung zu groß ist, in mehrere kleine physische Nachrichten, die dann vom empfangenden Warteschlangenmanager oder von der empfangenden Anwendung wieder zusammengesetzt werden.

**SEI (Security Enabling Interface)**

Siehe [Schnittstelle für Sicherheitsaktivierung](#).

**Selektor**

Eine ID für ein Datenelement. In WebSphere MQ Administration Interface (MQAI) gibt es zwei Selektortypen: einen Benutzerselektor und einen Systemselektor.

**Semaphor**

In UNIX- und Linux-Systemen eine allgemeine Methode der Kommunikation zwischen zwei Prozessen, die die Funktionen von Signalen erweitert.

**Senderkanal**

Beim Message-Queuing ein Kanal, der Übertragungen einleitet sowie Nachrichten aus Übertragungswarteschlangen abholt und über eine Kommunikationsverbindung in einen Empfänger- oder Requester-Kanal stellt.

**Sendeexit**

Ein Kanalexitprogramm, das unmittelbar, bevor ein Nachrichtenkanalagent eine Sendeanforderung zum Senden einer Dateneinheit über eine Kommunikationsverbindung absetzt, aufgerufen wird. Siehe auch [Empfangsexit](#).

**Sequenced Packet Exchange protocol (SPX)**

Ein sitzungsorientiertes Netzprotokoll, das verbindungsorientierte Services zwischen zwei Knoten im Netz bereitstellt und in erster Linie von Client/Server-Anwendungen verwendet wird. Das auf Internet Packet Exchange (IPX) basierende Protokoll sorgt für die Ablaufsteuerung und Fehlerbehebung und garantiert die Zuverlässigkeit des physischen Netzes.

**Wert für Folgenummernserie**

In WebSphere MQ eine Methode, mit der sichergestellt wird, dass beide Seiten einer Kommunikationsverbindung ihre aktuellen Nachrichtenfolgenummern gleichzeitig zurücksetzen. Durch die Übertragung von Nachrichten mit einer Folgenummer wird sichergestellt, dass der empfangende Kanal die Nachrichtenfolge wiederherstellen kann, wenn er die Nachrichten speichert.

**Serielle Verarbeitung**

In der objektorientierten Programmierung das sequenzielle Schreiben von Daten aus dem Programmspeicher in ein Kommunikationsmedium.

**Server**

1. Ein Warteschlangenmanager, der Warteschlangenservices für Clientanwendungen bereitstellt, die auf einer fernen Workstation aktiv sind.
2. Ein Softwareprogramm oder Computer, das bzw. der Services für andere Softwareprogramme oder Computer bereitstellt. Siehe auch [Client](#).

**Serverkanal**

Beim Message-Queuing ein Kanal, der einem Requesterkanal antwortet, Nachrichten aus einer Übertragungswarteschlange abholt und sie über eine Kommunikationsverbindung an den Requesterkanal übergibt. Siehe auch [Requesterkanal](#).

**Serververbindungskanaltyp**

Der MQI-Kanaldefinitionstyp, der dem Server zugeordnet ist, auf dem ein Warteschlangenmanager ausgeführt wird. Siehe auch [Clientverbindungskanaltyp](#).

**Serviceintervall**

Ein Zeitintervall, mit dem die abgelaufene Zeit zwischen einem PUT- bzw. GET-Aufruf und einem anschließenden GET-Aufruf vom Warteschlangenmanager verglichen wird. Anhand dieses Werts entscheidet der Warteschlangenmanager, ob ein Serviceintervallereignis ausgegeben wird oder nicht. Das Serviceintervall für eine Warteschlange wird mit einem Warteschlangenattribut angegeben.

**Serviceintervallereignis**

Ein Ereignis, das sich auf das Serviceintervall bezieht.

**Serviceobjekt**

Ein Objekt, mit dessen Hilfe zusätzliche Prozesse beim Start des Warteschlangenmanagers gestartet und beim Stoppen des Warteschlangenmanagers gestoppt werden können.

**Sitzung**

Eine logische oder virtuelle Verbindung zwischen zwei Stationen, Softwareprogrammen oder Einheiten in einem Netz, die die Kommunikation und den Datenaustausch zwischen diesen beiden Elementen für die Dauer der Sitzung ermöglicht.

**Sitzungs-ID**

In WebSphere MQ for z/OS die eindeutige CICS-Kennung, die die Kommunikationsverbindung definiert, die von einem Nachrichtenkanalagent verwendet werden soll, wenn er Nachrichten aus einer Übertragungswarteschlange an eine Verbindung übergibt.

**Authentifizierung auf Sitzungsebene**

In der Systemnetzwerkarchitektur (SNA) ein Sicherheitsprotokoll auf Sitzungsebene, über das sich zwei logische Einheiten (LUs) beim Aktivieren einer Sitzung gegenseitig authentifizieren können. Die Authentifizierung auf Sitzungsebene wird auch als LU-LU-Prüfung bezeichnet.

**Verschlüsselung auf Sitzungsebene**

In der Systemnetzwerkarchitektur (SNA) ein Verfahren zur Verschlüsselung und Entschlüsselung von Daten, die während einer Sitzung zwischen zwei logischen Einheiten (LUs) ausgetauscht werden.

**Gemeinsam genutzter eingehender Kanal**

In WebSphere MQ for z/OS ein Kanal, der von einem Empfangsprogramm über den Gruppenport gestartet wurde. Die Kanaldefinition eines gemeinsamen Kanals kann entweder in der Seitengruppe 0 (privat) oder im gemeinsamen Repository (global) gespeichert werden.

**Gemeinsam genutzter ausgehender Kanal**

In WebSphere MQ for z/OS ein Kanal, der Nachrichten aus einer gemeinsam genutzten Übertragungswarteschlange verschiebt. Die Kanaldefinition eines gemeinsamen Kanals kann entweder in der Seitengruppe 0 (privat) oder im gemeinsamen Repository (global) gespeichert werden.

**Gemeinsam genutzte Warteschlange**

In WebSphere MQ for z/OS ein spezieller Typ von lokaler Warteschlange. Die Nachrichten in der Warteschlange werden in der Coupling-Facility gespeichert und sind für einen oder mehrere Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange zugänglich. Die Definition der Warteschlange wird im gemeinsamen Repository gespeichert. Siehe auch [Gruppe mit gemeinsamer Warteschlange](#).

**Gemeinsames Repository**

In WebSphere MQ for z/OS, eine gemeinsam genutzte Db2-Datenbank, die zum Speichern von global definierten Objektdefinitionen verwendet wird.

**Gemeinsame Nutzung einer Kanalinstanz**

Die Funktion, die die gemeinsame Nutzung einer Kanalinstanz durch mehrere Dialoge ermöglicht, bzw. die Dialoge, die eine Kanalinstanz gemeinsam nutzen.

**shell**

Eine Softwareschnittstelle zwischen Benutzern und einem Betriebssystem. Es gibt grundsätzlich zwei Kategorien von Shells: eine Befehlszeilen-Shell, die eine Befehlszeilenschnittstelle für das Betriebssystem bereitstellt, und eine grafische Shell, die eine grafische Benutzerschnittstelle (GUI) bereitstellt.

**System-ID**

Siehe [Sicherheits-ID](#).

**Signal**

Ein Mechanismus, über den ein Prozess über ein Ereignis im System benachrichtigt oder von diesem betroffen sein kann. Beispiele für solche Ereignisse sind Hardwareausnahmebedingungen und bestimmte Aktionen von Prozessen.

**Signalisierung**

In WebSphere MQ for z/OS und WebSphere MQ for Windows eine Funktion, die es dem Betriebssystem ermöglicht, einem Programm die Ankunft einer erwarteten Nachricht in einer Warteschlange zu melden.

**Signatur**

Die Gruppe von Typen, die einer Methode zugeordnet sind. Die Signatur umfasst den Typ des Rückgabewerts, falls vorhanden, sowie die Nummer, die Reihenfolge und den Typ jedes einzelnen Arguments der Methode.

**Unterzeichnerzertifikat**

Ein Zertifikat, das zur Verschlüsselung oder Signierung verwendet wird.

**Einzelinstanz-Warteschlangenmanager**

Ein Warteschlangenmanager, von dem es nur eine einzige Instanz gibt. Siehe auch [Mehrinstanz-Warteschlangenmanager](#).

**Einzelprotokollierung**

Ein Verfahren zum Aufzeichnen der Aktivitäten in WebSphere MQ for z/OS, bei dem jede Änderung nur in eine einzige Datei geschrieben wird. Siehe auch [Doppelte Protokollierung](#).

**Einphasige Zurücksetzung**

Ein Verfahren, bei dem ein aktiver Vorgang nicht beendet werden darf, und alle Änderungen, die während des Vorgangs vorgenommen wurden, zurückgesetzt werden müssen.

**einphasiges Commit**

Ein Verfahren, bei dem ein Programm Aktualisierungen in einer Festschreibungsressource festschreiben kann, ohne dass diese Aktualisierungen mit den Änderungen koordiniert werden, die das Programm an Ressourcen vorgenommen hat, die von einem anderen Ressourcenmanager verwaltet werden.

**SIT (System Initialization Table)**

Siehe [Systeminitialisierungstabelle](#).

**SMF**

Siehe [System Management Facilities](#).

**SNA (Systems Network Architecture)**

Siehe [Systemnetzwerkarchitektur](#).

**Software-Development-Kit (SDK)**

Eine Sammlung von Tools, Anwendungsprogrammierschnittstellen und Dokumentationen zur Unterstützung der Entwicklung von Software in einer bestimmten Maschinensprache oder für eine bestimmte Betriebsumgebung.

**Quellen-Warteschlangenmanager**

Siehe [Lokaler Warteschlangenmanager](#).

**SP2**

Siehe [Scalable Parallel 2](#).

**SPX**

Siehe [Sequenced Packet Exchange-Protokoll](#).

**SSI**

Siehe [Security Support Provider Interface](#).

**SSL**

Siehe [Secure Sockets Layer](#).

**SSLPeer**

Der Wert im Aussteller entspricht dem registrierten Namen des fernen persönlichen Zertifikats.

**SSL- oder TLS-Client**

Die Seite einer Verbindung, die die Verbindung einleitet. Ein einziger Kanal für abgehende Nachrichten eines Warteschlangenmanagers ist auch ein SSL- oder TLS-Client.

**Standby-Warteschlangenmanagerinstanz**

Eine Instanz eines aktiven Multi-Instanz-Warteschlangenmanagers, die bereit ist, den Betrieb von der aktiven Instanz zu übernehmen. Es gibt eine oder mehrere Standby-Instanzen eines Multi-Instanz-Warteschlangenmanagers.

**Zeilengruppe**

Eine Gruppe von Zeilen in einer Datei, die eine gemeinsame Funktion haben oder einen Teil des Systems definieren. Zeilengruppen sind normalerweise durch Leerzeilen oder Doppelpunkte voneinander getrennt und jede Zeilengruppe hat einen Namen.

**Sternförmig verbundenes Kommunikationsnetz**

Ein Netz, in dem alle Knoten mit einem zentralen Knoten verbunden sind.

**Speicherklasse**

In WebSphere MQ for z/OS die Seitengruppe, in der die Nachrichten für eine bestimmte Warteschlange gespeichert werden sollen. Die Speicherklasse wird bei der Definition der Warteschlange angegeben.

**Store-and-forward-Verfahren**

Die temporäre Speicherung von Paketen, Nachrichten oder Rahmen in einem Datennetz, bevor sie erneut an ihr Ziel übertragen werden.

**Streaming**

In der objektorientierten Programmierung die serielle Verarbeitung von Klassendaten und Objektinstanzdaten.

**Abonnieren**

Die Anforderung von Informationen zu einem Thema.

**Subsystem**

In z/OS ein Serviceprovider, der eine oder viele Funktionen ausführt, aber erst auf Anforderung aktiv wird. Zum Beispiel ist jeder WebSphere MQ for z/OS-Warteschlangenmanager oder jede Instanz eines Db2 for z/OS-Datenbankmanagementsystems ein z/OS-Subsystem.

**Supervisoraufruf (SVC)**

Eine Anweisung, die ein laufendes Programm unterbricht und die Steuerung an den Supervisor übergibt, damit dieser den in der Anweisung angegebenen Service ausführen kann.

**SVC**

Siehe [Supervisoraufruf](#).

**Switchover**

Der Wechsel von der aktiven Instanz eines Multi-Instanz-Warteschlangenmanagers zu einer Standby-Instanz. Das Umschalten findet statt, wenn ein Bediener die aktive Instanz des Multi-Instanz-Warteschlangenmanagers absichtlich stoppt.

**Schalterprofil**

In WebSphere MQ for z/OS ein RACF-Profil, das verwendet wird, wenn WebSphere MQ gestartet oder ein Befehl zur Aktualisierung der Sicherheit ausgegeben wird. Jedes Schalterprofil, das WebSphere MQ erkennt, inaktiviert die Überprüfung der angegebenen Ressource.

**Symmetrische Verschlüsselung**

Ein Verschlüsselungssystem, bei dem der Sender und der Empfänger einer Nachricht einen einzigen, gemeinsamen und geheimen Schlüssel zum Ver- und Entschlüsseln der Nachricht verwenden. Dieses System ermöglicht keine Authentifizierung. Siehe auch [Asymmetrische Verschlüsselung](#).

**Symptomzeichenfolge**

Diagnoseinformationen, die in einem strukturierten Format angezeigt werden, das für die Suche in der Datenbank des IBM Software Support entworfen wurde.

**Synchrone Nachrichtenübertragung**

Eine Methode zur Kommunikation zwischen Programmen, bei der ein Programm eine Nachricht in eine Nachrichtenwarteschlange einreicht und dann auf eine Antwort auf die Nachricht wartet, bevor es die eigene Verarbeitung fortsetzt. Siehe auch [Asynchrone Nachrichtenübertragung](#).

**Synchronisationspunkt**

Ein Punkt während der Verarbeitung einer Transaktion, an dem geschützte Ressourcen konsistent sind.

**Sysplex**

Eine Gruppe von z/OS-Systemen, die über bestimmte Multisystem-Hardwarekomponenten und Softwareservices miteinander kommunizieren.

**Systembehälter**

Ein Typ von Datenbehälter, der von MQAI erstellt wird.

**Systemsteuerbefehle**

Befehle, die zur Manipulation plattformspezifischer Einheiten wie beispielsweise Pufferpools, Speicherklassen und Seitengruppen verwendet werden.

**Arbeitsbereich für Systemdiagnose**

In einer z/OS-Umgebung die Daten, die in einem SYS1.LOGREC-Eintrag aufgezeichnet werden, der einen Programm- oder Hardwarefehler beschreibt.

**Systeminitialisierungstabelle**

Eine Tabelle mit Parametern, die CICS beim Start verwendet.

**Systemelement**

Ein Datenelementtyp, der von MQAI erstellt wird.

**System Management Facilities (SMF)**

Eine z/OS-Komponente, die eine Vielzahl von system- und jobbezogenen Informationen erfasst und aufzeichnet.

**Systemselektor**

In WebSphere MQ Administration Interface (MQAI) eine Systemelement-ID, die bei der Erstellung des Systemelements in den Datenbehälter aufgenommen wird.

**Systemnetzwerkarchitektur (SNA)**

Die Beschreibung der logischen Struktur, Formate, Protokolle und Verarbeitungsfolgen für die Übertragung von Informationen in Netzen und die Steuerung der Konfiguration und des Betriebs von Netzen.

**T****Manipulationen**

Ein Verstoß gegen die Kommunikationssicherheit, bei dem in der Übertragung befindliche Informationen geändert oder ausgetauscht und dann an den Empfänger gesendet werden. Siehe auch [Abhören](#) und [Identitätsvortäuschung](#).

**Übergeordnetes Qualifikationsmerkmal für Zielbibliothek (thlqual)**

Ein übergeordnetes Qualifikationsmerkmal für z/OS-Zieldateinamen.

**Zielwarteschlangenmanager**

Siehe [Ferner Warteschlangenmanager](#).

**Tasksteuerblock**

Ein z/OS-Steuerblock, der zur Übertragung von Informationen zu Tasks in einem Adressraum, der mit einem Subsystem verbunden ist, verwendet wird.

**Wechsel zwischen Tasks**

Die Überschneidung von E/A- und Verarbeitungsoperationen zwischen mehreren Tasks.

**TCB**

Siehe [Tasksteuerblock](#).

**TCP**

Siehe [Transmission Control Protocol](#).

**TCP/IP**

Siehe [Transmission Control Protocol/Internet Protocol](#).

**Technische Hinweise**

Ein kurzes Dokument zu einem einzelnen Thema.

**Telemetriekanal**

Ein Telemetriekanal ist eine Kommunikationsverbindung zwischen einem Warteschlangenmanager in WebSphere MQ und MQTT-Clients. Mit jedem Kanal können ein oder mehrere Telemetriegeräte verbunden sein.

**Telemetrieclient (erweitert)**

Der erweiterte Telemetrieclient wird im Unterordner mqxr der WebSphere MQ-Hauptinstallation installiert. Dank des geringen Speicherbedarfs ermöglichen MQTT-Server es mehreren MQTT-Clients, eine Verbindung zum Client herzustellen und einen Uplink oder eine Bridge zu WebSphere MQ bereitzustellen. Erweiterte Clients können Nachrichten im Namen von Clients starten, wenn die Uplinkverbindung unterbrochen ist.

**Telemetrieclient**

Telemetrieclients sind MQTT-Clients, die im Unterordner mqxr der WebSphere MQ-Hauptinstallation installiert werden. Sie stellen über das MQTT-Protokoll eine Verbindung zu MQ her.

**Telemetrieservice (MQXR)**

Ein MQ-Service, der für die Serverseite des MQTT-Protokolls zuständig ist (siehe MQTT-Server). Der Telemetrieservice (MQXR) betreibt Telemetriekanäle.

**Temporäre dynamische Warteschlange**

Eine dynamische Warteschlange, die beim Schließen gelöscht wird. Temporäre dynamische Warteschlangen werden nach dem Ausfall des Warteschlangenmanagers nicht wiederhergestellt, d. h., sie können nur nicht persistente Nachrichten enthalten. Siehe auch [Permanente dynamische Warteschlange](#).

**Teraspace**

Ein temporärer Speicherbereich mit einer Größe von einem Terabyte, der als privater Speicher für einen Prozess dient.

**Beendigungsmeldung**

Ein anstehendes Ereignis, das aktiviert wird, wenn ein CICS-Subsystem erfolgreich eine Verbindung zu WebSphere MQ for z/OS herstellt.

**thlqual**

Siehe [Übergeordnetes Qualifikationsmerkmal für Zielbibliothek](#).

**Thread**

Ein Datenstrom aus Maschineninstruktionen, der einen Prozess steuert. In einigen Betriebssystemen ist ein Thread die kleinste Operationseinheit in einem Prozess. Mehrere Threads können gleichzeitig aktiv sein und unterschiedliche Jobs ausführen.

**TID**

Siehe [Transaktions-ID](#).

**Zeitunabhängige Nachrichtenübertragung**

Siehe [Asynchrone Nachrichtenübertragung](#).

**TLS**

Transport Layer Security (Nachfolger von SSL)

**TMF**

Siehe [Transaction Manager Facility](#).

**TMI (Trigger Monitor Interface)**

Siehe [Auslösemonitorschnittstelle](#).



**TP**

Siehe [Transaktionsprogramm](#).

**trace**

Die Aufzeichnung der Verarbeitung eines Computerprogramms oder einer Transaktion. Die in einem Trace erfassten Informationen können zur Beurteilung von Problemen und des Leistungsverhaltens verwendet werden.

**Transaktionskennung**

Siehe [Transaktions-ID](#).

**Transaktions-ID (TID, XID)**

Ein eindeutiger Name, der einer Transaktion zugewiesen wird und die der Transaktion zugeordneten Aktionen identifiziert.

**Transaktionsmanager**

Eine Softwareeinheit, die die Aktivitäten von Ressourcenmanagern koordiniert, indem sie globale Transaktionen verwaltet und die Entscheidung über Festschreibung (Commit) oder Zurücksetzung (Rollback) der Transaktionen koordiniert.

**Transaction Manager Facility (TMF)**

In IBM WebSphere MQ for HP Integrity NonStop Server ein Subsystem zum Schutz von Geschäftstransaktionen und der Integrität von Datenbanken. Wird häufig als Synonym für NonStop Transaction Manager/MP gebraucht.

**Transaktionsprogramm (TP)**

Ein Programm, das Transaktionen in einem SNA-Netz verarbeitet.

**Transmission Control Protocol (TCP)**

Ein Kommunikationsprotokoll im Internet und in anderen Netzen, das den IETF-Standards (Internet Engineering Task Force) für Internetprotokolle entspricht. TCP stellt ein zuverlässiges Host-to-Host-Protokoll in DFV-Netzen mit Paketvermittlung und in miteinander verbundenen Systemen in solchen Netzen bereit. Siehe auch [Internet Protocol](#).

**Transmission Control Protocol/Internet Protocol (TCP/IP)**

Eine standardisierte, nicht proprietäre Gruppe von Kommunikationsprotokollen, die zuverlässige End-to-End-Verbindungen zwischen Anwendungen über miteinander verbundene Netze unterschiedlichen Typs bereitstellt.

**Übertragungsprogramm**

Siehe [Nachrichtenkanalagent](#).

**Übertragungswarteschlange**

Eine lokale Warteschlange, in der vorbereitete Nachrichten für einen fernen Warteschlangenmanager temporär gespeichert werden.

**Warteschlange mit aktivierten Auslösern**

Eine lokale Warteschlange, die erfordert, dass Auslösenachrichten geschrieben werden, wenn Auslöser aktiviert und Auslösebedingungen erfüllt sind.

**Auslöse- ereignis**

Ein Ereignis, z. B. der Eingang einer Nachricht in einer Warteschlange, das einen Warteschlangenmanager veranlasst, eine Auslösenachricht in einer Initialisierungswarteschlange zu erstellen.

**Triggering**

In WebSphere MQ eine Funktion, die es einem Warteschlangenmanager ermöglicht, eine Anwendung automatisch zu starten, wenn vordefinierte Bedingungen in einer Warteschlange erfüllt sind.

**Auslöse- nachricht**

Eine Nachricht mit Informationen zu dem Programm, das von einem Auslösemonitor gestartet werden soll.

**Auslösemonitor**

Eine ständig aktive Anwendung, die eine oder mehrere Initialisierungswarteschlangen bedient. Sobald eine Auslösenachricht in einer Initialisierungswarteschlange ankommt, wird sie vom Auslösemonitor abgerufen. Er verwendet die Informationen in der Auslösenachricht, um einen Prozess zu starten, der die Warteschlange bedient, in der ein Auslöserereignis stattgefunden hat.

**Auslösemonitorschnittstelle (TMI)**

Die WebSphere-MQ-Schnittstelle, der von Kunden oder Anbietern geschriebene Auslösemonitoranwendungen entsprechen müssen. Ein Teil des WebSphere MQ-Frameworks.

**Truststore**

Der Ort, an dem Zertifikate von Zertifizierungsstellen abgelegt werden, um Zertifikate von einem fernen System zu prüfen. Siehe auch [Schlüsselspeicher](#).

**Bidirektionale Authentifizierung**

Bei dieser Authentifizierungsmethode legen sich der Warteschlangenmanager und der Client die Zertifikate gegenseitig vor. Wird auch als gegenseitige Authentifizierung bezeichnet.

**Zweiphasige Festschreibung**

Ein aus zwei Schritten bestehender Prozess, durch den wiederherstellbare Ressourcen und ein externes Subsystem festgeschrieben werden. Im ersten Schritt werden die Subsysteme des Datenbankmanagers abgefragt, um sicherzustellen, dass sie festgeschrieben werden können. Wenn alle Subsysteme positiv antworten, weist der Datenbankmanager ihre Festschreibung an.

**Typ**

Ein Merkmal, das das interne Format von Daten und deren Verwendungsmöglichkeiten festlegt.

**U****UDP**

Siehe [User Datagram Protocol](#).

**Unbefugter Zugriff**

Der unberechtigte Zugriff auf Ressourcen in einem Computersystem.

**Warteschlange für nicht zustellbare Nachrichten**

Siehe [Warteschlange für nicht zustellbare Nachrichten](#).

**Undo/Redo-Datensatz**

Ein Protokolleintrag, der für die Wiederherstellung verwendet wird. Der Redo-Teil (Widerruf zurücknehmen) des Datensatzes beschreibt eine Änderung, die an einem WebSphere MQ-Objekt vorgenommen werden muss. Der Undo-Teil (Widerrufen) beschreibt, wie die Änderung zurückgesetzt werden muss, wenn die Arbeitseinheit nicht festgeschrieben wird.

**Arbeitseinheit mit Wiederherstellung**

Eine wiederherstellbare Operationsfolge in einem einzelnen Ressourcenmanager, z. B. in einer Instanz von Db2 for z/OS. Siehe auch [Arbeitseinheit](#).

**Arbeitseinheit**

Eine wiederherstellbare Operationsfolge, die von einer Anwendung zwischen zwei Konsistenzzuständen ausgeführt wird. Eine Arbeitseinheit beginnt mit dem Start einer Transaktion oder an einem vom Benutzer angeforderten Synchronisationspunkt. Sie endet entweder an einem vom Benutzer angeforderten Synchronisationspunkt oder mit dem Abschluss einer Transaktion.

**UOW**

Siehe [Arbeitseinheit](#).

**Benutzerbehälter**

In WebSphere MQ Administration Interface (MQAI) ein Typ von Datenbehälter, der vom Benutzer erstellt wird.

**User Datagram Protocol (UDP)**

Ein Internetprotokoll, das einen störanfälligen, verbindungsunabhängigen Datagrammservice bereitstellt. Es ermöglicht einem Anwendungsprogramm auf einer Maschine oder in einem Prozess, ein Datagramm an ein Anwendungsprogramm auf einer anderen Maschine oder in einem anderen Prozess zu senden.

**Benutzerelement**

In WebSphere MQ Administration Interface (MQAI) ein Typ von Datenelement, das vom Benutzer erstellt wird.

### **Benutzerselektor**

In WebSphere MQ Administration Interface (MQAI) die Kennung, die mit einem Datenelement in einem Datenbehälter gespeichert wird, um das Datenelement zu identifizieren. WebSphere MQ stellt vordefinierte Benutzerselektoren für WebSphere MQ-Objekte bereit.

### **Benutzertoken (UTOKEN)**

Das RACF-Sicherheitstoken, das die Sicherheitsmerkmale eines Benutzers kapselt oder darstellt. RACF weist jedem Benutzer im System ein Benutzertoken zu.

### **Dienstprogramm**

Eine in WebSphere MQ bereitgestellte Gruppe von Programmen, die dem Systembediener oder Systemadministrator Funktionen zusätzlich zu den von den WebSphere-MQ-Befehlen bereitgestellten Funktionen bereitstellen.

### **UTOKEN**

Siehe [Benutzertoken](#).

## **V**

### **Wert**

Der Inhalt eines Datenelements. Dies kann eine ganze Zahl, eine Zeichenfolge oder die Kennung eines anderen Datenbehälters sein.

### **Virtuelle Methode**

In der objektorientierten Programmierung eine Methode, die Polymorphie unterstützt.

## **W**

### **WebSphere MQ**

Eine aus IBM Lizenzprogrammen bestehende Produktfamilie, die Message-Queuing-Services bereitstellt.

### **WebSphere MQ Administration Interface (MQAI)**

Eine Programmierschnittstelle, über die mithilfe von Datenbehältern Verwaltungsaufgaben für einen WebSphere MQ-Warteschlangenmanager ausgeführt werden. Datenbehälter ermöglichen dem Benutzer die Bearbeitung von Eigenschaften (oder Parametern) von WebSphere-MQ-Objekten.

### **WebSphere MQ-Klassen für .NET**

Eine Gruppe von Klassen, die einem im Programmierframework .NET geschriebenen Programm ermöglichen, als WebSphere MQ-Client eine Verbindung zu WebSphere MQ oder eine direkte Verbindung zu einem WebSphere MQ-Server herzustellen.

### **WebSphere MQ-Klassen für C++**

Eine Gruppe von Klassen, die die WebSphere MQ Message Queue Interface (MQI) in der Programmiersprache C++ kapseln.

### **WebSphere MQ-Klassen für Java**

Eine Gruppe von Klassen, die die WebSphere MQ Message Queue Interface (MQI) in der Programmiersprache Java kapseln.

### **Vollständig verwalteter WebSphere MQ-.NET-Client**

Eine Komponente des WebSphere MQ-Produkts, die auf einem System installiert werden kann, ohne den vollständigen Warteschlangenmanager zu installieren. Der WebSphere MQ-.NET-Client wird von vollständig verwalteten .NET-Anwendungen verwendet und kommuniziert mit einem Warteschlangenmanager auf einem Serversystem. Nicht vollständig verwaltete .NET-Anwendungen verwenden den WebSphere MQ MQI-Client. Siehe auch [Client](#), [WebSphere MQ-Client](#) und [WebSphere MQ-Java-Client](#).

### **WebSphere MQ-Java-Client**

Eine Komponente des WebSphere MQ-Produkts, die auf einem System installiert werden kann, ohne den vollständigen Warteschlangenmanager zu installieren. Der WebSphere MQ Java-Client wird von Java-Anwendungen verwendet (WebSphere MQ-Klassen für Java und WebSphere MQ-Klassen für JMS) und kommuniziert mit einem Warteschlangenmanager auf einem Serversystem. Siehe auch [Client](#), [WebSphere MQ-Client](#) und [Vollständig verwalteter WebSphere MQ-.NET-Client](#).

### **WebSphere MQ- MQI-Client**

Eine Komponente des WebSphere MQ-Produkts, die auf einem System installiert werden kann, ohne den vollständigen Warteschlangenmanager zu installieren. Der WebSphere-MQ-Client akzeptiert MQI-Aufrufe von Anwendungen und kommuniziert mit einem Warteschlangenmanager auf einem Serversystem. Siehe auch [Client](#), [WebSphere MQ-Java-Client](#) und [Vollständig verwalteter WebSphere MQ-.NET-Client](#).

### **WebSphere MQ-Scriptbefehle (MQSC)**

Plattformübergreifende lesbare Befehle, mit denen WebSphere MQ-Objekten bearbeitet werden können. Siehe auch [Programmable Command Format](#).

### **WebSphere MQ-Server**

Ein Warteschlangenmanager, der Queuing-Services für einen oder mehrere Clients bereitstellt. Alle WebSphere-MQ-Objekte, z. B. Warteschlangen, sind nur auf dem Warteschlangenmanagersystem, d. h. auf der MQI-Servermaschine, vorhanden. Ein Server kann auch normale lokale MQI-Anwendungen unterstützen.

### **WebSphere MQ Telemetry**

WebSphere MQ Telemetry stellt kleine Clientbibliotheken bereit, die in intelligente Geräte, die auf vielen unterschiedlichen Geräteplattformen aktiv sind, integriert werden können. Anwendungen, die mit den Clients erstellt werden, verwenden MQ Telemetry Transport (MQTT) und den WebSphere MQ Telemetrieservice (MQXR), um in Verbindung mit WebSphere MQ Nachrichten zuverlässig zu veröffentlichen und zu subscribieren. Wenn die angepasste WebSphere MQ-Installationsoption zur Installation der Telemetrie ausgewählt wird, wird folgendes installiert: 1) Telemetrieservice (MQXR), 2) Telemetryclients und 3) Erweiterte Telemetryclients.

### **WebSphere MQ Telemetry-Dämon für Geräte**

Der WebSphere MQ Telemetry-Dämon für Geräte ist ein erweiterter MQTT-Client (MQTT Version 3). Dieser MQTT-Server mit sehr geringem Speicherbedarf wurde für eingebettete Systeme entwickelt.

### **Windows NT Challenge/Response**

Das Authentifizierungsprotokoll, das in Netzen mit Windows NT-Systemen und auf eigenständigen Systemen verwendet wird.

### **Anzapfen von Übertragungsleitungen**

Der Zugriff auf Informationen, die über eine verdrahtete Verbindung oder einen anderen für die Datenübertragung verwendeten Leiter übertragen werden. Ziel des Anzapfens von Übertragungsleitungen ist es, sich unentdeckt unbefugten Zugriff auf Informationen zu verschaffen.

## **X**

### **X509**

ITU-T-Norm für PKI-Infrastruktur. Legt das Format des Public-Key-Zertifikats und die Public-Key-Verschlüsselung fest.

### **XCF**

Siehe [Cross-System Coupling-Facility](#).

### **XID**

Siehe [Transaktions-ID](#).

### **X/Open XA**

Die X/Open Distributed Transaction Processing XA-Schnittstelle. Ein empfohlener Standard für die Kommunikation bei verteilten Transaktionen. Der Standard spezifiziert eine bidirektionale Schnittstelle zwischen Ressourcenmanagern, die Zugriff auf gemeinsame Ressourcen in Transaktionen bereitstellen, und einem Transaktionsservice, der Transaktionen überwacht und auflöst.

## **Funktionen zur behindertengerechten Bedienung für IBM WebSphere MQ**

Funktionen zur behindertengerechten Bedienung (Eingabehilfefunktionen) unterstützen Benutzer mit einer Behinderung, wie eingeschränkter Mobilität oder Sehbehinderung, damit sie Inhalte der Informationstechnologie erfolgreich verwenden können.

## Funktionen zur behindertengerechten Bedienung

IBM WebSphere MQ umfasst die folgenden Funktionen zur behindertengerechten Bedienung:

- Bedienung ausschließlich über die Tastatur
- Operationen mit Sprachausgabeprogramm

IBM WebSphere MQ verwendet den neuesten W3C-Standard, WAI-ARIA 1.0 (<https://www.w3.org/TR/wai-aria/>), um die Einhaltung von US Section 508 (<https://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards>) und den Web Content Accessibility Guidelines (WCAG) 2.0 (<https://www.w3.org/TR/WCAG20/>) zu gewährleisten. Um die Vorteile der Funktionen zur behindertengerechten Bedienung nutzen zu können, sollten Sie das neueste Release Ihres Sprachausgabeprogramms in Verbindung mit dem neuesten Web-Browser verwenden, der von diesem Produkt unterstützt wird.

Die IBM WebSphere MQ Online-Produktinformationen in IBM Documentation sind für die behindertengerechte Bedienung aktiviert. Die Funktionen zur behindertengerechten Bedienung von IBM Documentation werden unter <https://www.ibm.com/docs/about/releasesnotes.html> beschrieben.

## Tastaturnavigation

Dieses Produkt verwendet Standardnavigationstasten.

## Schnittstelleninformationen

Über die Befehlszeilenschnittstelle bietet IBM WebSphere MQ das vollständige Spektrum der Eingabehilfen. Weitere Informationen zur Verwendung von Befehlen finden Sie in den Abschnitten [IBM WebSphere MQ-Steuerbefehle verwenden](#) und [Verwaltung mithilfe von MQSC-Befehlen](#).

Für Windows kann IBM WebSphere MQ über eine nicht interaktive Installation installiert werden. Weitere Informationen finden Sie im Abschnitt [Erweiterte Installation mit msiexec](#).

Die IBM WebSphere MQ-Benutzerschnittstellen haben keine Inhalte, die 2-55 Mal pro Sekunde blinken.

Die Webbenutzerschnittstelle von IBM WebSphere MQ setzt für die korrekte Wiedergabe der Inhalte und für ein ansehnliches Ergebnis keine Cascading Style Sheets voraus. Zur korrekten Anzeige der Produktdokumentation sind hingegen Cascading Style Sheets erforderlich. IBM WebSphere MQ bietet eine funktional entsprechende Möglichkeit für Benutzer mit eingeschränktem Sehvermögen, um die Systemanzeigeeinstellungen eines Benutzers zu verwenden, einschließlich des Modus für kontraststarke Anzeige. Die Schriftgröße kann mit den Geräte- oder Browsereinstellungen gesteuert werden.

## Zusätzliche Informationen zur behindertengerechten Bedienung

Neben dem herkömmlichen IBM Help-Desk und den üblichen Support-Websites hat IBM für seine Kunden mit Hörbeeinträchtigung einen TTY-Telefonservice eingerichtet, über den diese Kunden Vertriebs- und Supportleistungen erhalten:

TTY-Service  
800-IBM-3383 (800-426-3383)  
(in Nordamerika)

## IBM und behindertengerechte Bedienung

Weitere Informationen über den Einsatz von IBM für behindertengerechte Bedienung finden Sie unter [IBM Accessibility\(www.ibm.com/able\)](http://www.ibm.com/able).

## Eingabehilfen unter Windows

IBM WebSphere MQ-Benutzerschnittstellen verwenden keine speziellen Tastenkombinationen, sondern folgen den Richtlinien für Windows-Benutzerschnittstellen bezüglich Direktaufufen für Elemente wie

Kontextmenüs, Dialoge und Dialogsteuerelemente wie z. B. Schaltflächen. Der Zugriff auf die Direktauf-  
rufe erfolgt auf die übliche Weise. Weitere Informationen finden Sie in der Windows-Hilfe (suchen Sie im  
Windows-Hilfeindex nach *Tastatur*; Informationen zu Eingabehilfefunktionen finden Sie unter *Eingabehil-  
fen*).

## Spezielle Eingabehilfefunktionen

Einige Benutzerschnittstellen in IBM WebSphere MQ werden normal dargestellt, ändern jedoch ihr Verhal-  
ten, wenn Eingabehilfen aktiviert sind:

- Hochkontrastmodus

In diesem Modus werden in den Anzeigen 'Launchpad', 'IBM WebSphere MQ-Vorbereitungsassistent',  
'PostCard' und 'Standardkonfiguration' die Hintergrundbitmaps ausgeblendet. Stellen Sie sicher, dass  
die Systemtextfarben verwendet werden, damit sie leicht zu erkennen und lesbar sind.

- Modus für Sprachausgabeprogramm

Wenn ein Sprachausgabeprogramm aktiviert ist, werden die Anzeigen 'IBM WebSphere MQ-Vorberei-  
tungsassistent', 'Standardkonfiguration' und 'PostCard' vereinfacht dargestellt. Hintergrundbitmaps,  
Hervorhebungseffekte, schattierte Felder usw., die sich möglicherweise störend auf das Sprachausga-  
beprogramm auswirken, werden ausgeblendet.

- Objektstatus im Explorer

Der IBM WebSphere MQ Explorer verwendet Symbole zum Anzeigen des Objektstatus von Objekten (z.  
B. Warteschlangenmanager). Da Sprachausgabeprogramme diese Symbole nicht interpretieren können,  
kann optional die Beschreibung des jeweiligen Symbols angezeigt werden. Um diese Option auszuwäh-  
len, klicken Sie im Explorer auf **Fenster > Benutzervorgaben > WebSphere MQ Explorer** und wählen  
Sie **Status von Objekten nach Objektname anzeigen** aus.

## Bemerkungen

---

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieser Dokumentation ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe  
IBM Europe, Middle East and Africa  
Tour Descartes  
2, avenue Gambetta  
92066 Paris La Défense  
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Lizenzierung von geistigem Eigentum

IBM Japan, Ltd.

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in dieser Veröffentlichung werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East and Africa  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Die in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Um diese so realistisch wie möglich zu gestalten, enthalten sie auch Namen von Personen, Firmen, Marken und Produkten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musterprogramme, die in Quellensprache geschrieben sind. Sie dürfen diese Musterprogramme kostenlos (d. h. ohne Zahlung an IBM) kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farabbildungen.

## Informationen zu Programmierschnittstellen

---

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

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

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

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



## Marken

---

IBM, das IBM Logo, ibm.com, sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml). Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein.

Microsoft und Windows sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Dieses Produkt enthält Software, die von Eclipse Project (<http://www.eclipse.org/>) entwickelt wurde.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.







Teilenummer:

(1P) P/N: