

9.1

*Überwachung und Leistung für IBM MQ*

**IBM**

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 393 gelesen werden.

Diese Ausgabe bezieht sich auf Version 9 Release 1 von IBM® MQ und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuauflage 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

- Überwachung und Leistung.....5**
  - IBM MQ-Netz überwachen..... 5
    - Ereignisüberwachung..... 5
    - Nachrichtenüberwachung..... 59
    - Abrechnungs-und Statistiknachrichten.....141
    - Anwendungsaktivitätstrace..... 208
    - Systemthemen für Überwachung und Aktivitätstrace..... 298
    - IBM MQ Bridge to Salesforce überwachen.....306
    - Real-time Monitoring..... 310
    - Cluster überwachen..... 323
    - Lastausgleich von Anwendungen überwachen..... 326
    - Überwachen der Leistung und Ressourcennutzung.....331
  - IBM MQ-Netz optimieren..... 378
    - Tuning Client-und Serververbindungskanäle..... 378
    - Verteilte Publish/Subscribe-Netze optimieren..... 380
    - Reduzieren der Anzahl unerwünschter Themen in der Themenstruktur..... 390
    - Aspera gateway kann die Leistung in Netzen mit hoher Latenz verbessern..... 392
  
- Bemerkungen..... 393**
  - Informationen zu Programmierschnittstellen..... 394
  - Marken..... 395



# IBM MQ Überwachung und Leistung

---

Verwenden Sie die Überwachungsinformationen und die Anleitung in diesem Abschnitt sowie die spezifischen Optimierungstipps, um die Leistung Ihres Warteschlangenmanagernetzes zu verbessern.

## Informationen zu diesem Vorgang

Abhängig von der Größe und Komplexität Ihres Warteschlangenmanagernetzes können Sie eine Reihe von Informationen von der Überwachung des Netzes abrufen. Sie können diese Informationen zusammen mit den in bestimmten Optimierungstipps bereitgestellten Informationen verwenden, um Sie bei der Optimierung der Netzleistung zu unterstützen.

## IBM MQ-Netz überwachen

---

In IBM MQ sind eine Reihe von Überwachungstechniken verfügbar, um Statistikdaten und andere spezifische Informationen darüber zu erhalten, wie Ihr Warteschlangenmanagernetz ausgeführt wird. Verwenden Sie die Überwachungsinformationen und Anleitungen in diesem Abschnitt, um die Leistung Ihres Warteschlangenmanagernetzes zu verbessern.

Die folgende Liste enthält Beispiele für Gründe für die Überwachung Ihres Warteschlangenmanagernetzes:

- Erkennen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Unterstützung bei der Bestimmung der Ursachen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Verbessern Sie die Effizienz Ihres Warteschlangenmanagernetzes.
- Machen Sie sich mit der Ausführung Ihres WS-Manager-Netzes vertraut.
- Stellen Sie sicher, dass Ihr Warteschlangenmanagernetzwerk ordnungsgemäß ausgeführt wird.
- Generiert Nachrichten, wenn bestimmte Ereignisse auftreten.
- Nachrichtenaktivität aufzeichnen.
- Bestimmen Sie die letzte bekannte Position einer Nachricht.
- Überprüfen Sie die verschiedenen Statistiken eines Warteschlangenmanagernetzes in Echtzeit.
- Generieren Sie einen Prüfprotokoll.
- Konto für Anwendungsressourcennutzung.
- Kapazitätsplanung.

## Ereignisüberwachung

Die Ereignisüberwachung ist der Prozess zum Erkennen von Vorkommen von *Instrumentierungsereignissen* in einem WS-Manager-Netz. Ein Instrumentierungsereignis ist eine logische Kombination von Ereignissen, die von einem Warteschlangenmanager oder einer Kanalinstanz erkannt wird. Ein solches Ereignis bewirkt, dass der Warteschlangenmanager oder die Kanalinstanz eine spezielle Nachricht, die als *Ereignisnachricht* bezeichnet wird, in eine Ereigniswarteschlange eingibt.

IBM MQ-Instrumentierungsereignisse stellen Informationen zu Fehlern, Warnungen und anderen wichtigen Vorkommnissen in einem Warteschlangenmanager bereit. Verwenden Sie diese Ereignisse, um die Operation der Warteschlangenmanager in Ihrem WS-Manager-Netz zu überwachen, um die folgenden Ziele zu erreichen:

- Erkennen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Unterstützung bei der Bestimmung der Ursachen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Generieren Sie einen Prüfprotokoll.

- Änderungen am Status des Warteschlangenmanagers ändern

### **Zugehörige Verweise**

„Ereignistypen“ auf Seite 8

Verwenden Sie diese Seite, um die Typen von Instrumentierungsereignis anzuzeigen, die ein Warteschlangenmanager oder eine Kanalinstanz melden kann.

Ereignisnachrichtenreferenz

Ereignisnachrichtenformat

### **Instrumentierungsereignisse**

Ein Instrumentierungsereignis ist eine logische Kombination von Bedingungen, die ein Warteschlangenmanager oder eine Kanalinstanz feststellt und in eine Ereigniswarteschlange eine spezielle Nachricht, die so genannte *Ereignisnachricht*, einreicht.

IBM MQ-Instrumentierungsereignisse stellen Informationen zu Fehlern, Warnungen und anderen wichtigen Vorkommnissen in einem Warteschlangenmanager bereit. Sie können mit diesen Ereignissen die Operation von Warteschlangenmanagern überwachen (mit anderen Methoden wie beispielsweise Tivoli NetView für z/OS).

Abbildung 1 auf Seite 7 veranschaulicht das Konzept der Instrumentierungsereignisse.

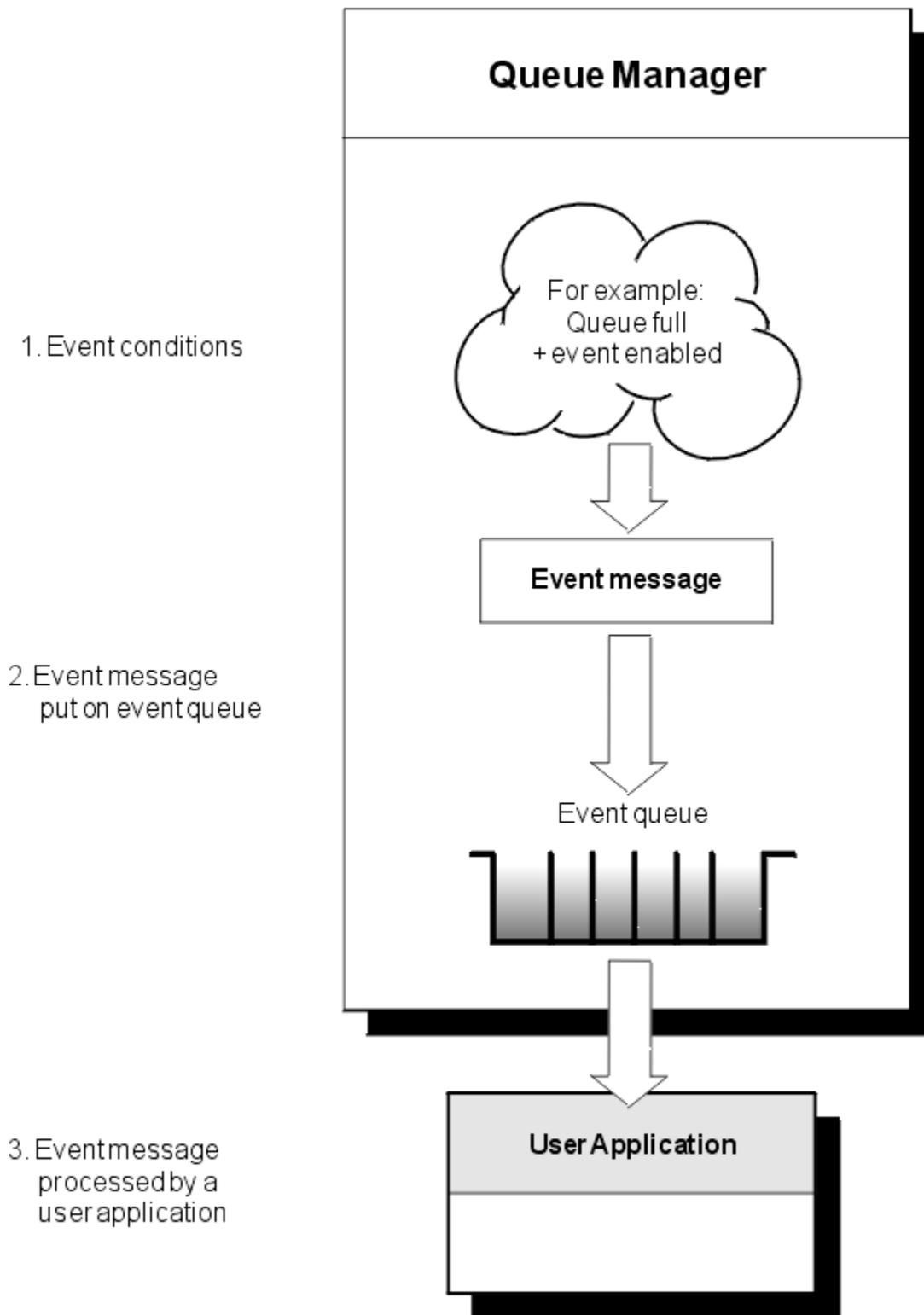


Abbildung 1. Informationen zu Instrumentierungs

### Ereignisüberwachungsanwendungen

Anwendungen, die Ereignisse für die Überwachung von Warteschlangenmanagern verwenden, müssen die folgenden Bestimmungen enthalten:

1. Richten Sie Kanäle zwischen den Warteschlangenmanagern in Ihrem Netz ein.

2. Implementieren Sie die erforderlichen Datenkonvertierungen. Es gelten die normalen Regeln für die Datenkonvertierung. Wenn Sie beispielsweise Ereignisse auf einem Warteschlangenmanager eines UNIX-Systems aus einem z/OS-Warteschlangenmanager überwachen, stellen Sie sicher, dass EBCDIC in ASCII umgewandelt wurde.

## Ereignisbenachrichtigung über Ereigniswarteschlangen

Wenn ein Ereignis eintritt, reiht der Warteschlangenmanager eine Ereignisnachricht in die entsprechende Ereigniswarteschlange ein, sofern diese definiert ist. Die Ereignisnachricht enthält Informationen zu dem Ereignis, das Sie abrufen können, indem Sie ein geeignetes MQI-Anwendungsprogramm schreiben, das die folgenden Schritte ausführt:

- Rufen Sie die Nachricht aus der Warteschlange ab.
- Verarbeiten Sie die Nachricht, um die Ereignisdaten zu extrahieren.

Die zugehörigen Informationen beschreiben das Format von Ereignisnachrichten.

## Bedingungen, die Ereignisse verursachen

Die folgende Liste enthält Beispiele für Bedingungen, die Instrumentierungsereignisse auslösen können:

- Es wird ein Schwellenwert für die Anzahl der Nachrichten in einer Warteschlange erreicht.
- Eine Kanalinstanz wird gestartet oder gestoppt.
- Ein WS-Manager wird aktiv oder wird zum Stoppen aufgefordert.
- Eine Anwendung versucht, eine Warteschlange mit einer Benutzer-ID zu öffnen, die auf IBM MQ for IBM i-, Windows-, UNIX and Linux®-Systemen nicht berechtigt ist.
- Objekte werden erstellt, gelöscht, geändert oder aktualisiert.
- Ein MQSC- oder PCF-Befehl wird erfolgreich ausgeführt.
- Ein Warteschlangenmanager beginnt mit dem Schreiben in einen neuen Protokollspeicherbereich.
- Wenn die Ereignisbedingungen erfüllt sind, wird eine Nachricht in die Warteschlange für dead-letter (dead-letter) ausgegeben.

### Zugehörige Konzepte

„Durchsatzereignisse“ auf Seite 21

Leistungsereignisse beziehen sich auf Bedingungen, die sich auf die Leistung von Anwendungen auswirken können, die eine angegebene Warteschlange verwenden. Der Umfang der Leistungsereignisse ist die Warteschlange. **MQPUT** -Aufrufe und **MQGET** -Aufrufe in einer Warteschlange haben keine Auswirkungen auf die Generierung von Leistungsereignissen in einer anderen Warteschlange.

„Beispielprogramm zur Überwachung von Instrumentierungsereignissen auf Multiplatforms“ auf Seite 56

**amqsevt** formatiert die Instrumentierungsereignisse, die ein Warteschlangenmanager erstellen kann, und wird mit IBM MQ for Multiplatforms bereitgestellt. Das Programm liest Nachrichten aus Ereigniswarteschlangen und formatiert sie in lesbare Zeichenfolgen.

## Ereignistypen

Verwenden Sie diese Seite, um die Typen von Instrumentierungsereignis anzuzeigen, die ein Warteschlangenmanager oder eine Kanalinstanz melden kann.

IBM MQ-Instrumentierungsereignisse weisen die folgenden Typen auf:

- WS-Manager-Ereignisse
- Kanal- und Brückenereignisse
- Durchsatzereignisse
- Konfigurationsereignisse
- Befehlsereignisse
- Ereignisse der Protokollfunktion

- Lokale Ereignisse

Für jeden WS-Manager hat jede Ereigniskategorie eine eigene Ereigniswarteschlange. Alle Ereignisse in dieser Kategorie führen dazu, dass eine Ereignisnachricht in die gleiche Warteschlange gestellt wird.

**Diese Ereigniswarteschlange:**

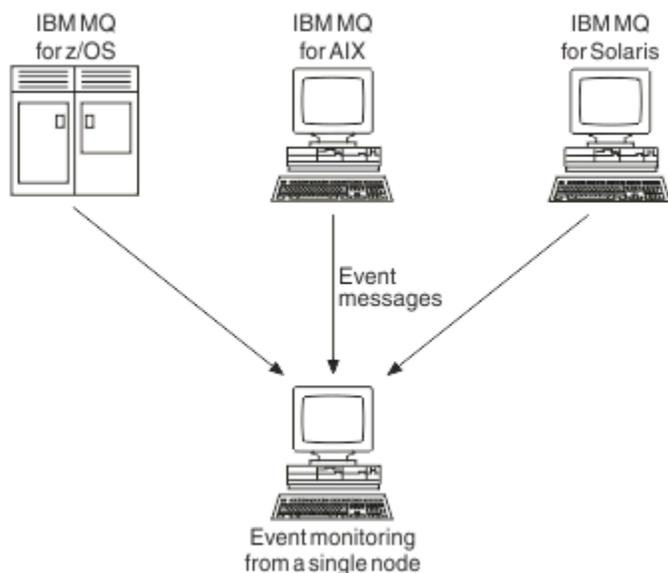
SYSTEM.ADMIN.QMGR.EVENT  
 SYSTEM.ADMIN.CHANNEL.EVENT  
 SYSTEM.ADMIN.PERFM.EVENT  
 SYSTEM.ADMIN.CONFIG.EVENT  
 SYSTEM.ADMIN.COMMAND.EVENT  
 SYSTEM.ADMIN.LOGGER.EVENT  
 SYSTEM.ADMIN.PUBSUB.EVENT

**Enthält Nachrichten von:**

WS-Manager-Ereignisse  
 Kanalereignisse  
 Durchsatzereignisse  
 Konfigurationsereignisse  
 Befehlsereignisse  
 Ereignisse der Protokollfunktion  
 Ruft Ereignisse im Zusammenhang mit Publish/Subscribe ab. Nur bei Multicasting verwendet. Weitere Informationen finden Sie im Abschnitt Multicast-Anwendungsüberwachung.

Durch die Integration von Instrumentierungsereignissen in Ihre eigene Systemmanagementanwendung können Sie die Aktivitäten in vielen Warteschlangenmanagern, über viele verschiedene Knoten hinweg und für mehrere IBM MQ-Anwendungen überwachen. Insbesondere können Sie alle Knoten in Ihrem System von einem einzigen Knoten aus überwachen (für die Knoten, die IBM MQ-Ereignisse unterstützten); weitere Informationen finden Sie unter Abbildung 2 auf Seite 9.

Instrumentierungsereignisse können über einen vom Benutzer geschriebenen Berichtsmechanismus an eine Verwaltungsanwendung gemeldet werden, die die Ereignisse einem Bediener präsentieren kann.



*Abbildung 2. Warteschlangenmanager auf verschiedenen Plattformen überwachen, auf einem einzelnen Knoten*

Mit Instrumentierungsereignissen können auch Anwendungen aktiviert werden, die als Agenten für andere Netzwerke fungieren (z. B. Tivoli NetView für z/OS), um Berichte zu überwachen und die zugehörigen Benachrichtigungen zu erstellen.

*WS-Manager-Ereignisse*

WS-Manager-Ereignisse beziehen sich auf die Verwendung von Ressourcen in Warteschlangenmanagern. Beispielsweise wird ein WS-Manager-Ereignis generiert, wenn eine Anwendung versucht, eine Nachricht in eine Warteschlange zu stellen, die nicht vorhanden ist.

Die folgenden Beispiele zeigen Bedingungen, die ein Warteschlangenmanagerereignis verursachen können:

- Eine Anwendung setzt einen MQI-Aufruf ab, der fehlschlägt. Der Ursachencode aus dem Aufruf ist mit dem Ursachencode in der Ereignisnachricht identisch.

Eine ähnliche Bedingung kann während der internen Operation eines Warteschlangenmanagers auftreten, z. B. bei der Generierung einer Berichtsnachricht. Der Ursachencode in einer Ereignisnachricht kann mit einem MQI-Ursachencode übereinstimmen, auch wenn er keiner Anwendung zugeordnet ist. Nehmen Sie nicht an, dass das Ereignis zwangsläufig durch einen nicht erfolgreichen MQI-Aufruf von einer Anwendung verursacht wurde, da ein Ursachencode für eine Ereignisnachricht wie ein MQI-Ursachencode aussieht.

- Ein Befehl wird an einen Warteschlangenmanager ausgegeben und die Verarbeitung dieses Befehls bewirkt, dass ein Ereignis ausgeführt wird. Beispiel:
  - Ein WS-Manager wurde gestoppt oder gestartet.
  - Ein Befehl wird ausgegeben, wenn die zugeordnete Benutzer-ID für diesen Befehl nicht berechtigt ist.

IBM MQ reiht Nachrichten für Warteschlangenmanagerereignisse auf dem SYSTEM.ADMIN.QMGR.EVENT und unterstützt die folgenden Ereignistypen des Warteschlangenmanagers:

Windows

UNIX

### **Berechtigung (nur unter Windows und UNIX)**

Berechtigungsereignisse melden eine Berechtigung, z. B. eine Anwendung, die versucht, eine Warteschlange zu öffnen, für die sie nicht über die erforderliche Berechtigung verfügt, oder von einem Befehl, der von einer Benutzer-ID ausgegeben wird, die nicht über die erforderliche Berechtigung verfügt. Die Berechtigungsereignisnachricht kann die folgenden Ereignisdaten enthalten:

- [Nicht berechtigt \(Typ 1\)](#)
- [Nicht berechtigt \(Typ 2\)](#)
- [Nicht berechtigt \(Typ 3\)](#)
- [Nicht berechtigt \(Typ 4\)](#)
- [Nicht berechtigt \(Typ 5\)](#)
- [Nicht berechtigt \(Typ 6\)](#)

Alle Berechtigungsereignisse sind nur unter Windows und UNIX gültig.

### **Inhibit**

Blockierungsereignisse weisen darauf hin, dass eine MQPUT-oder MQGET-Operation für eine Warteschlange versucht wurde, in der die Warteschlange für die Ein-oder Abreißerungen oder für ein Thema gesperrt ist, in dem das Thema für die Veröffentlichung gesperrt ist. Die Sperrereignisnachricht kann die folgenden Ereignisdaten enthalten:

- [Abrufen unterdrückt](#)
- [Einreihen unterdrückt](#)

### **Lokal**

Wenn eine Anwendung oder der Warteschlangenmanager nicht in der Lage war, auf eine lokale Warteschlange oder ein anderes lokales Objekt zuzugreifen, z. B. weil das Objekt nicht definiert wurde, kann der Warteschlangenmanager eine lokale Ereignisnachricht generieren. Die lokale Ereignisnachricht kann die folgenden Ereignisdaten enthalten:

- [Fehler im Typ der Aliasbasiswarteschlange](#)
- [Unbekannte Aliasbasiswarteschlange](#)
- [Unbekannter Objektname](#)

### **Fern**

Wenn eine Anwendung oder der Warteschlangenmanager nicht auf eine ferne Warteschlange auf einem anderen WS-Manager zugreifen kann, z. B. die Übertragungswarteschlange nicht korrekt definiert ist, kann der Warteschlangenmanager eine ferne Ereignisnachricht generieren. Die ferne Ereignisnachricht kann die folgenden Ereignisdaten enthalten:

- [Fehler beim Typ der Standardübertragungswarteschlange](#)
- [Fehler bei Verwendung der Standardübertragungswarteschlange](#)
- [Fehler beim Warteschlangentyp](#)
- [Fehler beim Namen der fernen Warteschlange](#)
- [Fehler beim Typ der Übertragungswarteschlange](#)
- [Fehler bei Verwendung der Übertragungswarteschlange](#)
- [Unbekannte Standardübertragungswarteschlange](#)
- [Unbekannter ferner Warteschlangenmanager](#)
- [Unbekannte Übertragungswarteschlange](#)

### Start und Stopp

Start- und Stoppereignisse weisen darauf hin, dass ein Warteschlangenmanager gestartet wurde oder angefordert wurde, um zu stoppen oder stillgelegt zu werden.

 z/OS unterstützt nur Startereignisse.

Stoppereignisse werden nicht aufgezeichnet, es sei denn, die Standardnachrichtenpersistenz der Warteschlange SYSTEM.ADMIN.QMGR.EVENT ist als persistent definiert. Die Ereignisnachricht 'start' und 'stop' kann die folgenden Ereignisdaten enthalten:

- [Warteschlangenmanager aktiv](#)
- [Warteschlangenmanager nicht aktiv](#)

Für jeden Ereignistyp in dieser Liste können Sie ein WS-Manager-Attribut festlegen, um den Ereignistyp zu aktivieren oder zu inaktivieren.

### Kanal- und Brückenereignisse

Kanäle melden diese Ereignisse als Ergebnis von Bedingungen, die während ihrer Operation erkannt wurden. Beispiel: Wenn eine Kanalinstanz gestoppt wird.

Kanalereignisse werden unter den folgenden Umständen generiert:

- Wenn ein Befehl gestartet oder gestoppt wird, wird ein Kanal gestoppt.
- Wenn eine Kanalinstanz gestartet oder gestoppt wird.
- Wenn ein Kanal beim Abrufen einer Nachricht eine Warnung zum Konvertierungsfehler empfängt.
- Wenn versucht wird, einen Kanal automatisch zu erstellen, wird das Ereignis generiert, unabhängig davon, ob der Versuch erfolgreich ist oder fehlschlägt.

**Anmerkung:** Clientverbindungen verursachen keinen Kanal Gestartet oder Gestoppte Ereignisse.

Wenn ein Befehl zum Starten eines Kanals verwendet wird, wird ein Ereignis generiert. Ein weiteres Ereignis wird generiert, wenn die Kanalinstanz gestartet wird. Beim Starten eines Kanals durch einen Listener, den Befehl `runmqchl` oder eine Auslösenachricht des Warteschlangenmanagers wird jedoch kein Ereignis generiert. In diesen Fällen wird ein Ereignis nur generiert, wenn die Kanalinstanz gestartet wird.

Bei einem erfolgreichen Start- oder Stoppkanalbefehl werden mindestens zwei Ereignisse generiert. Diese Ereignisse werden für beide Warteschlangenmanager generiert, die durch den Kanal verbunden sind (sofern sie Unterstützungsereignisse unterstützen).

Wenn ein Kanalereignis in eine Ereigniswarteschlange gestellt wird, bewirkt eine Fehlerbedingung, dass der WS-Manager ein Ereignis erstellt.

Die Ereignisnachrichten für Kanal- und Brückenereignisse werden in die Warteschlange SYSTEM.ADMIN.CHANNEL.EVENT gestellt.

Die Kanalereignisnachrichten können die folgenden Ereignisdaten enthalten:

- [Kanal aktiviert](#)
- [Fehler bei der automatischen Kanaldefinition](#)

- [Automatische Kanaldefinition OK](#)
- [Kanalkonvertierungsfehler](#)
- [Kanal nicht aktiviert](#)
- [Kanal gestartet](#)
- [Kanal gestoppt](#)
- [Kanal gestoppt nach Benutzer](#)
- [Kanal blockiert](#)

## IMS-Bridge-Ereignisse (nur z/OS)



Diese Ereignisse werden gemeldet, wenn eine IMS-Bridge gestartet oder gestoppt wird.

Die Ereignisnachrichten der IMS-Bridge können die folgenden Ereignisdaten enthalten:

- [Brücke gestartet](#)
- [Brücke gestoppt](#)

## SSL-Ereignisse

Das einzige TLS-Ereignis ist das Ereignis "Channel SSL Error". Dieses Ereignis wird gemeldet, wenn ein Kanal, der TLS verwendet, keine TLS-Verbindung herstellen kann.

Die SSL-Ereignisnachrichten können die folgenden Ereignisdaten enthalten:

- [Kanal-SSL-Fehler](#)
- [Channel SSL Warning](#)

### *Durchsatzereignisse*

Bei Leistungsereignissen handelt es sich um Benachrichtigungen, bei denen eine Ressource eine Schwellenwertbedingung erreicht hat. Es wurde beispielsweise eine Warteschlangenlänge erreicht.

Leistungsereignisse beziehen sich auf Bedingungen, die sich auf die Leistung von Anwendungen auswirken können, die eine angegebene Warteschlange verwenden. Sie werden nicht für die Ereigniswarteschlangen selbst generiert.

Der Ereignistyp wird im Feld 'Befehls-ID' in den Nachrichtendaten zurückgegeben.

Wenn ein Warteschlangenmanager versucht, eine Ereignis- oder Leistungsereignisnachricht eines Warteschlangenmanagers in eine Ereigniswarteschlange zu stellen und ein Fehler erkannt wird, der normalerweise ein Ereignis erstellen würde, wird ein anderes Ereignis nicht erstellt und es wird keine Aktion ausgeführt.

MQGET- und MQPUT-Aufrufe innerhalb einer Arbeitseinheit können Leistungsereignisse generieren, unabhängig davon, ob die UOWs festgeschrieben oder zurückgesetzt werden.

Die Ereignisnachrichten für Leistungsereignisse werden in die Warteschlange SYSTEM.ADMIN.PERFM.EVENT gestellt.

Es gibt zwei Typen von Leistungsereignis:

### **Ereignisse der Warteschlangenlänge**

Ereignisse der Warteschlangenlänge beziehen sich auf die Anzahl der Nachrichten in einer Warteschlange. Das heißt, wie voll oder leer die Warteschlange ist. Diese Ereignisse werden für gemeinsam genutzte Warteschlangen unterstützt. Die Ereignisnachrichten der Warteschlangenlänge können die folgenden Ereignisdaten enthalten:

- [Warteschlangenlänge hoch](#)
- [Warteschlangenlänge niedrig](#)
- [Warteschlange voll](#)

## Ereignis 'Intervall für Warteschlangenwartung'

Ereignisse des Warteschlangenserviceintervalls beziehen sich darauf, ob Nachrichten innerhalb eines vom Benutzer angegebenen Zeitintervalls verarbeitet werden. Diese Ereignisse werden für gemeinsam genutzte Warteschlangen nicht unterstützt.

 IBM MQ for z/OS unterstützt Warteschlangenlängenergebnisse für QSGDISP-Warteschlangen (SHARED), aber keine Serviceintervallereignisse. Warteschlangenmanager- und Kanalergebnisse bleiben von gemeinsam genutzten Warteschlangen nicht betroffen. Die Ereignisnachrichten des Warteschlangenservice können die folgenden Ereignisdaten enthalten:

- [Hohe Warteschlangenserviceintervall](#)
- [Warteschlangenserviceintervall OK](#)

### Konfigurationsergebnisse

Konfigurationsergebnisse werden generiert, wenn ein Konfigurationsergebnis explizit angefordert wird, oder automatisch, wenn ein Objekt erstellt, geändert oder gelöscht wird.

Eine Konfigurationsergebnisnachricht enthält Informationen zu den Attributen eines Objekts. Es wird beispielsweise eine Konfigurationsergebnisnachricht generiert, wenn ein Namenslistenobjekt erstellt wird und Informationen zu den Attributen des Namenslistenobjekts enthält.

Die Ereignisnachrichten für Konfigurationsergebnisse werden in die Warteschlange SYSTEM.ADMIN.CONFIG.EVENT gestellt.

Konfigurationsergebnisse funktionieren auf folgende Weise:

- Ein Prüfprotokoll über Änderungen an der Warteschlangenmanagerkonfiguration wird über Ereignisnachrichten gespeichert, die in die Warteschlange SYSTEM.ADMIN.CONFIG.EVENT geschrieben werden. Sie können diese Ereignisse mit Hilfe des Parameters **CONFIGEV** im Befehl [ALTER QMGR](#) aktivieren.
- Diese Ereignisse werden generiert, wenn ein Befehl DEFINE, ALTER oder DELETE auf ein Objekt oder ein MQSET-Aufruf angewendet wird.
- Sie können mit dem Befehl [REFRESH QMGR TYPE \(CONFIGEV\)](#) ein Basislinienbild der aktuellen WS-Manager-Konfiguration erstellen, wodurch für jedes Objekt im Warteschlangenmanager eine Ereignisnachricht erstellt wird. Beachten Sie, dass Sie, wenn Sie viele Objekte haben, die Task in kleinere Gruppen von Objekten mit den Qualifikationsmerkmalen NAME und OBJECT für den Befehl aufteilen können.
- In der Ereignisnachricht wird einer von vier möglichen Ursachen aufgezeichnet:
  - MQRC\_CONFIG\_CHANGE\_OBJECT
  - MQRC\_CONFIG\_CREATE\_OBJECT
  - MQRC\_CONFIG\_DELETE\_OBJECT
  - MQRC\_CONFIG\_REFRESH\_OBJECT

MQRC\_CONFIG\_CHANGE\_OBJECT, MQRC\_CONFIG\_CREATE\_OBJECT oder MQRC\_CONFIG\_DELETE\_OBJECT treten für die jeweiligen MQSC- oder PCF-Befehle auf, die Sie bei einem Objekt ausgeben können.

MQRC\_CONFIG\_REFRESH\_OBJECT tritt für diese Ereignisnachrichten auf, die beim Erstellen des Grundlinienbilds geschrieben werden.

Es gibt vier Typen von Konfigurationsergebnisereignis:

### Objektergebnisse erstellen

Objekt-Ereignisse erstellen werden generiert, wenn ein Objekt erstellt wird. Die Ereignisnachricht enthält die folgenden Ereignisdaten: [Objekt erstellen](#) .

### Objektergebnisse ändern

Änderungsobjektergebnisse werden generiert, wenn ein Objekt geändert wird. Die Ereignisnachricht enthält die folgenden Ereignisdaten: [Change-Objekt](#) .

## Objektereignisse löschen

Objektereignisse löschen werden generiert, wenn ein Objekt gelöscht wird. Die Ereignisnachricht enthält die folgenden Ereignisdaten: Objekt löschen .

## Objektereignisse aktualisieren

Aktualisierungsobjektereignisse werden durch eine explizite Anforderung zum Aktualisieren generiert. Die Ereignisnachricht enthält die folgenden Ereignisdaten: Aktualisierungsobjekt .

### *Befehlereignisse*

Befehlereignisse werden gemeldet, wenn ein MQSC- oder PCF-Befehl erfolgreich ausgeführt wird.

Eine Befehlereignisnachricht enthält Informationen über den Ursprung, den Kontext und den Inhalt eines Befehls. Beispiel: Eine Befehlereignisnachricht wird mit solchen Informationen generiert, wenn der MQSC-Befehl ALTER QLOCAL erfolgreich ausgeführt wird.

Die Ereignisnachrichten für Befehlereignisse werden in die Warteschlange SYSTEM.ADMIN.COMMAND.EVENT gestellt.

Befehlereignisse enthalten die folgenden Ereignisdaten: Befehl .

### **Multi** *Ereignisse der Protokollfunktion*

Protokollierungsereignisse werden dokumentiert, wenn ein Warteschlangenmanager, der die lineare Protokollierung verwendet, mit dem Schreiben von Protokolldatensätzen in einen neuen Protokollspeicherbereich **IBM i** oder unter IBM i in einen neuen Journalempfänger beginnt. **z/OS** Protokollierungsereignisse sind mit IBM MQ for z/OS nicht verfügbar.

Eine Protokollfunktionsereignisnachricht enthält Informationen, die die Protokollspeicherbereiche angeben, die vom Warteschlangenmanager für den Neustart des Warteschlangenmanagers oder für die Datenträgerwiederherstellung benötigt werden.

Die Ereignisnachrichten für Protokollfunktionsereignisse werden in die Warteschlange SYSTEM.ADMIN.LOGGER.EVENT gestellt.

Die Ereignisnachricht der Protokollfunktion enthält die folgenden Ereignisdaten: Logger .

### *Zusammenfassung der Ereignisnachrichtendaten*

Verwenden Sie diese Zusammenfassung, um Informationen zu den Ereignisdaten zu erhalten, die jeder Typ von Ereignisnachricht enthalten kann.

<b>Ereignistyp</b>	<b>Sehen Sie sich diese Themen</b>
Berechtigungsereignisse	<a href="#">Nicht berechtigt (Typ 1)</a>
	<a href="#">Nicht berechtigt (Typ 2)</a>
	<a href="#">Nicht berechtigt (Typ 3)</a>
	<a href="#">Nicht berechtigt (Typ 4)</a>
	<a href="#">Nicht berechtigt (Typ 5)</a>
	<a href="#">Nicht berechtigt (Typ 6)</a>

<b>Ereignistyp</b>	<b>Sehen Sie sich diese Themen</b>
Kanalereignisse	<a href="#">Kanal aktiviert</a>
	<a href="#">Fehler bei der automatischen Kanaldefinition</a>
	<a href="#">Automatische Kanaldefinition OK</a>
	<a href="#">Kanal blockiert</a>
	<a href="#">Kanalkonvertierungsfehler</a>
	<a href="#">Kanal nicht aktiviert</a>
	<a href="#">Kanal gestartet</a>
	<a href="#">Kanal gestoppt</a>
	<a href="#">Kanal gestoppt nach Benutzer</a>
Befehlsereignisse	<a href="#">Befehl</a>
Konfigurationsereignisse	<a href="#">Objekt erstellen</a>
	<a href="#">Objekt ändern</a>
	<a href="#">Objekt löschen</a>
	<a href="#">Objekt aktualisieren</a>
IMS-Brückenereignisse	<a href="#">Brücke gestartet</a>
	<a href="#">Brücke gestoppt</a>
Sperrereignisse	<a href="#">Abrufen unterdrückt</a>
	<a href="#">Einreihen unterdrückt</a>
Lokale Ereignisse	<a href="#">Fehler im Typ der Aliasbasiswarteschlange</a>
	<a href="#">Unbekannte Aliasbasiswarteschlange</a>
	<a href="#">Unbekannter Objektname</a>
Ereignisse der Protokollfunktion	<a href="#">Logger</a>
Durchsatzereignisse	<a href="#">Warteschlangenlänge hoch</a>
	<a href="#">Warteschlangenlänge niedrig</a>
	<a href="#">Warteschlange voll</a>
	<a href="#">Hohe Warteschlangenserviceintervall</a>
	<a href="#">Warteschlangenserviceintervall OK</a>

Ereignistyp	Sehen Sie sich diese Themen
Ferne Ereignisse	<a href="#">Fehler beim Typ der Standardübertragungswarteschlange</a>
	<a href="#">Fehler bei Verwendung der Standardübertragungswarteschlange</a>
	<a href="#">Fehler beim Warteschlangentyp</a>
	<a href="#">Fehler beim Namen der fernen Warteschlange</a>
	<a href="#">Fehler beim Typ der Übertragungswarteschlange</a>
	<a href="#">Fehler bei Verwendung der Übertragungswarteschlange</a>
	<a href="#">Unbekannte Standardübertragungswarteschlange</a>
	<a href="#">Unbekannter ferner Warteschlangenmanager</a>
	<a href="#">Unbekannte Übertragungswarteschlange</a>
SSL-Ereignisse	<a href="#">Kanal-SSL-Fehler</a>
Start- und Stoppereignisse	<a href="#">Warteschlangenmanager aktiv</a>
	<a href="#">Warteschlangenmanager nicht aktiv</a>

### **Ereignisse steuern**

Sie aktivieren und inaktivieren Ereignisse, indem Sie die entsprechenden Werte für Warteschlangenmanager, Warteschlangenattribute oder beides angeben, je nach Ereignistyp.

Sie müssen jedes Instrumentierungsereignis aktivieren, das generiert werden soll. Die Bedingungen, die das Ereignis "Warteschlange voll" verursachen, sind z. B.:

- Ereignisse in der Warteschlange 'Vollständig' werden für eine angegebene Warteschlange aktiviert und
- Eine Anwendung gibt eine MQPUT-Anforderung aus, um eine Nachricht in diese Warteschlange zu stellen, aber die Anforderung schlägt fehl, da die Warteschlange voll ist.

Sie können Ereignisse mithilfe einer der folgenden Methoden aktivieren und inaktivieren:

- IBM MQ-Scriptbefehle (MQSC).
- Die entsprechenden IBM MQ-PCF-Befehle
-  Die Operationen und Steuerkonsolen für Warteschlangenmanager unter z/OS.
- IBM MQ Explorer.

**Anmerkung:** Sie können Attribute, die sich auf Ereignisse beziehen, nur nach Befehl für Warteschlangen und Warteschlangenmanager festlegen. Der MQI-Aufruf MQSET unterstützt keine Attribute, die sich auf Ereignisse beziehen.

### **Zugehörige Konzepte**

„Instrumentierungsereignisse“ auf Seite 6

Ein Instrumentierungsereignis ist eine logische Kombination von Bedingungen, die ein Warteschlangenmanager oder eine Kanalinstanz feststellt und in eine Ereigniswarteschlange eine spezielle Nachricht, die so genannte *Ereignisnachricht*, einreicht.

[Einführung in die Betriebs- und Steuerkonsolen](#)

### **Zugehörige Tasks**

[Verwaltungstasks automatisieren](#)

[Programmierbare Befehlsformate verwenden](#)

### **Zugehörige Verweise**

„Ereignistypen“ auf Seite 8

Verwenden Sie diese Seite, um die Typen von Instrumentierungsereignis anzuzeigen, die ein Warteschlangenmanager oder eine Kanalinstanz melden kann.

## Die MQSC-Befehle

### Warteschlangenmanagerereignisse steuern

Warteschlangenmanager-Ereignisse werden über WS-Manager-Attribute gesteuert. Um WS-Manager-Ereignisse zu aktivieren, setzen Sie das entsprechende Attribut des Warteschlangenmanagers auf ENABLED . Um Warteschlangenmanagerereignisse zu inaktivieren, setzen Sie das entsprechende WS-Manager-Attribut auf DISABLED .

Verwenden Sie zum Aktivieren oder Inaktivieren von WS-Manager-Ereignissen den MQSC-Befehl **ALTER QMGR** und geben Sie dabei das entsprechende WS-Manager-Attribut an. In [Tabelle 1 auf Seite 17](#) ist zusammengefasst, wie WS-Manager-Ereignisse aktiviert werden. Wenn Sie ein WS-Manager-Ereignis inaktivieren möchten, setzen Sie den entsprechenden Parameter auf DISABLED .

<b>Ereignis</b>	<b>ALTER QMGR, Parameter</b>
Berechtigung	AUTHOREV (ENABLED)
Inhibit	INHIBTEV (ENABLED)
Lokal	LOCALEV (ENABLED)
Fern	REMOTEEV (ENABLED)
Start und Stopp	STRSTPEV (ENABLED)

### Kanal- und Brückenereignisse steuern

Sie können Kanalereignisse mithilfe von Warteschlangenmanagerattributen steuern. Um Kanalereignisse zu aktivieren, setzen Sie das entsprechende WS-Manager-Attribut auf ENABLED . Um Kanalereignisse zu inaktivieren, setzen Sie das entsprechende Attribut des Warteschlangenmanagers auf DISABLED .

Verwenden Sie zum Aktivieren oder Inaktivieren von Kanalereignissen den MQSC-Befehl **ALTER QMGR** und geben Sie das entsprechende Warteschlangenmanagerattribut an. In [Tabelle 2 auf Seite 17](#) ist zusammengefasst, wie Sie Kanal- und Brückenereignisse aktivieren. Wenn Sie ein WS-Manager-Ereignis inaktivieren möchten, setzen Sie den entsprechenden Parameter auf DISABLED .

**Einschränkung:**  Ereignisse zur automatischen Definition von Kanälen sind unter IBM MQ for z/OS nicht verfügbar.

<b>Ereignis</b>	<b>ALTER QMGR, Parameter</b>
Kanal	CHLEV (ENABLED)
Nur bei Kanalfehlern	CHLEV (EXCEPTION)
IMS Bridge	BRIDGEEV (ENABLED)
SSL	SSLEV (ENABLED)
Automatische Kanaldefinition	CHADEV (ENABLED)

Wenn CHLEV auf eine Ausnahmebedingung gesetzt wird, werden die folgenden Rückkehrcodes und die entsprechenden Ursachenqualifikationsmerkmale generiert:

- MQRC\_CHANNEL\_ACTIVATED
- MQRC\_CHANNEL\_CONV\_ERROR
- MQRC\_CHANNEL\_NOT\_ACTIVATED
- MQRC\_CHANNEL\_STOPPED
  - mit den folgenden 'ReasonQualifiers':
    - MQRQ\_CHANNEL\_STOPPED\_ERROR
    - MQRQ\_CHANNEL\_STOPPED\_RETRY
    - MQRQ\_CHANNEL\_STOPPED\_DISABLED

- MQRC\_CHANNEL\_STOPPED\_BY\_USER
- MQRC\_CHANNEL\_BLOCKED
  - mit den folgenden 'ReasonQualifiers':
    - MQRQ\_CHANNEL\_BLOCKED\_NOACCESS
    - MQRQ\_CHANNEL\_BLOCKED\_USERID
    - MQRQ\_CHANNEL\_BLOCKED\_ADDRESS

#### *Leistungsereignisse steuern*

Leistungsereignisse werden mithilfe des Warteschlangenmanagerattributs PERFMEV gesteuert. Wenn Sie Leistungsereignisse aktivieren möchten, setzen Sie PERFMEV auf ENABLED . Um Leistungsereignisse zu inaktivieren, setzen Sie das WS-Manager-Attribut PERFMEV auf DISABLED .

Verwenden Sie den folgenden MQSC-Befehl, um das WS-Manager-Attribut PERFMEV auf ENABLED zu setzen:

```
ALTER QMGR PERFMEV (ENABLED)
```

Wenn Sie bestimmte Leistungsereignisse aktivieren möchten, legen Sie das entsprechende Warteschlangenattribut fest. Geben Sie außerdem die Bedingungen an, die das Ereignis verursachen.

#### **Ereignisse der Warteschlangenlänge**

Standardmäßig sind alle Warteschlangenlänge-Ereignisse inaktiviert. Gehen Sie wie folgt vor, um eine Warteschlange für einen der Warteschlangenlänge zu konfigurieren:

1. Aktivieren Sie Leistungsereignisse auf dem Warteschlangenmanager.
2. Aktivieren Sie das Ereignis in der erforderlichen Warteschlange.
3. Legen Sie die Grenzwerte (falls erforderlich) auf die entsprechenden Ebenen fest, ausgedrückt als Prozentsatz der maximalen Warteschlangenlänge.

#### **Ereignis 'Intervall für Warteschlangenwartung'**

Gehen Sie wie folgt vor, um eine Warteschlange für Warteschlangenserviceintervallereignisse zu konfigurieren

1. Aktivieren Sie Leistungsereignisse auf dem Warteschlangenmanager.
2. Setzen Sie das Steuerattribut für ein Warteschlangenserviceintervall Hoch-oder OK-Ereignis in der Warteschlange nach Bedarf.
3. Geben Sie die Serviceintervallzeit an, indem Sie das Attribut Q SVCINT für die Warteschlange auf die entsprechende Zeitdauer setzen.

**Anmerkung:** Wenn diese Option aktiviert ist, kann zu einem beliebigen Zeitpunkt ein Warteschlangenserviceintervall generiert werden, das nicht unbedingt warten muss, bis ein MQI-Aufruf für die Warteschlange ausgegeben wird. Wenn jedoch ein MQI-Aufruf in einer Warteschlange zum Angeben oder Entfernen einer Nachricht verwendet wird, wird zu diesem Zeitpunkt ein beliebtes anwendbares Leistungsereignis generiert. Das Ereignis wird nicht generiert, wenn die abgelaufene Zeit gleich der Serviceintervallzeit wird.

#### *Steuerung von Konfigurations-, Befehls- und Protokollfunktionseignissen*

Mit den WS-Managerattributen CONFIGEV, CMDEV und LOGGEREV können Sie die Konfigurations-, Befehls- und Protokollfunktionseignisse steuern. Um diese Ereignisse zu aktivieren, setzen Sie das entsprechende WS-Manager-Attribut auf ENABLED . Wenn Sie diese Ereignisse inaktivieren möchten, setzen Sie das entsprechende Attribut des Warteschlangenmanagers auf DISABLED .

## Konfigurationsereignisse

Wenn Sie Konfigurationsereignisse aktivieren möchten, setzen Sie CONFIGEV auf ENABLED . Wenn Sie Konfigurationsereignisse inaktivieren möchten, setzen Sie CONFIGEV auf DISABLED . Sie können beispielsweise Konfigurationsereignisse mit dem folgenden MQSC-Befehl aktivieren:

```
ALTER QMGR CONFIGEV (ENABLED)
```

## Befehlsereignisse

Um Befehlsereignisse zu aktivieren, setzen Sie CMDEV auf ENABLED . Um Befehlsereignisse für Befehle außer DISPLAY MQSC-Befehle und Inquire PCF-Befehle zu aktivieren, setzen Sie die CMDEV auf NODISPLAY . Um Befehlsereignisse zu inaktivieren, setzen Sie CMDEV auf DISABLED . Sie können z. B. Befehlsereignisse mit dem folgenden MQSC-Befehl aktivieren:

```
ALTER QMGR CMDEV (ENABLED)
```

## Ereignisse der Protokollfunktion

Um Protokollfunktionsereignisse zu aktivieren, setzen Sie LOGGEREV auf ENABLED . Um Protokollfunktionsereignisse zu inaktivieren, setzen Sie LOGGEREV auf DISABLED . Sie können z. B. Protokollfunktionsereignisse mit dem folgenden MQSC-Befehl aktivieren:

```
ALTER QMGR LOGGEREV(ENABLED)
```

## Ereigniswarteschlangen

Wenn ein Ereignis eintritt, reißt der Warteschlangenmanager eine Ereignisnachricht in die definierte Ereigniswarteschlange ein. Die Ereignisnachricht enthält Informationen zu dem Ereignis.

Sie können Ereigniswarteschlangen wie folgt definieren:

- Lokale Warteschlangen
- Aliaswarteschlangen
- Lokale Definitionen von fernen Warteschlangen oder als
- Ferne Clusterwarteschlangen

Wenn Sie alle Ereigniswarteschlangen als lokale Definitionen derselben fernen Warteschlange in einem Warteschlangenmanager definieren, können Sie Ihre Überwachungsaktivitäten zentralisieren.

Sie dürfen Ereigniswarteschlangen nicht als Übertragungswarteschlangen definieren, da Ereignisnachrichten Formate aufweisen, die mit dem Nachrichtenformat, das für Übertragungswarteschlangen erforderlich ist, nicht kompatibel sind.

Gemeinsam genutzte Ereigniswarteschlangen sind lokale Warteschlangen, die mit dem Wert QSGDISP (SHARED) definiert werden.

Weitere Informationen zum Definieren von gemeinsam genutzten Warteschlangen unter z/OS, finden Sie im Abschnitt [Anwendungsprogrammierung mit gemeinsam genutzten Warteschlangen](#).

## Wenn eine Ereigniswarteschlange nicht verfügbar ist

Wenn ein Ereignis auftritt, wenn die Ereigniswarteschlange nicht verfügbar ist, geht die Ereignisnachricht verloren. Wenn Sie beispielsweise keine Ereigniswarteschlange für eine Kategorie von Ereignissen definieren, gehen alle Ereignisnachrichten für diese Kategorie verloren. Die Ereignisnachrichten werden z. B. nicht in der Warteschlange für nicht zustellbare Nachrichten (dead-letter) gespeichert.

Sie können die Ereigniswarteschlange jedoch als ferne Warteschlange definieren. Wenn dann ein Problem auf dem fernen System auftritt, das Nachrichten in die aufgelöste Warteschlange einreihen soll, wird die Ereignisnachricht in die Warteschlange für den dead-letter des fernen Systems gestellt.

Eine Ereigniswarteschlange ist möglicherweise aus vielen verschiedenen Gründen nicht verfügbar. Hierzu gehören:

- Die Warteschlange wurde nicht definiert.
- Die Warteschlange wurde gelöscht.
- Die Warteschlange ist voll.
- Die Warteschlange wurde gesperrt.

Das Fehlen einer Ereigniswarteschlange verhindert nicht, dass das Ereignis eintritt. Nach einem Leistungsereignis ändert der WS-Manager beispielsweise die Warteschlangenattribute und setzt die Warteschlangenstatistik zurück. Diese Änderung geschieht, wenn die Ereignisnachricht in die Leistungsereigniswarteschlange gestellt wird oder nicht. Dies gilt auch für Konfigurations- und Befehlsereignisse.

## Ausgelöste Ereigniswarteschlangen verwenden

Sie können die Ereigniswarteschlangen mit Triggern so konfigurieren, dass bei der Generierung eines Ereignisses die Ereignisnachricht, die in die Ereigniswarteschlange gestellt wird, eine vom Benutzer geschriebene Überwachungsanwendung startet. Diese Anwendung kann die Ereignisnachrichten verarbeiten und entsprechende Maßnahmen ergreifen. Für bestimmte Ereignisse kann es beispielsweise erforderlich sein, dass ein Bediener informiert wird. Andere Ereignisse können eine Anwendung starten, die einige Verwaltungstasks automatisch ausführt.

Ereigniswarteschlangen können Auslöseraktionen zugeordnet sein und Auslösenachrichten erstellen. Wenn diese Auslösenachrichten wiederum Bedingungen verursachen, die normalerweise ein Ereignis generieren würden, wird kein Ereignis generiert. Wenn in dieser Instanz kein Ereignis generiert wird, wird sichergestellt, dass keine Schleife auftritt.

### Zugehörige Konzepte

„Ereignisse steuern“ auf Seite 16

Sie aktivieren und inaktivieren Ereignisse, indem Sie die entsprechenden Werte für Warteschlangenmanager, Warteschlangenattribute oder beides angeben, je nach Ereignistyp.

„Format von Ereignisnachrichten“ auf Seite 20

Ereignisnachrichten enthalten Informationen zu einem Ereignis und dessen Ursache. Wie andere IBM MQ-Nachrichten besteht auch eine Ereignisnachricht aus zwei Teilen: einem Nachrichtendeskriptor und den Nachrichtendaten.

Anwendungsprogrammierung mit gemeinsam genutzten Warteschlangen

Bedingungen für ein Auslöserereignis

### Zugehörige Verweise

QSGDisp (MQLONG)

## Format von Ereignisnachrichten

Ereignisnachrichten enthalten Informationen zu einem Ereignis und dessen Ursache. Wie andere IBM MQ-Nachrichten besteht auch eine Ereignisnachricht aus zwei Teilen: einem Nachrichtendeskriptor und den Nachrichtendaten.

- Der Nachrichtendeskriptor basiert auf der MQMD-Struktur.
- Die Nachrichtendaten bestehen aus einem *Ereignisheader* und den *Ereignisdaten*. Der Ereignisheader enthält den Ursachencode, der den Ereignistyp angibt. Wenn die Ereignisnachricht und alle nachfolgenden Aktionen beendet werden, wirkt sich dies nicht auf den Ursachencode aus, der von dem MQI-Aufruf zurückgegeben wurde, der das Ereignis ausgelöst hat. Die Ereignisdaten stellen weitere Informationen zu dem Ereignis bereit.

In der Regel verarbeiten Sie Ereignisnachrichten mit einer Systemmanagementanwendung, die auf die Anforderungen des Unternehmens zugeschnitten ist, auf dem sie ausgeführt wird.

Wenn die Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange die Bedingungen für die Generierung einer Ereignisnachricht ermitteln, können mehrere Warteschlangenmanager eine Ereignisnachricht für die gemeinsam genutzte Warteschlange generieren, was zu mehreren Ereignisnachrichten führt. Um sicherzustellen, dass ein System mehrere Ereignisnachrichten von unterschiedlichen Warteschlangenmanagern korrelieren kann, wird in diesen Ereignisnachrichten eine eindeutige Korrelations-ID (*CorrelId*) im Nachrichtendeskriptor (MQMD) festgelegt.

## Zugehörige Verweise

[„MQMD des Aktivitätsberichts \(Nachrichtendeskriptor\)“ auf Seite 104](#)

Verwenden Sie diese Seite, um die Werte anzuzeigen, die in der MQMD-Struktur für einen Aktivitätsbericht enthalten sind.

[„Aktivitätenbericht MQEPH \(eingebetteter PCF-Header\)“ auf Seite 108](#)

Verwenden Sie diese Seite, um die Werte anzuzeigen, die in der MQEPH-Struktur für einen Aktivitätsbericht enthalten sind.

[„Aktivitätenbericht MQCFH \(PCF-Header\)“ auf Seite 110](#)

Verwenden Sie diese Seite, um die PCF-Werte anzuzeigen, die in der MQCFH-Struktur für einen Aktivitätsbericht enthalten sind.

[Ereignisnachrichtenreferenz](#)

[Ereignisnachrichtenformat](#)

[Ereignisnachricht MQMD \(Nachrichtendeskriptor\)](#)

[Ereignisnachricht MQCFH \(PCF-Header\)](#)

[Ereignisnachrichtenbeschreibungen](#)

## Durchsatzereignisse

Leistungsereignisse beziehen sich auf Bedingungen, die sich auf die Leistung von Anwendungen auswirken können, die eine angegebene Warteschlange verwenden. Der Umfang der Leistungsereignisse ist die Warteschlange. **MQPUT** -Aufrufe und **MQGET** -Aufrufe in einer Warteschlange haben keine Auswirkungen auf die Generierung von Leistungsereignissen in einer anderen Warteschlange.

Es können zu jedem geeigneten Zeitpunkt Leistungsereignisnachrichten generiert werden, die nicht unbedingt warten müssen, bis ein MQI-Aufruf für die Warteschlange ausgegeben wird. Wenn Sie jedoch einen MQI-Aufruf in einer Warteschlange verwenden, um eine Nachricht zu stellen oder zu entfernen, werden zu diesem Zeitpunkt alle entsprechenden Leistungsereignisse generiert.

Jede generierte Leistungsereignisnachricht wird in die Warteschlange SYSTEM.ADMIN.PERFM.EVENT gestellt.

Die Ereignisdaten enthalten einen Ursachencode, der die Ursache des Ereignisses, eine Gruppe von Leistungsereignisstatistikdaten und andere Daten angibt. In der folgenden Liste werden die Typen von Ereignisdaten beschrieben, die in Performance-Ereignisnachrichten zurückgegeben werden können:

- [Warteschlangenlänge hoch](#)
- [Warteschlangenlänge niedrig](#)
- [Warteschlange voll](#)
- [Hohe Warteschlangenserviceintervall](#)
- [Warteschlangenserviceintervall OK](#)

In den Beispielen, die die Verwendung von Leistungsereignissen darstellen, wird davon ausgegangen, dass Sie Warteschlangenattribute mithilfe der zugehörigen IBM MQ-Befehle (MQSC) festlegen. Unter z/OS können Sie Warteschlangenattribute auch mithilfe der Operationen und Steueranzeigen für Warteschlangenmanager festlegen.

## Zugehörige Verweise

[„Ereignistypen“ auf Seite 8](#)

Verwenden Sie diese Seite, um die Typen von Instrumentierungsereignis anzuzeigen, die ein Warteschlangenmanager oder eine Kanalinstanz melden kann.

## Performance-Ereignisstatistik

Die Leistungsereignisdaten in der Ereignisnachricht enthalten Statistikdaten zu dem Ereignis. Verwenden Sie die Statistikdaten, um das Verhalten einer angegebenen Warteschlange zu analysieren.

Die Ereignisdaten in der Ereignisnachricht enthalten Informationen zu dem Ereignis für Systemmanagementprogramme. Bei allen Leistungsereignissen enthalten die Ereignisdaten die Namen des Warteschlangenmanagers und der Warteschlange, die dem Ereignis zugeordnet sind. Die Ereignisdaten enthalten

außerdem Statistikdaten zu dem Ereignis. In [Tabelle 3](#) auf Seite 22 sind die Ereignisstatistikdaten zusammengefasst, die Sie zum Analysieren des Verhaltens einer Warteschlange verwenden können. Alle Statistikdaten beziehen sich auf das, was seit dem letzten Zurücksetzen der Statistik passiert ist.

<i>Tabelle 3. Performance-Ereignisstatistik</i>	
<b>Parameter</b>	<b>Beschreibung</b>
Zurücksetzungs-zeit (TimeS)	Die abgelaufene Zeit seit dem letzten Zurücksetzen der Statistik.
HighQDepth	Die maximale Anzahl der Nachrichten in der Warteschlange seit dem letzten Zurücksetzen der Statistik.
MsgEnqCount	Die Anzahl der in die Warteschlange eingereihten Nachrichten (die Anzahl der MQPUT-Aufrufe an die Warteschlange), da die Statistikdaten zum letzten Mal zurückgesetzt wurden.
MsgDeqCount	Die Anzahl der Nachrichten, die in die Warteschlange gestellt wurden (die Anzahl der MQGET-Aufrufe an die Warteschlange), da die Statistik zuletzt zurückgesetzt wurde.

Die Leistungsereignisstatistikdaten werden zurückgesetzt, wenn eine der folgenden Änderungen eintritt:

- Es tritt ein Leistungsereignis auf (Statistikdaten werden auf allen aktiven Warteschlangenmanagern zurückgesetzt).
- Ein WS-Manager wird gestoppt und erneut gestartet.
- Der PCF-Befehl "Warteschlangenstatistik zurücksetzen" wird von einem Anwendungsprogramm ausgegeben.
-  Der Befehl RESET QSTATS wird nur unter z/OS auf der Konsole ausgegeben.

### **Zugehörige Konzepte**

„Durchsatzereignisse“ auf Seite 21

Leistungsereignisse beziehen sich auf Bedingungen, die sich auf die Leistung von Anwendungen auswirken können, die eine angegebene Warteschlange verwenden. Der Umfang der Leistungsereignisse ist die Warteschlange. **MQPUT** -Aufrufe und **MQGET** -Aufrufe in einer Warteschlange haben keine Auswirkungen auf die Generierung von Leistungsereignissen in einer anderen Warteschlange.

„Der Servicezeitgeber“ auf Seite 24

Ereignisse des Warteschlangenserviceintervalls verwenden einen internen Zeitgeber, der als *Servicezeitgeber* bezeichnet wird und der vom Warteschlangenmanager gesteuert wird. Der Zeitgeber für den Service wird nur verwendet, wenn ein Warteschlangenserviceintervall aktiviert ist.

„Regeln für Warteschlangenserviceintervallereignisse“ auf Seite 24

Formale Regelsteuerung, wenn der Zeitgeber für die Servicezeitgeber festgelegt ist und die Intervallereignisse für den Warteschlangenservice generiert werden.

### **Zugehörige Tasks**

„Warteschlangenserviceintervallereignisse aktivieren“ auf Seite 25

Um eine Warteschlange für Warteschlangenserviceintervallereignisse zu konfigurieren, legen Sie die entsprechenden Warteschlangenmanager- und Warteschlangenattribute fest.

### **Zugehörige Verweise**

[Hohe Warteschlangenlänge](#)

[Warteschlangenstatistik zurücksetzen](#)

[RESET QSTATS](#)

### **Ereignis 'Intervall für Warteschlangenwartung'**

Ereignisse des Warteschlangenserviceintervalls geben an, ob eine Operation in einer Warteschlange innerhalb eines benutzerdefinierten Zeitintervalls, das als *Serviceintervall* bezeichnet wird, ausgeführt

wurde. Abhängig von Ihrer Installation können Sie Warteschlangenserviceintervall-Ereignisse verwenden, um zu überwachen, ob Nachrichten schnell genug in die Warteschlange genommen werden.

Ereignisse des Warteschlangenserviceintervalls werden in gemeinsam genutzten Warteschlangen nicht unterstützt.

Die folgenden Typen von Warteschlangenserviceintervallereignissen können auftreten, wenn der Begriff *Abrufoperation* auf einen **MQGET** -Aufruf oder eine Aktivität verweist, der bzw. die Nachrichten aus einer Warteschlange entfernt, z. B. mit dem Befehl **CLEAR QLOCAL** :

#### **Warteschlangenserviceintervall OK**

Gibt an, dass nach einer der folgenden Operationen:

- Ein MQPUT-Aufruf
- Eine Operation 'get', die eine nicht leere Warteschlange verlässt.

Es wurde eine Operation 'get' innerhalb eines benutzerdefinierten Zeitraums ausgeführt, der als *Serviceintervall* bezeichnet wird.

Nur eine Operation 'get' kann die Ereignisnachricht 'Warteschlangenserviceintervall OK' auslösen. Ereignisse des Warteschlangenserviceintervalls OK werden manchmal als OK-Ereignisse bezeichnet.

#### **Hohes Warteschlangenserviceintervall**

Gibt an, dass nach einer der folgenden Operationen:

- Ein MQPUT-Aufruf
- Eine Operation 'get', die eine nicht leere Warteschlange verlässt.

Eine get-Operation wurde **nicht** innerhalb eines benutzerdefinierten Serviceintervalls ausgeführt.

Entweder eine get-Operation oder ein MQPUT-Aufruf kann die Ereignisnachricht 'Warteschlangenserviceintervall hoch' auslösen. Ereignisse des Warteschlangenserviceintervalls 'Hoch' werden manchmal als 'Hoch' beschrieben.

Wenn Sie die beiden Ereignisse "Queue Service Interval OK" und "Queue Service Interval High" aktivieren möchten, setzen Sie das Steuerattribut `QServiceIntervalEvent` auf Hoch. Ereignisse des Warteschlangenserviceintervalls OK werden automatisch aktiviert, wenn ein Ereignis Warteschlangenserviceintervall hoch generiert wird. Sie müssen Warteschlangenserviceintervall OK-Ereignisse nicht unabhängig voneinander aktivieren.

OK und die Ereignisse 'Hoch' schließen sich gegenseitig aus, so dass die anderen Ereignisse inaktiviert sind. Beide Ereignisse können jedoch gleichzeitig inaktiviert werden.

Abbildung 3 auf Seite 23 zeigt ein Diagramm der Warteschlangenlänge gegen die Zeit an. Zum Zeitpunkt P1 gibt eine Anwendung einen MQPUT-Befehl aus, um eine Nachricht in die Warteschlange zu stellen. Zum Zeitpunkt G1 gibt eine andere Anwendung einen MQGET-Befehl aus, um die Nachricht aus der Warteschlange zu entfernen.

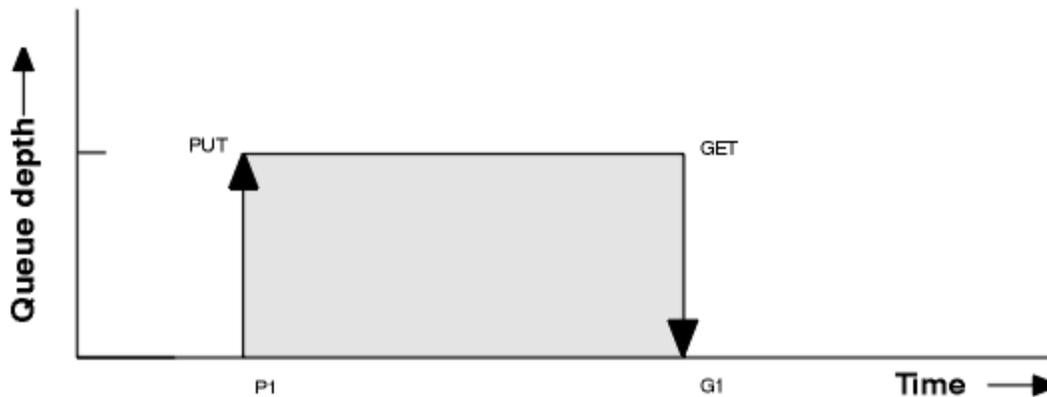


Abbildung 3. Informationen zum Warteschlangenserviceintervall

Die möglichen Ergebnisse von Warteschlangenserviceintervall-Ereignissen lauten wie folgt:

- Wenn die abgelaufene Zeit zwischen dem put- und dem get-Wert kleiner-gleich dem Serviceintervall ist:
  - Ein Ereignis *Warteschlangenserviceintervall OK* wird zum Zeitpunkt G1 generiert, wenn die Warteschlangenserviceintervallereignisse aktiviert sind.
- Wenn die abgelaufene Zeit zwischen dem put- und dem get-Wert größer als das Serviceintervall ist:
  - Ein Ereignis *Warteschlangenserviceintervall hoch* wird zum Zeitpunkt G1 generiert, wenn die Warteschlangenserviceintervallereignisse aktiviert sind.

Der Algorithmus zum Starten des Service-Zeitgebers und zum Generieren von Ereignissen wird in „Regeln für Warteschlangenserviceintervallereignisse“ auf Seite 24 beschrieben.

### **Zugehörige Verweise**

[Warteschlangenserviceintervall OK](#)

[Hohes Warteschlangenserviceintervall](#)

[QServiceIntervalEvent \(MQLONG\)](#)

[QServiceIntervalEvent \(10-stellige Ganzzahl mit Vorzeichen\)](#)

[Eigenschaft ServiceIntervalEvent](#)

### *Der Servicezeitgeber*

Ereignisse des Warteschlangenserviceintervalls verwenden einen internen Zeitgeber, der als *Servicezeitgeber* bezeichnet wird und der vom Warteschlangenmanager gesteuert wird. Der Zeitgeber für den Service wird nur verwendet, wenn ein Warteschlangenserviceintervall aktiviert ist.

### **Was genau misst die Service-Timer-Messung?**

Der Servicezeitgeber misst die abgelaufene Zeit zwischen einem MQPUT-Aufruf an eine leere Warteschlange oder eine get-Operation und die nächste PUT- oder GET-Operation, vorausgesetzt, die Warteschlangenlänge ist ungleich null zwischen diesen beiden Operationen.

### **Wann ist der Servicezeitgeber aktiv?**

Der Servicezeitgeber ist immer aktiv (aktiv), wenn die Warteschlange Nachrichten enthält (Tiefe ist ungleich null) und ein Warteschlangenserviceintervall-Ereignis aktiviert ist. Wenn die Warteschlange leer wird (Warteschlangenlänge null), wird der Zeitgeber in den Status OFF versetzt, um bei der nächsten Einschaltung erneut gestartet zu werden.

### **Wann wird der Service-Timer zurückgesetzt?**

Der Servicezeitgeber wird immer nach einer get-Operation zurückgesetzt. Sie wird auch durch einen MQPUT-Aufruf in eine leere Warteschlange zurückgesetzt. Es wird jedoch nicht unbedingt auf ein Warteschlangenserviceintervall zurückgesetzt.

### **Wie wird der Service-Timer verwendet?**

Nach einer Abrufoperation oder einem MQPUT-Aufruf vergleicht der Warteschlangenmanager die abgelaufene Zeit, die vom Servicegeber gemessen wurde, mit dem benutzerdefinierten Serviceintervall. Das Ergebnis dieses Vergleichs ist Folgendes:

- Es wird ein OK-Ereignis generiert, wenn eine get-Operation vorhanden ist und die abgelaufene Zeit kleiner-gleich dem Serviceintervall ist UND dieses Ereignis aktiviert ist.
- Es wird ein hohes Ereignis generiert, wenn die abgelaufene Zeit größer als das Serviceintervall ist UND dieses Ereignis aktiviert ist.

### **Können Anwendungen den Servicezeitgeber lesen?**

Nein, der Zeitgeber für den Service ist ein interner Zeitgeber, der für Anwendungen nicht verfügbar ist.

### **Was ist mit dem Parameter *TimeSinceReset* ?**

Der Parameter *TimeSinceReset* wird in den Ereignisdaten in den Ereignisdaten als Teil der Ereignisstatistik zurückgegeben. Sie gibt die Zeit zwischen aufeinanderfolgenden Warteschlangenserviceintervall-Ereignissen an, es sei denn, die Ereignisstatistik wird zurückgesetzt.

### *Regeln für Warteschlangenserviceintervallereignisse*

Formale Regelsteuerung, wenn der Zeitgeber für die Servicezeitgeber festgelegt ist und die Intervallereignisse für den Warteschlangenservice generiert werden.

## Regeln für den Servicezeitgeber

Der Zeitgeber für den Service wird auf null zurückgesetzt und wie folgt erneut gestartet:

- Nach einem MQPUT-Aufruf an eine leere Warteschlange.
- Nach einem MQGET-Aufruf, wenn die Warteschlange nach dem MQGET-Aufruf nicht leer ist.

Das Zurücksetzen des Zeitgebers hängt nicht davon ab, ob ein Ereignis generiert wurde.

Beim Start des Warteschlangenmanagers wird der Zeitgeber für den Service auf den Startzeitpunkt gesetzt, wenn die Warteschlangenlänge größer als null ist.

Wenn die Warteschlange nach einer get-Operation leer ist, wird der Zeitgeber in den Status OFF versetzt.

## Ereignis 'Queue Service Interval High'

Das Ereignis "Warteschlangenserviceintervall" muss aktiviert sein (muss auf HIGH gesetzt werden).

Ereignisse des Warteschlangenserviceintervalls 'Hoch' werden automatisch aktiviert, wenn ein Ereignis 'Queue Service Interval OK' generiert wird.

Wenn die Servicezeit größer als das Serviceintervall ist, wird ein Ereignis für die nächste MQPUT- oder GET-Operation generiert oder vor der nächsten MQPUT-Operation.

## Warteschlangenserviceintervall OK-

Ereignisse des Warteschlangenserviceintervalls OK werden automatisch aktiviert, wenn ein Ereignis Warteschlangenserviceintervall hoch generiert wird.

Wenn die Servicezeit (abgelaufene Zeit) kleiner-gleich dem Serviceintervall ist, wird ein Ereignis generiert oder vor der nächsten Operation 'get'.

### Zugehörige Tasks

„Warteschlangenserviceintervallereignisse aktivieren“ auf Seite 25

Um eine Warteschlange für Warteschlangenserviceintervallereignisse zu konfigurieren, legen Sie die entsprechenden Warteschlangenmanager- und Warteschlangenattribute fest.

*Warteschlangenserviceintervallereignisse aktivieren*

Um eine Warteschlange für Warteschlangenserviceintervallereignisse zu konfigurieren, legen Sie die entsprechenden Warteschlangenmanager- und Warteschlangenattribute fest.

## Informationen zu diesem Vorgang

Die Ereignisse "high" und "OK" schließen sich gegenseitig aus, d. a. wenn eine aktiviert ist, wird die andere automatisch inaktiviert:

- Wenn ein hohes Ereignis in einer Warteschlange generiert wird, inaktiviert der Warteschlangenmanager automatisch hohe Ereignisse und aktiviert die OK-Ereignisse für diese Warteschlange.
- Wenn ein OK-Ereignis in einer Warteschlange generiert wird, inaktiviert der Warteschlangenmanager automatisch die OK-Ereignisse und aktiviert hohe Ereignisse für diese Warteschlange.

Ereignis für Warteschlangenserviceinterv	Warteschlangenattribute
Hohes Warteschlangenserviceintervall	QSVCIHV (HIGH)
Warteschlangenserviceintervall OK	QSVCIHV (OK)
Keine Ereignisse des Warteschlangenserviceinterv	QSVCIHV (NONE)

Tabelle 4. Warteschlangenserviceintervallereignisse unter Verwendung von MQSC aktivieren (Forts.)	
Ereignis für Warteschlangenserviceinterv	Warteschlangenattribute
Serviceintervall	QSVCINT ( <i>tt</i> ); hierbei steht <i>tt</i> für den ServiceIntervallzeit in Millisekunden.

Führen Sie die folgenden Schritte aus, um Warteschlangenserviceintervallereignisse zu aktivieren

## Vorgehensweise

1. Setzen Sie das WS-Managerattribut **PERFMEV** auf ENABLED .  
Leistungsereignisse werden auf dem Warteschlangenmanager aktiviert.
2. Setzen Sie das Steuerattribut **QSVCIIEV** für ein Warteschlangenserviceintervall-Hoch-oder OK-Ereignis in der Warteschlange nach Bedarf.
3. Legen Sie das Attribut **QSVCINT** für die Warteschlange fest, um die entsprechende Serviceintervallzeit anzugeben.

## Beispiel

Verwenden Sie die folgenden MQSC-Befehle, um Warteschlangenserviceintervall-Hoch-Ereignisse mit einer Serviceintervallzeit von 10 Sekunden (10 000 Millisekunden) zu aktivieren:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QSVCINT(10000) QSVCIIEV(HIGH)
```

## Beispiele für Warteschlangenserviceintervall

Verwenden Sie die Beispiele in diesem Abschnitt, um die Informationen zu verstehen, die Sie aus Warteschlangenserviceintervall-Ereignissen abrufen können.

Die drei Unterthemenbeispiele stellen progressiv komplexere Darstellungen der Verwendung von Warteschlangenserviceintervallereignissen bereit.

Die Abbildungen, die die Beispiele in den einzelnen Unterthemen begleiten, haben dieselbe Struktur:

- Abbildung 1 ist ein Diagramm der Warteschlangenlänge mit der Zeit, in dem einzelne MQGET-Aufrufe und MQPUT-Aufrufe angezeigt werden.
- Im Abschnitt "Kommentar" wird ein Vergleich der Zeitvorgaben angezeigt. Es gibt drei Zeiträume, in denen Sie Folgendes beachten müssen:
  - Das vom Benutzer definierte Serviceintervall.
  - Die Zeit, die vom Servicezeitgeber gemessen wird.
  - Die Zeit, seit die Ereignisstatistik zuletzt zurückgesetzt wurde (TimeSinceReset in den Ereignisdaten).
- Im Übersichtsabschnitt Ereignisstatistik wird angezeigt, welche Ereignisse zu einem beliebigen Zeitpunkt aktiviert sind und welche Ereignisse generiert werden.

Die Beispiele veranschaulichen die folgenden Aspekte von Warteschlangenserviceintervall-Ereignissen:

- Gibt an, wie sich die Warteschlangenlänge im Laufe der Zeit ändert
- Gibt an, wie die vom Servicezeitgeber gemessene abgelaufene Zeit mit dem Serviceintervall verglichen wird.
- Welches Ereignis aktiviert ist.
- Welche Ereignisse generiert werden.

**Hinweis:** Beispiel 1 zeigt einen einfachen Fall, bei dem die Nachrichten intermittierend sind und jede Nachricht aus der Warteschlange entfernt wird, bevor die nächste empfangen wird. Aus den Ereignisdaten wissen Sie, dass die maximale Anzahl an Nachrichten in der Warteschlange eins war. Sie können also herausfinden, wie lange jede Nachricht in der Warteschlange war.

Wenn jedoch im allgemeinen Fall mehrere Nachrichten in der Warteschlange vorhanden sind und die Reihenfolge von MQGET-Aufrufen und MQPUT-Aufrufen nicht vorhersagbar ist, können Sie keine Warteschlangenserviceintervallereignisse verwenden, um zu berechnen, wie lange eine einzelne Nachricht in einer Warteschlange verbleibt. Der Parameter 'TimeSinceReset', der in den Ereignisdaten zurückgegeben wird, kann einen Teil der Zeit enthalten, wenn keine Nachrichten in der Warteschlange vorhanden sind. Daher werden alle Ergebnisse, die Sie aus diesen Statistiken ableiten, implizit gemittelt, um diese Zeiten einzuschließen.

### **Zugehörige Konzepte**

„Ereignis 'Intervall für Warteschlangenwartung'“ auf Seite 22

Ereignisse des Warteschlangenserviceintervalls geben an, ob eine Operation in einer Warteschlange innerhalb eines benutzerdefinierten Zeitintervalls, das als *Serviceintervall* bezeichnet wird, ausgeführt wurde. Abhängig von Ihrer Installation können Sie Warteschlangenserviceintervall-Ereignisse verwenden, um zu überwachen, ob Nachrichten schnell genug in die Warteschlange genommen werden.

„Der Servicezeitgeber“ auf Seite 24

Ereignisse des Warteschlangenserviceintervalls verwenden einen internen Zeitgeber, der als *Servicezeitgeber* bezeichnet wird und der vom Warteschlangenmanager gesteuert wird. Der Zeitgeber für den Service wird nur verwendet, wenn ein Warteschlangenserviceintervall aktiviert ist.

*Ereignisse des Warteschlangenserviceintervalls: Beispiel 1*

Eine Grundfolge von MQGET-Aufrufen und MQPUT-Aufrufen, bei denen die Warteschlangenlänge immer eins oder null ist.

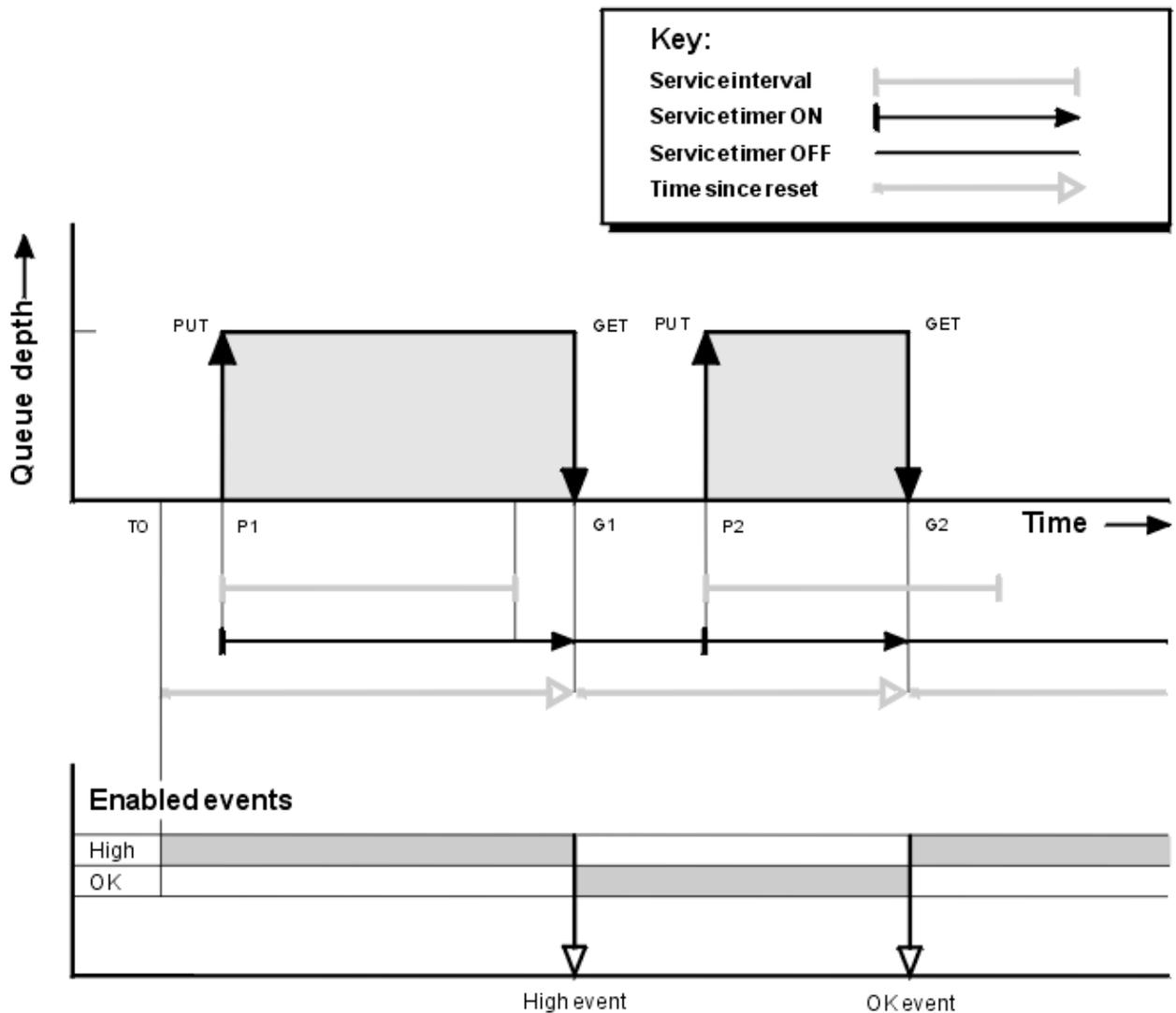


Abbildung 4. Ereignisse des Warteschlangenserviceintervalls-Beispiel 1

### Commentary

1. Bei P1 reiht eine Anwendung eine Nachricht in eine leere Warteschlange ein. Damit wird der Servicezeitgeber gestartet.

Beachten Sie, dass T0 möglicherweise die Startzeit des Warteschlangenmanagers ist.

2. Bei G1 ruft eine andere Anwendung die Nachricht aus der Warteschlange ab. Da die abgelaufene Zeit zwischen P1 und G1 größer als das Serviceintervall ist, wird im MQGET-Aufruf von G1 ein Ereignis des Typs "Queue Service Interval High" (Warteschlangenserviceintervall) generiert. Wenn das Ereignis "Hoch" generiert wird, setzt der Warteschlangenmanager das Ereignissteuerungsattribut so zurück, dass
  - a. Das OK-Ereignis wird automatisch aktiviert.
  - b. Das Ereignis "Hoch" ist inaktiviert.

Da die Warteschlange jetzt leer ist, wird der Servicezeitgeber in den Status OFF (Aus) umgeschaltet.

3. Bei P2 wird eine zweite Nachricht in die Warteschlange gestellt. Damit wird der Servicezeitgeber erneut gestartet.
4. Bei G2 wird die Nachricht aus der Warteschlange entfernt. Da jedoch die abgelaufene Zeit zwischen P2 und G2 kleiner als das Serviceintervall ist, wird im MQGET-Aufruf von G2 ein Ereignis 'Warteschlangen-

serviceintervall OK' generiert. Wenn das OK-Ereignis generiert wird, setzt der Warteschlangenmanager das Steuerattribut so zurück, dass

- a. Das hohe Ereignis wird automatisch aktiviert.
- b. Das OK-Ereignis ist inaktiviert.

Da die Warteschlange leer ist, wird der Servicezeitgeber wieder in den Status OFF (Aus) umgeschaltet.

## Zusammenfassung der Ereignisstatistik

In [Tabelle 5 auf Seite 29](#) sind die Ereignisstatistikdaten für dieses Beispiel zusammengefasst.

<i>Tabelle 5. Ereignisstatistikzusammenfassung für Beispiel 1</i>		
<b>Attribut</b>	<b>Ereignis 1</b>	<b>Ereignis 2</b>
Zeitpunkt des Ereignisses	T (G1)	T (G2)
Typ des Ereignisses	Hoch	OK
Zurücksetzungs-zeit (TimeS)	T (G1)-T (0)	T (G2)-T (G1)
Höchste QDepth	1	1
MsgEnqCount	1	1
MsgDeqCount	1	1

Der mittlere Teil von [Abbildung 4 auf Seite 28](#) zeigt die abgelaufene Zeit an, die vom Servicezeitgeber gemessen wurde, verglichen mit dem Serviceintervall für diese Warteschlange. Um zu sehen, ob ein Warteschlangenserviceintervall-Ereignis auftreten kann, vergleichen Sie die Länge der horizontalen Linie, die den Servicezeitgeber darstellt (mit Pfeil), mit der Länge der Linie, die das Serviceintervall darstellt. Wenn die Servicezeitgeberzeile länger ist und das Ereignis "Hoch" des Warteschlangenserviceintervalls aktiviert ist, wird beim nächsten get ein Ereignis "Queue Service Interval High" (Warteschlangenserviceintervall hoch) angezeigt. Wenn die Zeitgeberzeile kürzer ist und das Ereignis 'Warteschlangenserviceintervall OK' aktiviert ist, wird beim nächsten get ein Ereignis des Warteschlangenserviceintervalls OK ausgeführt.

### *Ereignisse des Warteschlangenserviceintervalls: Beispiel 2*

Eine Folge von MQPUT-Aufrufen und MQGET-Aufrufen, bei denen die Warteschlangenlänge nicht immer eine oder null ist.

In diesem Beispiel werden auch Instanzen des Zeitgebers angezeigt, die zurückgesetzt werden, ohne dass Ereignisse generiert werden, z. B. zum Zeitpunkt P2.

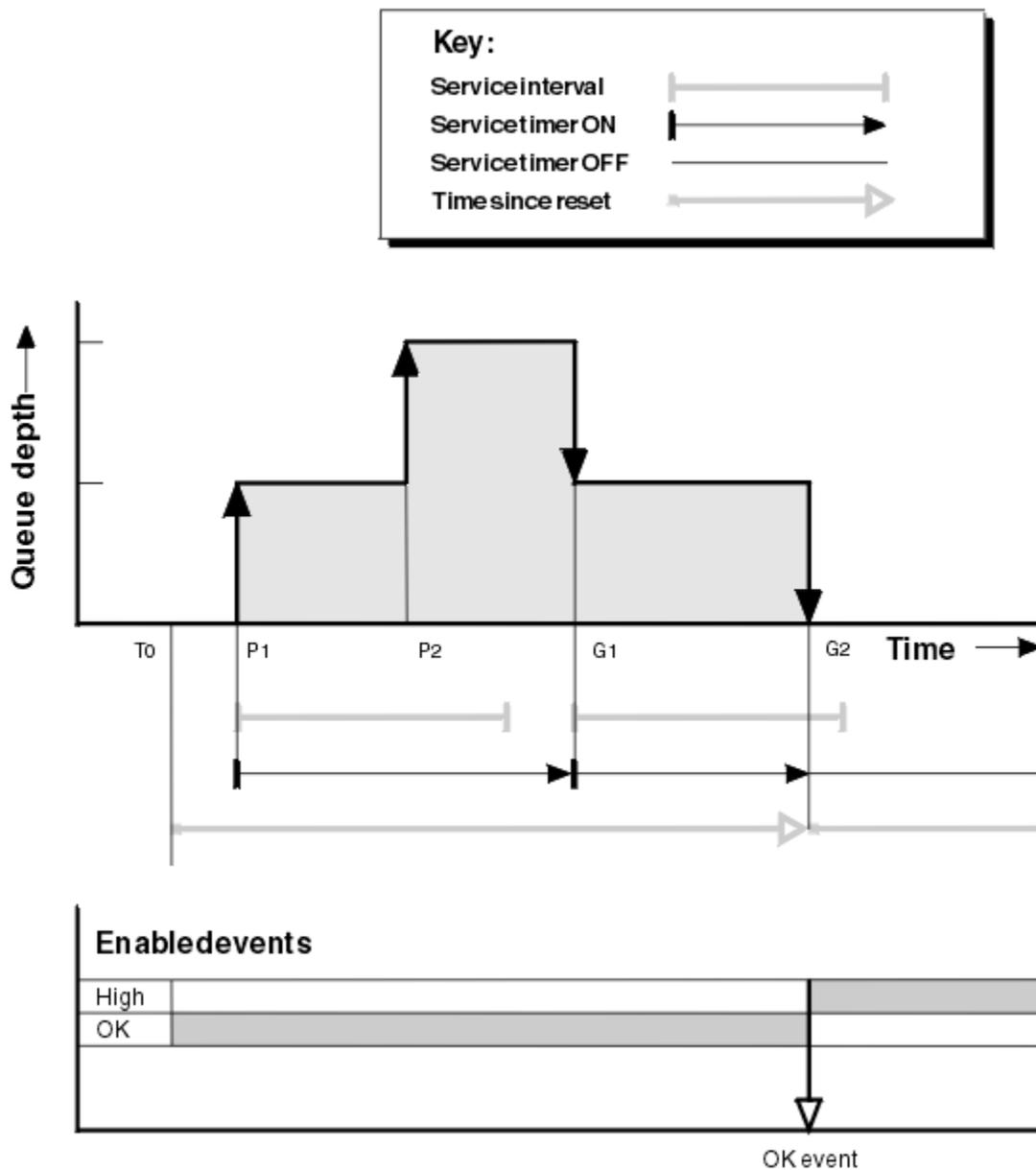


Abbildung 5. Ereignisse des Warteschlangenserviceintervalls-Beispiel 2

### Commentary

In diesem Beispiel sind OK-Ereignisse aktiviert, und die Warteschlangenstatistik wurde zum Zeitpunkt T0 zurückgesetzt.

1. Bei P1 startet der erste put den Service-Timer.
2. Bei P2 generiert die zweite put kein Ereignis, da ein put kein OK-Ereignis verursachen kann.
3. Bei G1 wurde das Serviceintervall jetzt überschritten, und daher wird kein OK-Ereignis generiert. Der MQGET-Aufruf bewirkt jedoch, dass der Servicezeitgeber zurückgesetzt wird.
4. Bei G2 erfolgt die zweite Aktion innerhalb des Serviceintervalls, und dieses Mal wird ein OK-Ereignis generiert. Der Warteschlangenmanager setzt das Ereignissteuerungsattribut so zurück, dass
  - a. Das hohe Ereignis wird automatisch aktiviert.
  - b. Das OK-Ereignis ist inaktiviert.

Da die Warteschlange jetzt leer ist, wird der Servicezeitgeber in den Status OFF (Aus) umgeschaltet.

## Zusammenfassung der Ereignisstatistik

In [Tabelle 6 auf Seite 31](#) sind die Ereignisstatistikdaten für dieses Beispiel zusammengefasst.

Attribut	Ereignis 2
Zeitpunkt des Ereignisses	T (G2)
Typ des Ereignisses	OK
Zurücksetzungs-zeit (TimeS)	T (G2)-T (0)
Höchste QDepth	2
MsgEnqCount	2
MsgDeqCount	2

### Ereignisse des Warteschlangenserviceintervalls: Beispiel 3

Eine Folge von MQGET-Aufrufen und MQPUT-Aufrufen, die sporadischer als die vorherigen Beispiele sind.

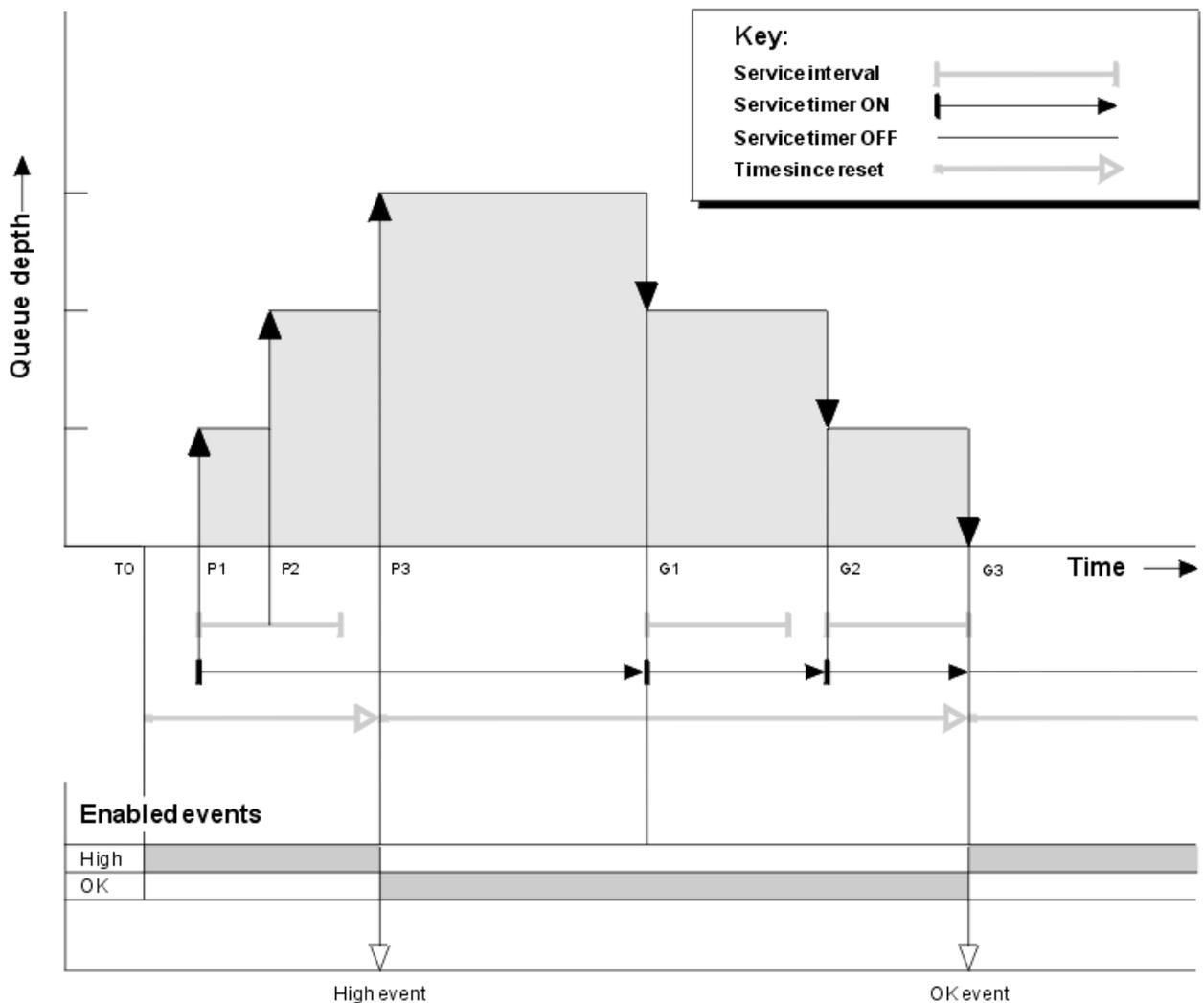


Abbildung 6. Ereignisse des Warteschlangenserviceintervalls-Beispiel 3

## Commentary

1. Zum Zeitpunkt T (0) werden die Warteschlangenstatistikdaten zurückgesetzt und das Warteschlangenserviceintervall Hoch-Ereignisse aktiviert.
2. Bei P1 startet der erste put den Service-Timer.
3. Bei P2 erhöht die zweite Position die Warteschlangenlänge auf zwei. Ein hohes Ereignis wird hier nicht generiert, da die Serviceintervallzeit nicht überschritten wurde.
4. Bei P3 verursacht das dritte Ereignis ein hohes Ereignis, das generiert werden soll. (Der Zeitgeber hat das Serviceintervall überschritten.) Der Zeitgeber wird nicht zurückgesetzt, da die Warteschlangenlänge vor dem Einlegen nicht null war. Es sind jedoch OK-Ereignisse aktiviert.
5. Bei G1 generiert der MQGET-Aufruf kein Ereignis, da das Serviceintervall überschritten wurde und OK-Ereignisse aktiviert sind. Der MQGET-Aufruf setzt jedoch den Servicezeitgeber zurück.
6. Bei G2 generiert der MQGET-Aufruf kein Ereignis, da das Serviceintervall überschritten wurde und OK-Ereignisse aktiviert sind. Auch hier setzt der MQGET-Aufruf den Servicezeitgeber zurück.
7. Bei G3 wird der dritte Wert für die Warteschlange geleert und der Servicezeitgeber ist gleich dem Serviceintervall. Daher wird ein OK-Ereignis generiert. Der Zeitgeber für den Service wird zurückgesetzt, und es werden hohe Ereignisse aktiviert. Mit dem MQGET-Aufruf wird die Warteschlange geleert, und dadurch wird der Zeitgeber in den Status OFF versetzt.

## Zusammenfassung der Ereignisstatistik

In Tabelle 7 auf Seite 32 sind die Ereignisstatistikdaten für dieses Beispiel zusammengefasst.

<b>Attribut</b>	<b>Ereignis 1</b>	<b>Ereignis 2</b>
Zeitpunkt des Ereignisses	T (P3)	T (G3)
Typ des Ereignisses	Hoch	OK
Zurücksetzungs-zeit (TimeS)	T (P3)-T (0)	T (G3)-T (P3)
Höchste QDepth	3	3
MsgEnqCount	3	0
MsgDeqCount	0	3

### **Ereignisse der Warteschlangenlänge**

Ereignisse der Warteschlangenlänge beziehen sich auf die Warteschlangenlänge, d. h. die Anzahl der Nachrichten in der Warteschlange.

In IBM MQ-Anwendungen dürfen Warteschlangen nicht voll werden. Wenn dies der Fall ist, können Anwendungen keine Nachrichten mehr in die Warteschlange stellen, die sie angeben. Auch wenn die Nachricht nicht verloren geht, kann eine vollständige Warteschlange erhebliche Unannehmlichkeiten verursachen. Die Anzahl der Nachrichten kann in einer Warteschlange erstellt werden, wenn die Nachrichten schneller in die Warteschlange gestellt werden, als die Anwendungen, die sie verarbeiten, sie abschalten können.

Die Lösung für dieses Problem hängt von den jeweiligen Umständen ab, kann jedoch Folgendes beinhalten:

- Einige Nachrichten werden in eine andere Warteschlange gesendet.
- Neue Anwendungen werden gestartet, um mehr Nachrichten aus der Warteschlange zu nehmen.
- Der nicht wesentliche Nachrichtenverkehr wird gestoppt.
- Erhöhen Sie die Warteschlangenlänge, um ein temporales Maximum zu überwinden.

Die Vorabwarnung, dass Probleme auf dem Weg sind, macht es einfacher, vorbeugende Maßnahmen zu ergreifen. Zu diesem Zweck stellt IBM MQ die folgenden Warteschlangenlängeneignisse bereit:

### Warteschlangenlänge hoch-Ereignisse

Geben Sie an, dass die Warteschlangenlänge auf einen vordefinierten Schwellenwert erhöht wurde, der als "Warteschlangenlänge hoch" bezeichnet wird.

### Ereignisse mit Warteschlangenlänge niedrig

Geben Sie an, dass die Warteschlangenlänge auf einen vordefinierten Schwellenwert, der als "Warteschlangenlänge niedrig" bezeichnet wird, verringert wurde.

### Ereignisse in Warteschlange voll

Geben Sie an, dass die Warteschlange ihre maximale Tiefe erreicht hat, d. -d. die Warteschlange voll ist.

Ein Ereignis 'Warteschlange voll' wird generiert, wenn eine Anwendung versucht, eine Nachricht in eine Warteschlange zu stellen, die ihre maximale Tiefe erreicht hat. Ereignisse vom Typ "Queue Depth High" geben Vorwarnung an, dass eine Warteschlange gefüllt wird. Dies bedeutet, dass der Systemadministrator eine vorbeugende Maßnahme ergreifen muss, nachdem er dieses Ereignis empfangen hat. Sie können den Warteschlangenmanager so konfigurieren, dass der Warteschlangenmanager, wenn die Präventivaktion erfolgreich ist und die Warteschlangenlänge auf eine sicherere Ebene fällt, ein Ereignis "Queue Depth Low" (Warteschlangenlänge niedrig) generiert.

Das erste Beispiel für eine Warteschlangenlänge zeigt die Auswirkungen mutmaßlicher Aktionen an, die verhindern, dass die Warteschlange voll wird.

### Zugehörige Konzepte

„Beispiele für Ereignisse der Warteschlangenlänge“ auf Seite 37

Verwenden Sie die folgenden Beispiele, um die Informationen zu verstehen, die Sie von Warteschlangentiefenereignissen erhalten können.

### Zugehörige Verweise

[Warteschlange voll](#)

[Hohe Warteschlangenlänge](#)

[Niedrige Warteschlangenlänge](#)

### *Warteschlangenlänge aktivieren*

Zum Konfigurieren einer Warteschlange für beliebige Warteschlangenlänge-Ereignisse legen Sie die entsprechenden Warteschlangenmanager- und Warteschlangenattribute fest.

### Informationen zu diesem Vorgang

Standardmäßig sind alle Warteschlangenlänge-Ereignisse inaktiviert. Wenn diese Option aktiviert ist, werden die Warteschlangentiefenereignisse wie folgt generiert:

- Ein Ereignis vom Typ "Queue Depth High" wird generiert, wenn eine Nachricht in die Warteschlange gestellt wird, wodurch die Warteschlangenlänge größer oder gleich dem Wert ist, der für **QDepthHighLimit** festgelegt wurde.
  - Ein Ereignis vom Typ "Queue Depth High" wird automatisch durch ein Ereignis "Queue Depth Low" in derselben Warteschlange aktiviert.
  - Ein Ereignis vom Typ "Queue Depth High" aktiviert automatisch sowohl eine Warteschlangenlänge als auch ein Ereignis "Warteschlange voll" in derselben Warteschlange.
-  Ein Ereignis 'Warteschlangenlänge niedrig' wird generiert, wenn eine Nachricht durch eine GET-Operation aus einer Warteschlange entfernt wird, wodurch die Warteschlangenlänge kleiner-gleich dem für **QDepthLowLimit** festgelegten Wert ist.
-  Ein Ereignis vom Typ "Queue Depth Low" wird generiert, wenn eine Nachricht aus einer Warteschlange durch eine get-Operation entfernt wird oder entfernt worden wäre, aber abgelaufen ist, wodurch die Warteschlangenlänge kleiner gleich dem Wert für **QDepthLowLimit** ist.
  - Ein Ereignis vom Typ "Queue Depth Low" wird automatisch durch ein Ereignis "Queue Depth High" (Warteschlangenlänge hoch) oder ein Ereignis "Warteschlange voll" in derselben Warteschlange aktiviert.

- Ein Ereignis "Queue Depth Low" aktiviert automatisch sowohl eine Warteschlangenlänge als auch ein Ereignis "Warteschlange voll" in derselben Warteschlange.
- Ein Ereignis 'Warteschlange voll' wird generiert, wenn eine Anwendung keine Nachricht in eine Warteschlange einlegen kann, weil die Warteschlange voll ist.
  - Ein Ereignis 'Warteschlange voll' wird automatisch durch eine Warteschlangenlänge hoch oder ein Ereignis 'Warteschlangenlänge niedrig' in derselben Warteschlange aktiviert.
  - Ein Ereignis 'Warteschlange voll' aktiviert automatisch ein Ereignis 'Queue Depth Low' (Warteschlangenlänge niedrig) in derselben Warteschlange.

Führen Sie die folgenden Schritte aus, um eine Warteschlange für alle Warteschlangenlänge-Ereignisse zu konfigurieren:

## Vorgehensweise

1. Aktivieren Sie die Leistungsereignisse auf dem Warteschlangenmanager unter Verwendung des Warteschlangenmanagerattributs **PERFMEV**.

Die Ereignisse werden in die SYSTEM.ADMIN.PERFM.EVENT-Warteschlange gestellt.

2. Legen Sie eines der folgenden Attribute fest, um das Ereignis in der erforderlichen Warteschlange zu aktivieren:

- *QDepthHighEvent* (**QDPHIEV** in MQSC)
- *QDepthLowEvent* (**QDPLOEV** in MQSC)
- *QDepthMaxEvent* (**QDPMAXEV** in MQSC)

3. Optional: Wenn Sie die Grenzwerte festlegen möchten, ordnen Sie die folgenden Attribute als Prozentsatz der maximalen Warteschlangenlänge zu:

- *QDepthHighLimit* (**QDEPTHHI** in MQSC)
- *QDepthLowLimit* (**QDEPTHLO** in MQSC)

**Einschränkung:** **QDEPTHHI** darf nicht kleiner als **QDEPTHLO** sein.

Wenn **QDEPTHHI** gleich **QDEPTHLO** ist, wird jedesmal, wenn die Warteschlangenlänge den Wert in beide Richtungen übergibt, eine Ereignisnachricht generiert, da der hohe Schwellenwert aktiviert ist, wenn die Warteschlangenlänge unter dem Wert liegt und der niedrige Schwellenwert aktiviert ist, wenn die Tiefe über dem Wert liegt.

## Ergebnisse

### Anmerkung:

**Multi** Ein Ereignis 'Warteschlangenlänge niedrig' wird nicht generiert, wenn abgelaufene Nachrichten aus einer Warteschlange durch eine Operation 'GET' entfernt werden, die dazu führt, dass die Warteschlangenlänge kleiner als oder gleich dem Wert ist, der durch den Wert für **QDepthLowLimit** festgelegt wird. IBM MQ generiert die Nachricht mit dem Ereignis 'Warteschlangenlänge niedrig' nur während einer erfolgreichen GET-Operation. Wenn die abgelaufenen Nachrichten aus der Warteschlange entfernt werden, wird daher keine Nachricht über eine niedrige Warteschlangenlänge generiert. Darüber hinaus werden nach dem Entfernen dieser abgelaufenen Nachrichten aus der Warteschlange *QDepthHighEvent* und *QDepthLowEvent* nicht zurückgesetzt.

**z/OS** IBM MQ generiert die Nachricht mit dem Ereignis 'Warteschlangenlänge niedrig' während einer erfolgreichen destruktiven GET-Operation oder einer destruktiven GET-Operation, die erfolgreich gewesen wäre, wenn eine entsprechende Nachricht nicht abgelaufen wäre. Wenn abgelaufene Nachrichten während einer allgemeinen Hintergrundverarbeitung aus der Warteschlange entfernt werden, wird keine Nachricht über eine niedrige Warteschlangenlänge generiert. Darüber hinaus werden nach dem Entfernen abgelaufener Nachrichten aus der Warteschlange während der allgemeinen Hintergrundverarbeitung *QDepthHighEvent* und *QDepthLowEvent* nicht zurückgesetzt. Weitere Informationen zur Verarbeitung von abgelaufenen Nachrichten finden Sie unter [Warteschlangenmanager unter IBM MQ for z/OS optimieren](#).

## Beispiel

Verwenden Sie die folgenden MQSC-Befehle, um Warteschlangenlänge-Hoch-Ereignisse in der Warteschlange MYQUEUE mit einem Grenzwert von 80% zu aktivieren:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHHI(80) QDPHIEV(ENABLED)
```

Verwenden Sie die folgenden MQSC-Befehle, um die Warteschlangenlänge in der Warteschlange MYQUEUE mit einem Grenzwert von 20% zu aktivieren:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHLO(20) QDPLOEV(ENABLED)
```

Verwenden Sie die folgenden MQSC-Befehle, um die Warteschlange Vollständige Ereignisse in der Warteschlange MYQUEUE zu aktivieren:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDPMAXEV(ENABLED)
```

 **Gemeinsam genutzte Warteschlangen und Warteschlangenlängenergebnisse unter z/OS**  
Unter IBM MQ for z/OS ist die Ereignisüberwachung für eine Anwendung mit gemeinsam genutzten Warteschlangen einfacher, wenn alle Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange die gleiche Einstellung für das Attribut PERFMEV verwenden.

Wenn ein Warteschlangenlängenergebnis in einer gemeinsam genutzten Warteschlange auftritt und das Warteschlangenmanagerattribut **PERFMEV** auf ENABLED gesetzt ist, erzeugen die Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange eine Ereignisnachricht. Wenn **PERFMEV** auf einigen Warteschlangenmanagern auf DISABLED gesetzt ist, werden Ereignisnachrichten nicht von diesen Warteschlangenmanagern erstellt, wodurch die Ereignisüberwachung von einer Anwendung schwieriger wird. Für eine einfachere Überwachung geben Sie jedem WS-Manager die gleiche Einstellung für das Attribut **PERFMEV** an.

Diese Ereignisnachricht, die jeder WS-Manager generiert, stellt die individuelle Nutzung der gemeinsam genutzten Warteschlange dar. Wenn ein WS-Manager keine Aktivität in der gemeinsam genutzten Warteschlange ausführt, sind verschiedene Werte in der Ereignisnachricht null oder null. Sie können Nullereignisnachrichten wie folgt verwenden:

- Sicherstellen, dass jeder aktive Warteschlangenmanager in einer Gruppe mit gemeinsamer Warteschlange eine Ereignisnachricht generiert
- Hervorheben von Fällen ohne Aktivität in einer gemeinsam genutzten Warteschlange für den Warteschlangenmanager, der die Ereignisnachricht erstellt hat

## Koordinationswarteschlangenmanager

Wenn ein Warteschlangenmanager ein Warteschlangenentitätsereignis absetzt, wird die Definition des gemeinsam genutzten Warteschlangenobjekts aktualisiert, um die aktiven Attribute des Leistungsereignisses zu aktivieren. Abhängig von der Definition der Warteschlangenattribute kann beispielsweise ein Ereignis "Queue Depth High" (Warteschlangenlänge hoch) ein Ereignis "Queue Depth Low" und ein Ereignis "Queue Full" (Warteschlange voll) aktiviert werden. Nachdem das gemeinsam genutzte Warteschlangenobjekt erfolgreich aktualisiert wurde, wird der Warteschlangenmanager, der das Leistungsereignis erkannt hat, zunächst zum *koordinierenden Warteschlangenmanager*.

Wenn für Leistungsereignisse aktiviert ist, führt der koordinierende Warteschlangenmanager die folgenden Aktionen aus:

1. Gibt eine Ereignisnachricht aus, die alle Leistungsdaten der gemeinsam genutzten Warteschlange erfasst, die seit dem letzten Zeitpunkt der Erstellung einer Ereignisnachricht erfasst wurden, oder

seit der letzten Zurücksetzung der Warteschlangenstatistik. Der Nachrichtendeskriptor (MQMD) dieser Nachricht enthält eine eindeutige Korrelations-ID (*CorrelId*), die vom koordinierenden WS-Manager erstellt wird.

2. Es gibt Broadcasts an alle anderen *aktiven* Warteschlangenmanager in der gleichen Gruppe mit gemeinsamer Warteschlange, um die Erzeugung einer Ereignisnachricht für die gemeinsam genutzte Warteschlange anzufordern. Der Broadcast enthält die Korrelations-ID, die vom koordinierenden Warteschlangenmanager für die Gruppe der Ereignisnachrichten erstellt wurde.

Wenn in der Gruppe mit gemeinsamer Warteschlange ein aktiver Warteschlangenmanager vorhanden ist, der für Leistungsereignisse aktiviert ist, gibt dieser aktive Warteschlangenmanager nach dem Empfang einer Anforderung vom Koordinationswarteschlangenmanager eine Ereignisnachricht für die gemeinsam genutzte Warteschlange aus. Die ausgegebene Ereignisnachricht enthält Informationen zu allen Operationen, die vom empfangenden (aktiven) Warteschlangenmanager seit dem letzten Erstellen einer Ereignisnachricht ausgeführt wurden, oder seit der letzten Zurücksetzung der Statistikdaten. Der Nachrichtendeskriptor (MQMD) dieser Ereignisnachricht enthält die eindeutige Korrelations-ID (*CorrelId*), die durch den koordinierenden Warteschlangenmanager angegeben wird.

Wenn Leistungsereignisse in einer gemeinsam genutzten Warteschlange auftreten, werden  $n$  Ereignisnachrichten erzeugt, wobei  $n$  eine Zahl zwischen 1 und der Anzahl aktiver Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange ist. Jede Ereignisnachricht enthält Daten, die sich auf die Aktivität der gemeinsamen Warteschlange für den WS-Manager beziehen, der die Ereignisnachricht generiert hat.

## Unterschiede zwischen gemeinsam genutzten und nicht gemeinsam genutzten Warteschlangen

Das Aktivieren von Warteschlangentiefenereignissen in gemeinsam genutzten Warteschlangen unterscheidet sich von der Aktivierung in nicht gemeinsam genutzten Warteschlangen. Ein wesentlicher Unterschied besteht darin, dass Ereignisse für gemeinsam genutzte Warteschlangen aktiviert werden, auch wenn **PERFMEV** auf dem Warteschlangenmanager **DISABLED** ist. Dies ist bei nicht gemeinsam genutzten Warteschlangen nicht der Fall.

Betrachten Sie das folgende Beispiel, das diesen Unterschied veranschaulicht:

- WSM1 ist ein Warteschlangenmanager mit *PerformanceEvent* (**PERFMEV** in MQSC), der auf **DISABLED** gesetzt ist.
- SQ1 ist eine gemeinsam genutzte Warteschlange mit **QSGDISP**, die in MQSC auf (SHARED) QLOCAL gesetzt ist.
- LQ1 ist eine nicht gemeinsam genutzte Warteschlange mit **QSGDISP**, die in MQSC auf (QMGR) QLOCAL gesetzt ist.

Für beide Warteschlangen sind die folgenden Attribute für ihre Definitionen festgelegt:

- **QDPHIEV (ENABLED)**
- **QDPLOEV (DISABLED)**
- **QDPMAXEV (DISABLED)**

Wenn Nachrichten in beide Warteschlangen gestellt werden, sodass die Tiefe den Schwellenwert von **QDEPTHHI** erreicht oder überschreitet, wird der Wert von **QDPHIEV** auf SQ1 auf **DISABLED** umgestellt. Außerdem werden **QDPLOEV** und **QDPMAXEV** auf **ENABLED** umgeschaltet. Die SQ1-Attribute werden automatisch für jedes Leistungsereignis zu dem Zeitpunkt, zu dem die Ereigniskriterien erfüllt sind, umgeschaltet.

Im Gegensatz dazu bleiben die Attribute für LQ1 unverändert, bis **PERFMEV** auf dem WS-Manager **ENABLED** ist. Dies bedeutet, dass, wenn z. B. das Attribut **PERFMEV** des Warteschlangenmanagers **ENABLED**, **DISABLED** und dann erneut **ENABLED** ist, die Einstellungen für das Leistungsereignis in gemeinsam genutzten Warteschlangen nicht mit denen von nicht gemeinsam genutzten Warteschlangen konsistent sind, auch wenn sie ursprünglich identisch waren.

## **Beispiele für Ereignisse der Warteschlangenlänge**

Verwenden Sie die folgenden Beispiele, um die Informationen zu verstehen, die Sie von Warteschlangentiefenereignissen erhalten können.

Das erste Beispiel bietet eine einfache Darstellung von Warteschlangentiefenereignissen. Das zweite Beispiel ist umfangreicher, aber die Prinzipien sind die gleichen wie für das erste Beispiel. In beiden Beispielen wird die gleiche Warteschlangendefinition wie folgt verwendet:

Die Warteschlange MYQUEUE1 hat eine maximale Tiefe von 1000 Nachrichten. Der Grenzwert für die hohe Warteschlangenlänge beträgt 80%, und die Tiefe der unteren Warteschlangenlänge beträgt 20%. Zunächst werden die Ereignisse in der Warteschlange für die Warteschlangenlänge aktiviert, während die anderen Ereignisse in der Warteschlangenlänge inaktiviert sind.

Die IBM MQ-Befehle (MQSC) zum Konfigurieren dieser Warteschlange sind:

```
ALTER QMGR PERFMV(ENABLED)

DEFINE QLOCAL('MYQUEUE1') MAXDEPTH(1000) QDPMAXEV(DISABLED) QDEPTHHI(80)
QDPHIEV(ENABLED) QDEPTHLO(20) QDPLOEV(DISABLED)
```

### **Zugehörige Konzepte**

[„Ereignisse der Warteschlangenlänge“ auf Seite 32](#)

Ereignisse der Warteschlangenlänge beziehen sich auf die Warteschlangenlänge, d. h. die Anzahl der Nachrichten in der Warteschlange.

### **Zugehörige Tasks**

[„Warteschlangenlänge aktivieren“ auf Seite 33](#)

Zum Konfigurieren einer Warteschlange für beliebige Warteschlangenlänge-Ereignisse legen Sie die entsprechenden Warteschlangenmanager- und Warteschlangenattribute fest.

### **Zugehörige Verweise**

[Die MQSC-Befehle](#)

*Warteschlangenlänge-Ereignisse: Beispiel 1*

Eine grundlegende Folge von Warteschlangentiefenereignissen.

[Abbildung 7 auf Seite 38](#) zeigt die Variation der Warteschlangenlänge im Verlauf der Zeit an.

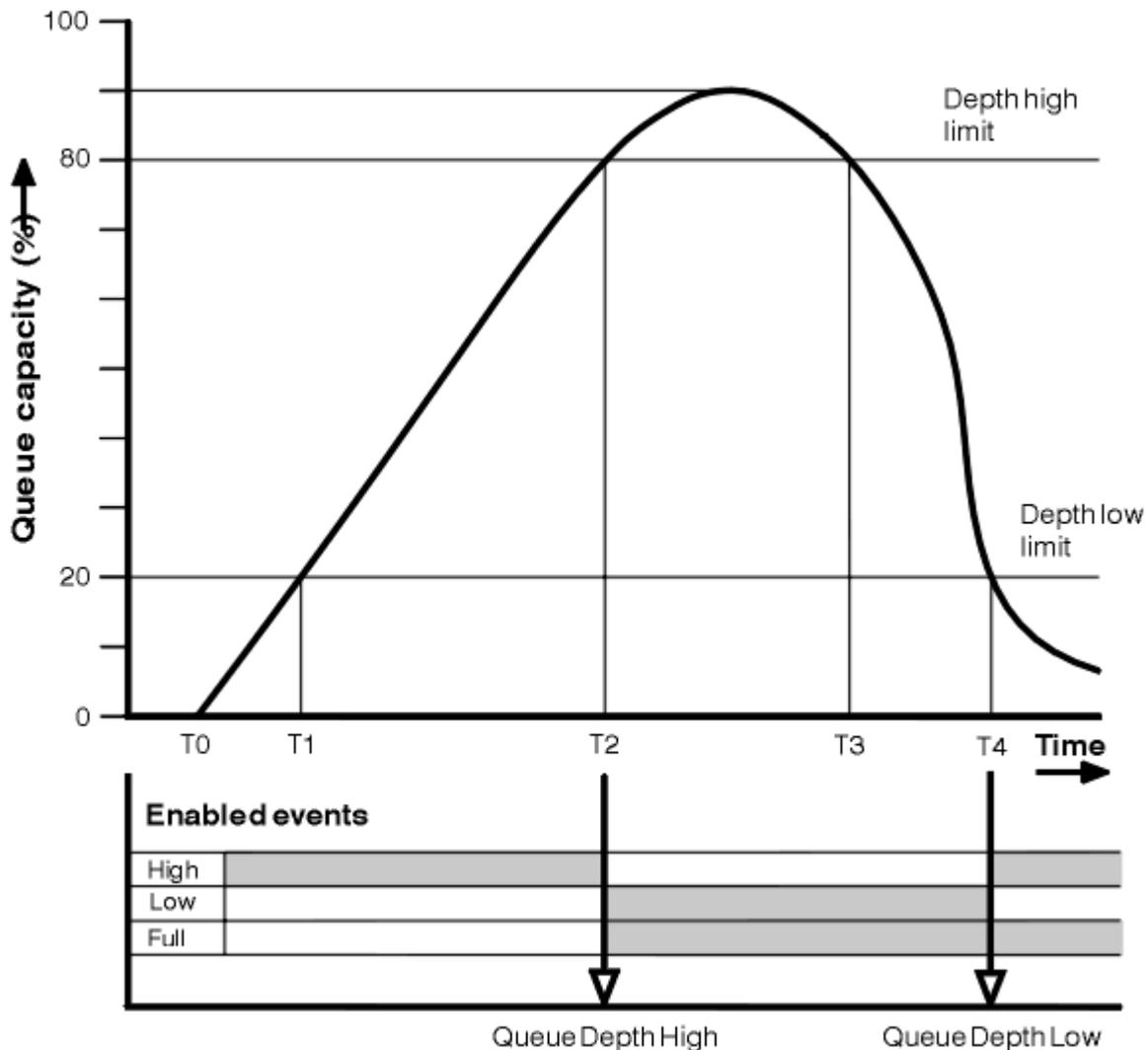


Abbildung 7. Ereignisse der Warteschlangenlänge (1)

### Commentary

1. Bei T (1) erhöht sich die Warteschlangenlänge (mehr MQPUT-Aufrufe als MQGET-Aufrufe) und überschneidet die Warteschlangenlänge (Niedrig). Es wird zu diesem Zeitpunkt kein Ereignis generiert.
2. Die Warteschlangenlänge wird weiter bis T (2) erhöht, wenn die tiefe Obergrenze (80%) erreicht ist und ein Ereignis vom Typ "Queue Depth High" (Warteschlangenlänge hoch) generiert wird.  
Auf diese Weise werden sowohl die Warteschlangenlänge als auch die Warteschlangenlänge niedrig.
3. Durch die (mutmaßlichen) vorbeugenden Maßnahmen, die durch das Ereignis eingeleitet werden, wird verhindert, dass die Warteschlange voll wird. Mit der Zeit T (3) wurde die Warteschlangenlänge High-Limit wieder erreicht, diesmal von oben. Es wird zu diesem Zeitpunkt kein Ereignis generiert.
4. Die Warteschlangenlänge bleibt bis zum Zeitpunkt T (4), wenn sie die Tiefgrenze (20%) erreicht, und es wird ein Ereignis "Queue Depth Low" (Warteschlangenlänge niedrig) generiert.

Dadurch wird sowohl die Warteschlangenlänge als auch die Warteschlangenlänge hoch.

### Zusammenfassung der Ereignisstatistik

In [Tabelle 8](#) auf Seite 39 sind die Warteschlangenereignisstatistik zusammengefasst, und [Tabelle 9](#) auf Seite 39 fasst zusammen, welche Ereignisse aktiviert sind.

Tabelle 8. Ereignisstatistikzusammenfassung für Warteschlangentiefenereignisse (Beispiel 1)

Ereignisstatistik	Ereignis 2	Ereignis 4
Zeitpunkt des Ereignisses	T (2)	T (4)
Typ des Ereignisses	Hohe Warteschlangenlänge	Niedrige Warteschlangenlänge
Zurücksetzungszeit (TimeS)	T (2)-T (0)	T (4)-T (2)
HighQDepth (Maximale Warteschlangenlänge seit Zurücksetzen)	800	900
MsgEnqCount	1157	1220
MsgDeqCount	357	1820

Tabelle 9. Zusammenfassung, die anzeigt, welche Ereignisse aktiviert sind

Zeitperiode	Ereignis "Queue Depth High"	Ereignis "Queue Depth Low"	Ereignis 'Warteschlange voll'
Vor T (1)	ENABLED	-	-
T (1) bis T (2)	ENABLED	-	-
T (2) bis T (3)	-	ENABLED	ENABLED
T (3) bis T (4)	-	ENABLED	ENABLED
Nach T (4)	ENABLED	-	ENABLED

*Warteschlangenlänge-Ereignisse: Beispiel 2*

Eine umfangreichere Abfolge von Warteschlangentiefenereignissen.

Abbildung 8 auf Seite 40 zeigt die Variation der Warteschlangenlänge im Verlauf der Zeit an.



## Zusammenfassung der Ereignisstatistik

Tabelle 10 auf Seite 41 fasst die Warteschlangenereignisstatistik zusammen und Tabelle 11 auf Seite 41 fasst zusammen, welche Ereignisse für dieses Beispiel zu unterschiedlichen Zeiten aktiviert sind.

*Tabelle 10. Ereignisstatistikzusammenfassung für Warteschlangentiefenereignisse (Beispiel 2)*

Ereignisstatistik	Ereignis 4	Ereignis 6	Ereignis 8	Ereignis 9	Ereignis 12
Zeitpunkt des Ereignisses	T (4)	T (6)	T (8)	T (9)	T (12)
Typ des Ereignisses	Hohe Warteschlangenlänge	Niedrige Warteschlangenlänge	Hohe Warteschlangenlänge	Warteschlange voll	Niedrige Warteschlangenlänge
Zurücksetzungszeit (TimeS)	T (4)-T (0)	T (6)-T (4)	T (8)-T (6)	T (9)-T (8)	T (12)-T (9)
Höchste QDepth	800	855	800	1000	1000
MsgEnqCount	1645	311	1377	324	221
MsgDeqCount	845	911	777	124	1021

*Tabelle 11. Zusammenfassung, die anzeigt, welche Ereignisse aktiviert sind*

Zeitperiode	Ereignis "Queue Depth High"	Ereignis "Queue Depth Low"	Ereignis 'Warteschlange voll'
T (0) bis T (4)	ENABLED	-	-
T (4) bis T (6)	-	ENABLED	ENABLED
T (6) bis T (8)	ENABLED	-	ENABLED
T (8) bis T (9)	-	ENABLED	ENABLED
T (9) bis T (12)	-	ENABLED	-
Nach T (12)	ENABLED	-	ENABLED

**Anmerkung:** Ereignisse sind nicht syncpoint. Daher können Sie eine leere Warteschlange haben, dann die Warteschlange für ein Ereignis ausfüllen und dann alle Nachrichten unter der Steuerung eines Synchronisationspunktmanagers rückgängig machen. Das Ereignis, das aktiviert wurde, wurde jedoch automatisch gesetzt, sodass beim nächsten Ausfüllen der Warteschlange kein Ereignis generiert wird.

## Konfigurationsereignisse

Konfigurationsereignisse sind Benachrichtigungen, die generiert werden, wenn ein Objekt erstellt, geändert oder gelöscht wird, und können auch durch explizite Anforderungen generiert werden.

Informationen zum Aktivieren von Ereignissen finden Sie unter „[Steuerung von Konfigurations-, Befehls- und Protokollfunktionsereignissen](#)“ auf Seite 18.

Konfigurationsereignisse benachrichtigen Sie über Änderungen an den Attributen eines Objekts. Es gibt vier Typen von Konfigurationsereignissen:

- Objektereignisse erstellen
- Objektereignisse ändern
- Objektereignisse löschen
- Objektereignisse aktualisieren

Die Ereignisdaten enthalten die folgenden Informationen:

### **Ursprungsinformationen**

enthält den Warteschlangenmanager, von dem aus die Änderung vorgenommen wurde, die ID des Benutzers, der die Änderung vorgenommen hat, und die Art und Weise, in der die Änderung vorgenommen wurde, z. B. durch einen Konsolbefehl.

### **Kontextinformationen**

Ein Replikat der Kontextinformationen in den Nachrichtendaten aus der Befehlsnachricht.

Kontextinformationen werden nur dann in die Ereignisdaten aufgenommen, wenn der Befehl als Nachricht in der Warteschlange SYSTEM.COMMAND.INPUT eingegeben wurde.

### **Objektidentität**

umfasst den Namen, die Art und die Disposition des Objekts.

### **Objektattribute**

enthält die Werte aller Attribute im Objekt.

Das Ereignis ist eine Nachricht im PCF-Format. Weitere Informationen finden Sie unter:

- [Objekt ändern](#)
- [Objekt erstellen](#)
- [Objekt löschen](#)
- [Objekt aktualisieren](#)

Im Falle von Änderungsobjekt ereignissen wird eine Nachricht vor der Änderung und eine Nachricht nach der Änderung erzeugt. Die folgenden Felder sind dabei identisch, so dass Sie übereinstimmende Nachrichten identifizieren können:

- Die gleiche Korrelations-ID
- Im PCF-Header:
  - Die Nachricht vor der Änderung hat den Header `MsgSeqNumber': 1, 'Control': 'NOT_LAST'`
  - Die Nachricht nach der Änderung hat den Header `MsgSeqNumber': 2, 'Control': 'LAST'`

Jede generierte Konfigurationsereignisnachricht wird in die Warteschlange SYSTEM.ADMIN.CONFIG.EVENT gestellt.

### **Zugehörige Konzepte**

„Konfigurationsereignisse“ auf Seite 13

Konfigurationsereignisse werden generiert, wenn ein Konfigurationsereignis explizit angefordert wird, oder automatisch, wenn ein Objekt erstellt, geändert oder gelöscht wird.

### **Zugehörige Verweise**

[Objekt erstellen](#)

[Objekt ändern](#)

[Objekt löschen](#)

[Objekt aktualisieren](#)

„Ereignistypen“ auf Seite 8

Verwenden Sie diese Seite, um die Typen von Instrumentierungsereignis anzuzeigen, die ein Warteschlangenmanager oder eine Kanalinstanz melden kann.

### **Generierung des Konfigurationsereignis**

Auf dieser Seite können Sie die Befehle anzeigen, mit denen Konfigurationsereignisse generiert werden, und die Umstände zu verstehen, unter denen Konfigurationsereignisse nicht generiert werden.

Eine Konfigurationsereignisnachricht wird in die Konfigurationsereigniswarteschlange gestellt, wenn das Attribut CONFIGEV queue manager ENABLED ist und

- werden die folgenden Befehle oder deren äquivalente PCF-Befehle ausgegeben:
  - DELETE AUTHINFO
  - DELETE CFSTRUCT

- DELETE CHANNEL
- DELETE NAMELIST
- DELETE PROCESS
- DELETE QMODEL/QALIAS/QREMOTE
- DELETE STGCLASS
- DELETE TOPIC
- REFRESH QMGR
- jeder der folgenden Befehle oder die entsprechenden PCF-Äquivalente werden auch dann ausgegeben, wenn das Objekt nicht geändert wird:
  - DEFINE/ALTER AUTHINFO
  - DEFINE/ALTER CFSTRUCT
  - DEFINE/ALTER CHANNEL
  - DEFINE/ALTER NAMELIST
  - DEFINE/ALTER PROCESS
  - DEFINE/ALTER QMODEL/QALIAS/QREMOTE
  - DEFINE/ALTER STGCLASS
  - DEFINE/ALTER TOPIC
  - DEFINE MAXSMSGS
  - SET CHLAUTH
  - ALTER QMGR, es sei denn, das Attribut CONFIGEV ist DISABLED und wird nicht in ENABLED geändert.
- jeder der folgenden Befehle oder die entsprechende PCF-Entsprechung wird für eine lokale Warteschlange ausgegeben, die nicht temporär dynamisch ist, selbst wenn keine Änderung an der Warteschlange erfolgt.
  - DELETE QLOCAL
  - DEFINE/ALTER QLOCAL
- Es wird ein MQSET-Aufruf ausgegeben, der nicht für eine temporäre dynamische Warteschlange gilt, auch wenn keine Änderungen am Objekt vorhanden sind.

### **Wenn Konfigurationsereignisse nicht generiert werden**

Konfigurationsereignisnachrichten werden unter den folgenden Umständen nicht generiert:

- Wenn ein Befehl oder ein MQSET-Aufruf fehlschlägt
- Wenn ein Warteschlangenmanager bei dem Versuch, ein Konfigurationsereignis in die Ereigniswarteschlange zu stellen, einen Fehler feststellt, wird der Befehl oder der MQSET-Aufruf abgeschlossen, es wird jedoch keine Ereignisnachricht generiert.
- Für eine temporäre dynamische Warteschlange
- Wenn interne Änderungen am Attribut TRIGGER queue vorgenommen werden,
- Für die Konfigurationsereigniswarteschlange SYSTEM.ADMIN.CONFIG.EVENT, mit Ausnahme des Befehls REFRESH QMGR
- Für Befehle REFRESH/RESET CLUSTER und RESUME/SUSPEND QMGR, die Clustering-Änderungen bewirken
- Beim Erstellen oder Löschen eines Warteschlangenmanagers beim Erstellen oder Löschen

### **Zugehörige Konzepte**

„Konfigurationsereignisse“ auf Seite 41

Konfigurationsereignisse sind Benachrichtigungen, die generiert werden, wenn ein Objekt erstellt, geändert oder gelöscht wird, und können auch durch explizite Anforderungen generiert werden.

## Zugehörige Verweise

[Die MQSC-Befehle](#)

[Einführung in Programmierbare Befehlsformate](#)

[MQSET - Objektattribute festlegen](#)

[MQSET - Objektattribute festlegen](#)

## Konfigurationsereignisverwendung

Verwenden Sie diese Seite, um zu sehen, wie Sie Konfigurationsereignisse verwenden können, um Informationen zu Ihrem System zu erhalten und um die Faktoren, wie z. B. CMDSCOPE, zu verstehen, die sich auf die Verwendung von Konfigurationsereignissen auswirken können.

Sie können Konfigurationsereignisse für die folgenden Zwecke verwenden:

1. Erstellung und Verwaltung eines zentralen Konfigurationsrepositorys, aus dem Berichte erstellt werden können und Informationen über die Struktur des Systems generiert werden können.
2. So generieren Sie einen Prüfprotokoll. Wenn z. B. ein Objekt unerwartet geändert wird, können Informationen darüber gespeichert werden, wer die Änderung vorgenommen hat und wann diese ausgeführt wurde.

Dies kann besonders nützlich sein, wenn Befehlsereignisse ebenfalls aktiviert sind. Wenn ein MQSC- oder PCF-Befehl bewirkt, dass ein Konfigurationsereignis und ein Befehlsereignis generiert werden, verwenden beide Ereignisnachrichten die gleiche Korrelations-ID in ihrem Nachrichtendeskriptor.

Für einen MQSET-Aufruf oder einen der folgenden Befehle:

- DEFINE, Objekt
- ALTER, Objekt
- Objekt DELETE

Wenn das WS-Manager-Attribut CONFIGEV aktiviert ist, die Konfigurationsereignisnachricht jedoch nicht in die Konfigurationsereigniswarteschlange gestellt werden kann, z. B. die Ereigniswarteschlange nicht definiert wurde, wird der Befehl oder der MQSET-Aufruf unabhängig ausgeführt.

## Auswirkungen von CMDSCOPE

Für Befehle, bei denen CMDSCOPE verwendet wird, werden die Konfigurationsereignisnachricht oder -nachrichten auf dem Warteschlangenmanager oder den Warteschlangenmanagern generiert, auf denen der Befehl ausgeführt wird, und nicht, wo der Befehl eingegeben wird. Alle Quellen- und Kontextinformationen in den Ereignisdaten beziehen sich jedoch auf den ursprünglichen Befehl wie eingegeben, selbst wenn der Befehl mit CMDSCOPE eine Datei ist, die vom Quellenwarteschlangenmanager generiert wurde.

Wenn eine Gruppe mit gemeinsamer Warteschlange Warteschlangenmanager enthält, die sich nicht in der aktuellen Version befinden, werden Ereignisse für jeden Befehl generiert, der über CMDSCOPE auf einem Warteschlangenmanager ausgeführt wird, der sich in der aktuellen Version befindet, jedoch nicht in den Versionen, die sich in einer früheren Version befinden. Dies geschieht auch dann, wenn der Warteschlangenmanager, in dem der Befehl eingegeben wird, die vorherige Version hat, obwohl in einem solchen Fall keine Kontextinformationen in den Ereignisdaten enthalten sind.

## Zugehörige Konzepte

[Einführung in Programmierbare Befehlsformate](#)

„Konfigurationsereignisse“ auf Seite 41

Konfigurationsereignisse sind Benachrichtigungen, die generiert werden, wenn ein Objekt erstellt, geändert oder gelöscht wird, und können auch durch explizite Anforderungen generiert werden.

## Zugehörige Verweise

[MQSET - Objektattribute festlegen](#)

[MQSET - Objektattribute festlegen](#)

## **Ereigniskonfigurationsereignis aktualisieren**

Das Konfigurationsereignis für die Aktualisierungsobjektkonfiguration unterscheidet sich von den anderen Konfigurationsereignissen, da es nur dann auftritt, wenn es explizit angefordert wird.

Die Erstellungs-, Änderungs- und Löscheignisse werden durch einen MQSET-Aufruf oder durch einen Befehl zum Ändern eines Objekts generiert, das Aktualisierungsobjekt ereignis tritt jedoch nur auf, wenn dies explizit vom MQSC-Befehl, REFRESH QMGR oder dessen PCF-Äquivalent angefordert wird.

Der Befehl REFRESH QMGR unterscheidet sich von allen anderen Befehlen, die Konfigurationsereignisse generieren. Alle anderen Befehle gelten für ein bestimmtes Objekt und generieren ein einzelnes Konfigurationsereignis für dieses Objekt. Der Befehl REFRESH QMGR kann viele Konfigurationsereignisnachrichten erzeugen, die möglicherweise jede Objektdefinition darstellen, die von einem Warteschlangenmanager gespeichert wird. Für jedes Objekt, das ausgewählt ist, wird eine Ereignisnachricht generiert.

Der Befehl REFRESH QMGR verwendet eine Kombination aus drei Auswahlkriterien, um die Anzahl der betroffenen Objekte zu filtern:

- Objektname
- Objekttyp
- Aktualisierungsintervall

Wenn Sie keines der Auswahlkriterien im Befehl REFRESH QMGR angeben, werden die Standardwerte für jedes Auswahlkriterium verwendet, und für jede Objektdefinition, die vom WS-Manager gespeichert wird, wird eine Ereignisnachricht für die Aktualisierungskonfiguration generiert. Dies kann zu unzulässigen Verarbeitungszeiten und zur Generierung von Ereignisnachrichten führen. Ziehen Sie die Angabe einiger Auswahlkriterien in Betracht.

Der Befehl REFRESH QMGR, der die Aktualisierungsereignisse generiert, kann in den folgenden Situationen verwendet werden:

- Wenn Konfigurationsdaten für alle oder einige der Objekte in einem System gesucht werden, unabhängig davon, ob die Objekte kürzlich bearbeitet wurden, z. B. wenn Konfigurationsereignisse zuerst aktiviert werden.

Ziehen Sie die Verwendung mehrerer Befehle mit jeweils einer anderen Auswahl von Objekten in Betracht, die jedoch alle eingeschlossen sind.

- Wenn in der Warteschlange SYSTEM.ADMIN.CONFIG.EVENT ein Fehler aufgetreten ist. In diesem Fall werden keine Konfigurationsereignisnachrichten für Ereignisse des Erstellungs-, Änderungs- oder Löschvorgangs generiert. Wenn der Fehler in der Warteschlange korrigiert wurde, kann der Befehl zum Aktualisieren des Warteschlangenmanagers verwendet werden, um die Generierung von Ereignisnachrichten anzufordern, die verloren gingen, während ein Fehler in der Warteschlange aufgetreten ist. In dieser Situation können Sie das Aktualisierungsintervall auf die Zeit setzen, für die die Warteschlange nicht verfügbar war.

### **Zugehörige Konzepte**

[„Konfigurationsereignisse“ auf Seite 41](#)

Konfigurationsereignisse sind Benachrichtigungen, die generiert werden, wenn ein Objekt erstellt, geändert oder gelöscht wird, und können auch durch explizite Anforderungen generiert werden.

### **Zugehörige Verweise**

[REFRESH QMGR](#)

[WS-Manager aktualisieren](#)

## **Befehlsereignisse**

Befehlsereignisse sind Benachrichtigungen, die ein MQSC- oder PCF-Befehl erfolgreich ausgeführt hat.

Die Ereignisdaten enthalten die folgenden Informationen:

### **Ursprungsinformationen**

umfasst den Warteschlangenmanager, von dem aus der Befehl abgesetzt wurde, die ID des Benutzers, der den Befehl abgesetzt hat, und die Art und Weise, in der der Befehl ausgegeben wurde, z. B. durch einen Konsolbefehl.

### **Kontextinformationen**

Ein Replikat der Kontextinformationen in den Nachrichtendaten aus der Befehlsnachricht. Wenn ein Befehl nicht über eine Nachricht eingegeben wird, werden die Kontextinformationen weggelassen.

Kontextinformationen werden nur dann in die Ereignisdaten aufgenommen, wenn der Befehl als Nachricht in der Warteschlange SYSTEM.COMMAND.INPUT eingegeben wurde.

### **Befehlsinformationen**

Der Typ des Befehls, der ausgegeben wurde.

### **Befehlsdaten**

- für PCF-Befehle, ein Replikat der Befehlsdaten
- für MQSC-Befehle, den Befehlstext

Das Befehlsdatenformat stimmt nicht unbedingt mit dem Format des ursprünglichen Befehls überein. Unter Multiplatforms haben die Befehlsdaten beispielsweise immer ein PCF-Format, selbst wenn es sich bei der ursprünglichen Anforderung um einen MQSC-Befehl handelt.

Jede Befehlsereignisnachricht, die generiert wird, wird in die Befehlsereigniswarteschlange SYSTEM.ADMIN.COMMAND.EVENT gestellt.

### **Zugehörige Verweise**

Befehl

„Ereignistypen“ auf Seite 8

Verwenden Sie diese Seite, um die Typen von Instrumentierungsereignis anzuzeigen, die ein Warteschlangenmanager oder eine Kanalinstanz melden kann.

### ***Befehlsereignisgenerierung***

Verwenden Sie diese Seite, um die Situationen anzuzeigen, die zum Erstellen von Befehlsereignissen führen, und die Umstände zu verstehen, unter denen Befehlsereignisse nicht generiert werden.

In den folgenden Situationen wird eine Befehlsereignisnachricht generiert:

- Wenn das WS-Managerattribut CMDEV als ENABLED angegeben wird und ein MQSC-oder PCF-Befehl erfolgreich ausgeführt wird.
- Wenn das Attribut CMDEV-WS-Manager als NODISPLAY angegeben ist und jeder Befehl erfolgreich ausgeführt wird, mit Ausnahme von DISPLAY-Befehlen (MQSC) und Inquire-Befehlen (PCF).
- Wenn Sie den MQSC-Befehl, den Befehl ALTER QMGR oder den PCF-Befehl, den WS-Manager ändern, und das WS-Manager-Attribut CMDEV einer der folgenden Bedingungen ausführen:
  - CMDEV wird nach der Änderung nicht als DISABLED angegeben
  - CMDEV wurde vor der Änderung nicht als DISABLED angegeben.

Wenn ein Befehl für die Befehlsereigniswarteschlange SYSTEM.ADMIN.COMMAND.EVENT ausgeführt wird, wird ein Befehlsereignis generiert, wenn die Warteschlange noch vorhanden ist und sie nicht gesperrt ist.

### **Wenn Befehlsereignisse nicht generiert werden**

Eine Befehlsereignisnachricht wird unter den folgenden Umständen nicht generiert:

- Wenn ein Befehl fehlschlägt
- Wenn ein Warteschlangenmanager bei dem Versuch, ein Befehlsereignis in die Ereigniswarteschlange zu stellen, einen Fehler feststellt, wird der Befehl unabhängig davon ausgeführt, aber es wird keine Ereignisnachricht generiert.
- Für den MQSC-Befehl REFRESH QMGR TYPE (EARLY)

- Für den MQSC-Befehl START QMGR MQSC
- Für den MQSC-Befehl SUSPEND QMGR, wenn der Parameter LOG angegeben ist.
- Für den MQSC-Befehl RESUME QMGR, wenn der Parameter LOG angegeben ist.

### Zugehörige Konzepte

„Befehlsereignisse“ auf Seite 45

Befehlsereignisse sind Benachrichtigungen, die ein MQSC-oder PCF-Befehl erfolgreich ausgeführt hat.

### Zugehörige Verweise

[REFRESH QMGR](#)

[START QMGR](#)

[SUSPEND QMGR](#)

[RESUME QMGR](#)

[SUSPEND QMGR, RESUME QMGR und Cluster](#)

### Befehlsereignisverwendung

Auf dieser Seite können Sie anzeigen, wie Sie Befehlsereignisse verwenden können, um einen Prüfprotokoll für die Befehle zu generieren, die ausgeführt wurden.

Wenn z. B. ein Objekt unerwartet geändert wird, können Informationen darüber gespeichert werden, wer die Änderung vorgenommen hat und wann diese ausgeführt wurde. Dies kann besonders nützlich sein, wenn Konfigurationsereignisse ebenfalls aktiviert sind. Wenn ein MQSC-oder PCF-Befehl bewirkt, dass ein Befehlsereignis und ein Konfigurationsereignis generiert werden, verwenden beide Ereignisnachrichten die gleiche Korrelations-ID in ihrem Nachrichtendeskriptor.

Wenn eine Befehlsereignisnachricht generiert wird, aber nicht in die Befehlsereigniswarteschlange gestellt werden kann, z. B. wenn die Befehlsereigniswarteschlange nicht definiert wurde, wird der Befehl, für den das Befehlsereignis generiert wurde, trotzdem ausgeführt, unabhängig davon, ob es sich um eine Ereigniswarteschlange handelt.

### Auswirkungen von CMDSCOPE

Für Befehle, bei denen CMDSCOPE verwendet wird, werden die Befehlsereignisnachricht oder -nachrichten auf dem Warteschlangenmanager oder den Warteschlangenmanagern generiert, auf denen der Befehl ausgeführt wird, und nicht, wo der Befehl eingegeben wird. Alle Quellen- und Kontextinformationen in den Ereignisdaten beziehen sich jedoch auf den ursprünglichen Befehl wie eingegeben, selbst wenn der Befehl mit CMDSCOPE eine Datei ist, die vom Quellenwarteschlangenmanager generiert wurde.

### Zugehörige Konzepte

„Befehlsereignisse“ auf Seite 45

Befehlsereignisse sind Benachrichtigungen, die ein MQSC-oder PCF-Befehl erfolgreich ausgeführt hat.

„Befehlsereignisgenerierung“ auf Seite 46

Verwenden Sie diese Seite, um die Situationen anzuzeigen, die zum Erstellen von Befehlsereignissen führen, und die Umstände zu verstehen, unter denen Befehlsereignisse nicht generiert werden.

### Zugehörige Verweise

[Die MQSC-Befehle](#)

[PCF-Befehle und -Antworten in Gruppen](#)

## Ereignisse der Protokollfunktion

Bei Protokollierungsereignissen handelt es sich um Benachrichtigungen, die ein Warteschlangenmanager in einen neuen Protokollbereich  oder unter IBM i in einen Journalempfänger geschrieben hat.  Protokollierungsereignisnachrichten sind mit IBM MQ for z/OS nicht verfügbar.

Die Ereignisdaten enthalten die folgenden Informationen:

- Der Name des aktuellen Protokollspeicherbereichs.

- Der Name des frühesten Protokollspeicherbereichs, der für die Neustartwiederherstellung benötigt wird.
- Der Name des frühesten Protokollspeicherbereichs, der für die Datenträgerwiederherstellung benötigt wird.
- Das Verzeichnis, in dem sich die Protokollextents befinden.
- **V 9.1.0** Der Name des frühesten Protokollspeicherbereichs, für den eine Archivierungsbenachrichtigung erforderlich ist.

Ab IBM MQ 9.1.0 wird ein Protokollierungsereignis für die Verwaltung des Archivprotokolls generiert, also wenn **ARCHLOG** geändert und der Wert **ARCHLOG** in die Protokollierungsereignisnachricht aufgenommen wird.

Weitere Informationen zu allen diesen Parametern finden Sie in [DISPLAY QMSTATUS](#).

Sie können die Werte **CURRLOG** und **ARCHLOG** verwenden, um festzulegen, was archiviert werden soll. Wenn sich **CURRLOG** ändert, können Dateien mit Nummern kleiner als **CURRLOG** zur Archivierung gesendet werden. Sobald die Archivierung für den Bereich abgeschlossen ist, sollten Sie [SET LOG](#) aufrufen, um den Warteschlangenmanager zu benachrichtigen.

**ARCHLOG** ist der älteste Bereich, der archiviert werden muss. Wenn Sie diesen Speicherbereich archiviert haben und [SET LOG](#) aufgerufen haben, um den Warteschlangenmanager zu benachrichtigen, dass er archiviert wurde, verschiebt der Warteschlangenmanager **ARCHLOG** in den nächsten Speicherbereich in Folge. Der Warteschlangenmanager terminiert Speicherbereiche, die älter als **ARCHLOG** sind, um gelöscht oder wiederverwendet zu werden.

Nachdem **SET LOG** für diesen Speicherbereich aufgerufen wurde, wird ein neues Ereignis ausgegeben, da sich **ARCHLOG** geändert hat, und Sie müssen wissen, was nach dem neuen Ereignis archiviert werden muss.

Wenn aus irgendeinem Grund Ihr Archivierungsprozess fehlschlägt und eine große Anzahl von Benachrichtigungen auftritt, kann Ihr Administrator manuell den Befehl [RESET QMGR TYPE \(ARCHLOG\)](#) ausgeben. Dadurch wird der Warteschlangenmanager darüber benachrichtigt, dass er alle Speicherbereiche, die älter als und einschließlich des von Ihnen angegebenen Speicherbereichs sind, wiederverwenden oder löschen kann.

Jede Ereignisnachricht der Protokollfunktion, die generiert wird, wird in die Ereigniswarteschlange der Protokollfunktion (SYSTEM.ADMIN.LOGGER.EVENT) gestellt.

### Zugehörige Verweise

[Logger](#)

„Ereignistypen“ auf Seite 8

Verwenden Sie diese Seite, um die Typen von Instrumentierungsereignis anzuzeigen, die ein Warteschlangenmanager oder eine Kanalinstanz melden kann.

### **Multi** Generierung von Protokollfunktionsereignis

Verwenden Sie diese Seite, um die Situationen anzuzeigen, die dazu führen, dass Protokollfunktionsereignisse generiert werden, und die Umstände zu verstehen, unter denen Protokollfunktionsereignisse nicht generiert werden.

In den folgenden Situationen wird eine Protokollfunktionsereignisnachricht generiert:

- Wenn das Warteschlangenmanagerattribut LOGGEREV als ENABLED angegeben ist und der Warteschlangenmanager mit dem Schreiben in einen neuen Protokollspeicherbereich bzw. unter IBM i in einen Journaempfänger beginnt.
- Wenn das WS-Managerattribut LOGGEREV als ENABLED angegeben wird und der WS-Manager gestartet wird.
- Wenn das WS-Managerattribut LOGGEREV von DISABLED in ENABLED geändert wird.

- **V 9.1.0** Wenn das WS-Manager-Attribut LOGGEREV benachrichtigt wird, dass der Name des ältesten Protokollspeicherbereichs, für den der Warteschlangenmanager auf die Archivierung wartet (ARCHLOG), archiviert wurde, wird er archiviert.

**Tipp:** Sie können mit dem MQSC-Befehl RESET QMGR einen Warteschlangenmanager anfordern, um das Schreiben in einen neuen Protokollspeicherbereich zu starten.

## Wenn Protokollfunktionsereignisse nicht generiert werden

Eine Ereignisnachricht der Protokollfunktion wird unter den folgenden Umständen nicht generiert:

- Wenn ein WS-Manager für die Verwendung der Umlaufprotokollierung konfiguriert ist.  
In diesem Fall wird das WS-Managerattribut LOGGEREV auf DISABLED gesetzt und kann nicht geändert werden.
- Wenn ein Warteschlangenmanager einen Fehler feststellt, der versucht, ein Protokollfunktionsereignis in die Ereigniswarteschlange zu stellen, wird die Aktion, die das Ereignis ausgelöst hat, abgeschlossen, es wird jedoch keine Ereignisnachricht generiert.

### Zugehörige Konzepte

„Ereignisse der Protokollfunktion“ auf Seite 47

Bei Protokollierungsereignissen handelt es sich um Benachrichtigungen, die ein Warteschlangenmanager in einen neuen Protokollbereich **IBM i** oder unter IBM i in einen Journalempfänger geschrieben hat. **z/OS** Protokollierungsereignisnachrichten sind mit IBM MQ for z/OS nicht verfügbar.

### Zugehörige Verweise

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(10-stellige Ganzzahl mit Vorzeichen\)](#)

[RESET QMGR](#)

## **Multi** **Logger-Ereignisverwendung**

Auf dieser Seite können Sie anzeigen, wie Sie Protokollerweiterungen verwenden können, um die Protokoll extents zu ermitteln, die für den Neustart des Warteschlangenmanagers oder die Datenträgerwiederherstellung nicht mehr benötigt werden.

Sie können überflüssige Protokoll extents auf einem Datenträger, wie z. B. Band für die Wiederherstellung nach einem Katastrophenfall, archivieren, bevor Sie sie aus dem Verzeichnis für aktive Protokolldateien entfernen. Durch die regelmäßige Entfernung überflüssiger Protokoll extents wird die Plattenspeicherplatzbelegung auf ein Minimum gehalten.

Wenn das WS-Manager-Attribut LOGGEREV aktiviert ist, aber keine Protokollfunktionsereignisnachricht in die Ereigniswarteschlange der Protokollfunktion gestellt werden kann, weil beispielsweise die Ereigniswarteschlange nicht definiert wurde, wird die Aktion, die das Ereignis ausgelöst hat, unabhängig von der Ereigniswarteschlange fortgesetzt.

### Zugehörige Konzepte

„Ereignisse der Protokollfunktion“ auf Seite 47

Bei Protokollierungsereignissen handelt es sich um Benachrichtigungen, die ein Warteschlangenmanager in einen neuen Protokollbereich **IBM i** oder unter IBM i in einen Journalempfänger geschrieben hat. **z/OS** Protokollierungsereignisnachrichten sind mit IBM MQ for z/OS nicht verfügbar.

### Zugehörige Verweise

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(10-stellige Ganzzahl mit Vorzeichen\)](#)

„Generierung von Protokollfunktionsereignis“ auf Seite 48

Verwenden Sie diese Seite, um die Situationen anzuzeigen, die dazu führen, dass Protokollfunktionsereignisse generiert werden, und die Umstände zu verstehen, unter denen Protokollfunktionsereignisse nicht generiert werden.

## Beispielprogramm C zur Überwachung der Ereigniswarteschlange der Protokollfunktion

Verwenden Sie diese Seite, um ein Beispiel-C-Programm anzuzeigen, das die Ereigniswarteschlange der Protokollfunktion für neue Ereignisnachrichten überwacht, diese Nachrichten liest und den Inhalt der Nachricht in stdout einreicht.

```

/*****
/*
/* Program name: AMQSLOG0.C
/*
/* Description: Sample C program to monitor the logger event queue and
/* display formatted message content to stdout when a logger
/* event occurs
/*
/* <copyright
/* notice="lm-source-program"
/* pids="5724-H72,"
/* years="2005, 2024"
/* crc="186943832" >
/* Licensed Materials - Property of IBM
/*
/* 5724-H72,
/*
/* (C) Copyright IBM Corp. 2005, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* </copyright>
*****/
/*
/* Function: AMQSLOG is a sample program which monitors the logger event
/* queue for new event messages, reads those messages, and displays the
/* formatted contents of the message to stdout.
/*
*****/
/*
/* AMQSLOG has 1 parameter - the queue manager name (optional, if not
/* specified then the default queue manager is implied)
/*
*****/

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

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

/*****
/* Constants
*****/

#define MAX_MESSAGE_LENGTH 8000

typedef struct _ParmTableEntry
{
    MQLONG ConstVal;
    PMQCHAR Desc;
} ParmTableEntry;

ParmTableEntry ParmTable[] =
{
    {0, ""},
    {MQCA_Q_MGR_NAME, "Queue Manager Name"},
    {MQCMD_LOGGER_EVENT, "Logger Event Command"},
    {MQRC_LOGGER_STATUS, "Logger Status"},
    {MQCACF_ARCHIVE_LOG_EXTENT_NAME, "Archive Log Extent"},
    {MQCACF_CURRENT_LOG_EXTENT_NAME, "Current Log Extent"},
    {MQCACF_RESTART_LOG_EXTENT_NAME, "Restart Log Extent"},
    {MQCACF_MEDIA_LOG_EXTENT_NAME, "Media Log Extent"},
    {MQCACF_LOG_PATH, "Log Path"}
};

#if defined(MQ_64_BIT)

```

```

#define Int32
#else
#define Int32 "l"
#endif

/*****
/* Function prototypes */
*****/

static void ProcessPCF(MQHCONN hConn,
                      MQHOBJ hEventQueue,
                      PMQCHAR pBuffer);

static PMQCHAR ParmToString(MQLONG Parameter);

/*****
/* Function: main */
*****/

int main(int argc, char * argv[])
{
    MQLONG CompCode;
    MQLONG Reason;
    MQHCONN hConn = MQHC_UNUSABLE_HCONN;
    MQOD ObjDesc = { MQOD_DEFAULT };
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1] = "";
    MQCHAR LogEvQ[MQ_Q_NAME_LENGTH] = "SYSTEM.ADMIN.LOGGER.EVENT";
    MQHOBJ hEventQueue = MQHO_UNUSABLE_HOBJ;
    PMQCHAR pBuffer = NULL;

    printf("\n/*****/\n");
    printf("/* Sample Logger Event Monitor start */\n");
    printf("/*****/\n");

    /*****
    /* Parse any command line options */
    *****/
    if (argc > 1)
    {
        strncpy(QMName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    }

    pBuffer = (PMQCHAR)malloc(MAX_MESSAGE_LENGTH);
    if (pBuffer == NULL)
    {
        printf("Can't allocate %d bytes\n", MAX_MESSAGE_LENGTH);
        goto MOD_EXIT;
    }

    /*****
    /* Connect to the specified (or default) queue manager */
    *****/
    MQCONN( QMName,
            &hConn,
            &CompCode,
            &Reason);

    if (Reason != MQRC_NONE)
    {
        printf("MQCONN ended with reason code %" Int32 "d\n", Reason);
        goto MOD_EXIT;
    }

    /*****
    /* Open the logger event queue for input */
    *****/
    strncpy(ObjDesc.ObjectQMgrName, QMName, MQ_Q_MGR_NAME_LENGTH);
    strncpy(ObjDesc.ObjectName, LogEvQ, MQ_Q_NAME_LENGTH);

    MQOPEN( hConn,
            &ObjDesc,
            MQOO_INPUT_EXCLUSIVE,
            &hEventQueue,
            &CompCode,
            &Reason );

    if (Reason != MQRC_NONE)
    {
        printf("MQOPEN failed for queue manager %.48s Queue %.48s Reason: %" Int32 "d\n",
              ObjDesc.ObjectQMgrName,
              ObjDesc.ObjectName,
              Reason);
    }
}

```

```

    goto MOD_EXIT;
}
else
{
    /*****
    /* Start processing event messages */
    /*****
    ProcessPCF(hConn, hEventQueue, pBuffer);
}

MOD_EXIT:
if (pBuffer != NULL)
{
    free(pBuffer);
}

/*****
/* Close the logger event queue */
/*****
if (hEventQueue != MQHO_UNUSABLE_HOBJ)
{
    MQCLOSE(hConn, &hEventQueue, MQCO_NONE, &CompCode, &Reason);
}

/*****
/* Disconnect */
/*****
if (hConn != MQHC_UNUSABLE_HCONN)
{
    MQDISC(&hConn, &CompCode, &Reason);
}

return 0;
}

/*****
/* Function: ProcessPCF */
/*****
/*
/* Input Parameters: Handle to queue manager connection */
/* Handle to the opened logger event queue object */
/* Pointer to a memory buffer to store the incoming PCF */
/* message */
/*
/* Output Parameters: None */
/*
/* Logic: Wait for messages to appear on the logger event queue and display */
/* their formatted contents. */
/*
/*
/*****

static void ProcessPCF(MQHCONN hConn,
                     MQHOBJ hEventQueue,
                     PMQCHAR pBuffer)
{
    MQCFH * pCfh;
    MQCFST * pCfst;
    MQGMO Gmo = { MQGMO_DEFAULT };
    MQMD Mqmd = { MQMD_DEFAULT };
    PMQCHAR pPCFCmd;
    MQLONG CompCode = MQCC_OK;
    MQLONG Reason = MQRC_NONE;
    MQLONG MsgLen;
    PMQCHAR Parm = NULL;

    Gmo.Options = MQGMO_WAIT +
                 MQGMO_CONVERT +
                 MQGMO_FAIL_IF QUIESCING;
    Gmo.WaitInterval = MQWI_UNLIMITED; /* Set timeout value */

    /*****
    /* Process response Queue */
    /*****
    while (Reason == MQRC_NONE)
    {
        memcpy(&Mqmd.MsgId, MQMI_NONE, sizeof(Mqmd.MsgId));
        memset(&Mqmd.CorrelId, 0, sizeof(Mqmd.CorrelId));

        MQGET( hConn,
              hEventQueue,
              &Mqmd,

```

```

        &Gmo,
        MAX_MESSAGE_LENGTH,
        pBuffer,
        &MsgLen,
        &CompCode,
        &Reason );

if (Reason != MQRC_NONE)
{
    switch(Reason)
    {
        case MQRC_NO_MSG_AVAILABLE:
            printf("Timed out");
            break;

        default:
            printf("MQGET ended with reason code %" Int32 "d\n", Reason);
            break;
    }
    goto MOD_EXIT;
}

/*****
/* Only expect PCF event messages on this queue */
/*****
if (memcmp(Mqmd.Format, MQFMT_EVENT, MQ_FORMAT_LENGTH))
{
    printf("Unexpected message format '%8.8s' received\n", Mqmd.Format);
    continue;
}

/*****
/* Build the output by parsing the received PCF message, first the */
/* header, then each of the parameters */
/*****
pCfh = (MQCFH *)pBuffer;

if (pCfh->Reason != MQRC_NONE)
{
    printf("-----\n");
    printf("Event Message Received\n");

    Parm = ParmToString(pCfh->Command);
    if (Parm != NULL)
    {
        printf("Command  :%s \n",Parm);
    }
    else
    {
        printf("Command  :%" Int32 "d \n",pCfh->Command);
    }

    printf("CompCode :%" Int32 "d\n" ,pCfh->CompCode);

    Parm = ParmToString(pCfh->Reason);
    if (Parm != NULL)
    {
        printf("Reason   :%s \n",Parm);
    }
    else
    {
        printf("Reason   :%" Int32 "d \n",pCfh->Reason);
    }
}

pPCFCmd = (PMQCHAR) (pCfh+1);
printf("-----\n");
while(pCfh->ParameterCount--)
{
    pCfst = (MQCFST *) pPCFCmd;
    switch(pCfst->Type)
    {
        case MQCFT_STRING:
            Parm = ParmToString(pCfst->Parameter);
            if (Parm != NULL)
            {
                printf("%-32s",Parm);
            }
            else
            {
                printf("%-32" Int32 "d",pCfst->Parameter);
            }
        }
    }
}

```

```

        fwrite(pCfst->String, pCfst->StringLength, 1, stdout);
        pPCFCmd += pCfst->StrucLength;
        break;

    default:
        printf("Unrecognised datatype %" Int32 "d returned\n", pCfst->Type);
        goto MOD_EXIT;
    }
    putchar('\n');
}

```

## Beispielausgabe

Diese Anwendung erzeugt die folgende Form der Ausgabe:

```

/*****
/* Sample Logger Event Monitor start */
*****/
-----
Event Message Received
Command :Logger Event Command
CompCode :0
Reason :Logger Status
-----
Queue Manager Name          CSIM

Current Log Extent          AMQA000001
Restart Log Extent          AMQA000001
Media Log Extent            AMQA000001
Log Path                     QMCSIM
-----

```

### Zugehörige Konzepte

„[Logger-Ereignisverwendung](#)“ auf Seite 49

Auf dieser Seite können Sie anzeigen, wie Sie Protokollerweiterungen verwenden können, um die Protokoll extents zu ermitteln, die für den Neustart des Warteschlangenmanagers oder die Datenträgerwiederherstellung nicht mehr benötigt werden.

„[Befehlsereignisverwendung](#)“ auf Seite 47

Auf dieser Seite können Sie anzeigen, wie Sie Befehlsereignisse verwenden können, um einen Prüfprotokoll für die Befehle zu generieren, die ausgeführt wurden.

### Zugehörige Verweise

„[Generierung von Protokollfunktionsereignis](#)“ auf Seite 48

Verwenden Sie diese Seite, um die Situationen anzuzeigen, die dazu führen, dass Protokollfunktionsereignisse generiert werden, und die Umstände zu verstehen, unter denen Protokollfunktionsereignisse nicht generiert werden.

## Berechtigungskonfigurationsereignisse

Berechtigungskonfigurationsereignisse werden ausgegeben, wenn an einer der Operationen zur Sicherheitssteuerungen eine Änderung über die Befehlszeile, MQSC-, PCF- oder entsprechende IBM i-Befehle vorgenommen wird.

Die Ereignisdaten enthalten die folgenden Informationen:

### Ursprungsinformationen

enthält den Warteschlangenmanager, von dem aus die Änderung vorgenommen wurde, die ID des Benutzers, der die Änderung vorgenommen hat, und die Art und Weise, in der die Änderung vorgenommen wurde, z. B. durch einen Konsolbefehl.

### Kontextinformationen

Ein Replikat der Kontextinformationen in den Nachrichtendaten aus der Befehlsnachricht.

Kontextinformationen sind in den Ereignisdaten enthalten, wenn der Befehl als Nachricht in der Warteschlange SYSTEM.ADMIN.COMMAND.QUEUE eingegeben wurde.

## **ID des Berechtigungsdatensatzes**

enthält den Profilnamen und den Objekttyp des Berechtigungsdatensatzes.

## **Objektattribute**

enthält die Werte aller Attribute im Berechtigungssatz.

Bei Satzereignissen der Änderungsberechtigung werden zwei Nachrichten generiert, eine mit den Informationen vor der Änderung, die andere mit den Informationen nach der Änderung.

Jede generierte Ereignisnachricht wird in die Warteschlange SYSTEM.ADMIN.CONFIG.EVENT gestellt.

## **Zugehörige Verweise**

„Ereignistypen“ auf Seite 8

Verwenden Sie diese Seite, um die Typen von Instrumentierungsereignis anzuzeigen, die ein Warteschlangenmanager oder eine Kanalinstanz melden kann.

## **Ereignisgenerierung der Berechtigungskonfiguration**

Verwenden Sie diese Seite, um die Situationen anzuzeigen, die dazu führen, dass Berechtigungskonfigurationsereignisse generiert werden, und die Umstände zu verstehen, unter denen Berechtigungskonfigurationsereignisse nicht generiert werden.

Berechtigungskonfigurationsereignisse benachrichtigen Sie über Änderungen an den Attributen eines Berechtigungsdatensatzes. Es gibt drei Typen von Berechtigungskonfigurationsereignissen:

- [Change Authority Record \(Berechtigungssatz ändern\)](#)
- [Berechtigungsdatensatz löschen](#)
- [Berechtigungsdatensatz aktualisieren](#)

Eine Berechtigungsergebnisnachricht wird in die Konfigurationsereigniswarteschlange gestellt, wenn das Attribut **CONFIGEV** des Warteschlangenmanagers auf *ENABLED* gesetzt ist und jeder der folgenden Befehle oder ihr MQSC-Äquivalent ausgegeben wird, auch wenn keine tatsächliche Änderung des Berechtigungsdatensatzes vorhanden ist:

- PCF-Befehl [Delete Authority Record](#)
- PCF-Befehl [Set Authority Record](#)
- Steuerbefehl [setmqaut](#)
- CL-Befehl [RVKMQMAUT](#)
- CL-Befehl [GRTMQMAUT](#)

## **Wenn Berechtigungskonfigurationsereignisse nicht generiert werden**

Die Ereignisnachrichten der Berechtigungskonfiguration werden unter den folgenden Umständen nicht generiert:

- Wenn ein Befehl fehlschlägt
- Wenn ein Warteschlangenmanager einen Fehler feststellt, der versucht, eine Nachricht in die Ereigniswarteschlange zu stellen, wird der Befehl beendet, aber es wird keine Ereignisnachricht generiert.
- Beim Erstellen oder Löschen eines Warteschlangenmanagers
- Wenn ein Objekt unabhängig von der Option **AUTHREC** im Löschbefehl gelöscht wird. Das entsprechende Befehlsereignis zeigt diese Operation an, die nicht auf den Berechtigungsdatensatz für einzelne Benutzer angewendet wird.

## **Zugehörige Konzepte**

„Befehlsereignisse“ auf Seite 45

Befehlsereignisse sind Benachrichtigungen, die ein MQSC- oder PCF-Befehl erfolgreich ausgeführt hat.

## **Zugehörige Verweise**

[REFRESH QMGR](#)

## Multi Beispielprogramm zur Überwachung von Instrumentierungsereignissen auf Multiplatforms

**amqsevt** formatiert die Instrumentierungsereignisse, die ein Warteschlangenmanager erstellen kann, und wird mit IBM MQ for Multiplatforms bereitgestellt. Das Programm liest Nachrichten aus Ereigniswarteschlangen und formatiert sie in lesbare Zeichenfolgen.

Als Musterprogramm werden sowohl die Quelle als auch die Binärdatei bereitgestellt. Das Beispiel wird auf allen Multiplatforms-Versionen (einschließlich IBM i) bereitgestellt.

Die einzelne Binärdatei `amqsevt` (oder `amqsevt.exe`) wird in den Beispieldatesets geliefert und im Verzeichnis 'bin' (`tools\c\samples\bin` oder `bin64`) installiert.

Die Quellendateien `amqsevt.c` werden ebenfalls in der Beispieldateigruppe geliefert und im Beispieldateiverzeichnis installiert, also `tools\c\samples` unter Windows.

Beachten Sie, dass das Programm aus mehreren Ereigniswarteschlangen lesen und mehrere Themen abonnieren kann. Verwenden Sie dazu `MQCB`, um die Nachrichten abzurufen.

Bei der Ausführung als Client kann das Beispiel eine Verbindung zu jedem Warteschlangenmanager (einschließlich z/OS) herstellen.

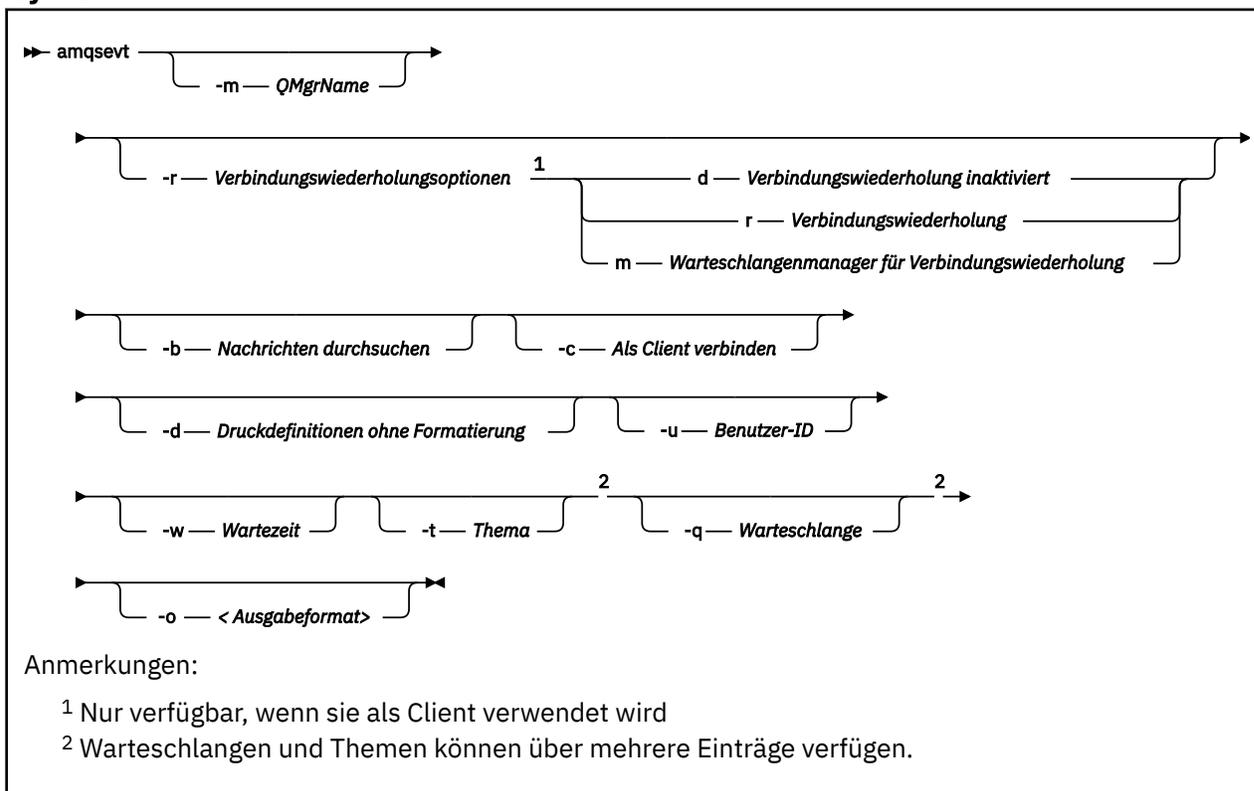


**Achtung:** Sie können das Programm ohne Angabe von Parametern verwenden. In diesem Fall versucht das Programm, eine Verbindung zum Standardwarteschlangenmanager herzustellen und Nachrichten aus der Standardgruppe von Ereigniswarteschlangen zu lesen (SYSTEM.ADMIN.\*.EVENT).

In dieser Situation wartet das Programm immer auf Nachrichten, bis Sie die Eingabetaste drücken, um das Programm zu beenden.

Es ist jedoch wahrscheinlicher, dass Sie das Programm mit den verschiedenen beschriebenen Optionen verwenden.

### Syntax



## Optionale Parameter

### **-m *QueueManagerName***

Geben Sie einen bestimmten Warteschlangenmanager für das Lesen von Ereignissen an.

### **-r *Reconnection Options***

Optionen für automatische Verbindungswiederverbindung, wenn sie als Client verwendet werden. Folgende Werte sind möglich:

**d**

Verbindung zum Client wiederherstellen

**r**

Verbindung zum Client wiederherstellen

**m**

Verbindung zum WS-Manager wiederherstellen

### **-b**

Nur Datensätze durchsuchen, statt die Nachrichten destruktiv zu lesen

### **-c**

Wählt die Verbindung als Client aus.

### **-d**

Wählt den Druckmodus aus, der im zweiten Beispiel verwendet wird. Die MQI-Konstanten werden genau so gedruckt, wie sie in den Headerdateien erscheinen.

### **-u *User ID***

Geben Sie einen bestimmten Benutzer an und veranlassen Sie, dass eine Eingabeaufforderung zum Anfordern eines Kennworts angezeigt wird

### **-w *Wait***

Bewirkt, dass das Programm beendet wird, wenn innerhalb der angegebenen Anzahl von Sekunden keine Ereignisnachrichten eingegangen sind.

Wenn Sie keine Zeit angeben, wird das Programm nur normal beendet, wenn Sie die Eingabetaste drücken.

### **-t *Topic* und**

### **-q *Queue***

Sowohl die Optionen **-q** als auch **-t** können in der Befehlszeile mehrfach angegeben werden.

Aus diesem Grund ist es möglich, aus einigen Standardwarteschlangen und aus Themen (wenn Ereignisse an sie gesendet werden) von einem einzigen Programmablauf aus zu lesen.

Wenn in der Befehlszeile keine Warteschlangen oder Topics benannt sind, werden die Standardereigniswarteschlangen geöffnet.

**Anmerkung:** Das Programm erkennt, ob es eine Verbindung zu einem z/OS -Warteschlangenmanager als Client hergestellt hat, und ändert die Standardgruppe der Ereigniswarteschlangen entsprechend, da z/OS nicht über das SYSTEM.ADMIN.LOGGER.EVENT -Warteschlange.

Wenn Themen verwendet werden, verwendet das Programm eine nicht permanente Subskription mit einer verwalteten Warteschlange, so dass alles beim Verlassen des Programms bereinigt wird.

**V 9.1.0**

### **-o *<output format>***

Das Format der Ausgabe. Mögliche Werte:

#### **Text**

Standardtextformat; dies ist der Standardwert.

#### **json**

Standard-JSON-Format. Alle Anwendungen, die JSON verstehen, können diese Ausgabe verwenden und sie direkt verarbeiten.

## Beispielausgabe

Die folgenden drei Beispiele zeigen die Ausgabe des Programms.

Im ersten Beispiel wird die Standardformatierungsoption verwendet, bei der das Programm die MQI-Definition eines Feldes verwendet und die Ausgabe formatiert, um die Ausgabe lesbarer zu machen.

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : Queue Mgr Event
Reason               : Unknown Alias Base Queue
Event created        : 2015/06/17 13:47:07.02 GMT
  Queue Mgr Name     : V8003_A
  Queue Name         : EVT.NO.BASE.QUEUE
  Base Object Name   : EVT.NOT.DEFINED
  Appl Type          : Unix
  Appl Name          : amqsput
  Base Type          : Queue
```

Das zweite Beispiel zeigt die alternative Formatierung unter Verwendung der Option -d, die nicht versucht, MQI-Konstanten zu übersetzen. Dies kann bei einigen Scripting-Tools, die nach bestimmten MQI-Werten suchen, besser sein.

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : MQCMD_Q_MGR_EVENT
Reason               : MQRC_UNKNOWN_ALIAS_BASE_Q
Event created        : 2015/06/17 13:52:48.18 GMT
  MQCA_Q_MGR_NAME    : V8003_A
  MQCA_Q_NAME        : EVT.NO.BASE.QUEUE
  MQCA_BASE_OBJECT_NAME : EVT.NOT.DEFINED
  MQIA_APPL_TYPE     : MQAT_UNIX
  MQCACF_APPL_NAME   : amqsput
  MQIA_BASE_TYPE     : MQOT_Q
```

### V 9.1.0

Das dritte Beispiel zeigt die JSON-Ausgabe des Befehls.

```
amqsevt -m QM1 -q SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE -o json
```

```
{
  "eventSource" : { "objectName": "SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE",
                    "objectType" : "Queue" },
  "eventType" : {
    "name" : "Activity Trace",
    "value" : 209
  },
  "eventReason" : {
    "name" : "None",
    "value" : 0
  },
  "eventCreation" : {
    "timeStamp" : "2018-07-10T12:44:26Z",
    "epoch" : 1531226666
  },
  "eventData" : {
    "queueMgrName" : "QM1",
    "hostName" : "<yourhostname>",
    "startDate" : "2018-07-10",
    "startTime" : "13:44:25",
    "endDate" : "2018-07-10",
    "endTime" : "13:44:26",
    "commandLevel" : 910,
    ...
  }
}
```

## Verwendungsbeispiel

Das folgende Beispiel zeigt, wie mehrere Warteschlangen verwendet werden:

```
amqsevt -m QM1 -q SYSTEM.ADMIN.QMGR.EVENT -q SYSTEM.ADMIN.PERM.EVENT -w 1
```

## Zugehörige Konzepte

„Ereignisüberwachung“ auf Seite 5

Die Ereignisüberwachung ist der Prozess zum Erkennen von Vorkommen von *Instrumentierungsereignissen* in einem WS-Manager-Netz. Ein Instrumentierungsereignis ist eine logische Kombination von Ereignissen, die von einem Warteschlangenmanager oder einer Kanalinstanz erkannt wird. Ein solches Ereignis bewirkt, dass der Warteschlangenmanager oder die Kanalinstanz eine spezielle Nachricht, die als *Ereignisnachricht* bezeichnet wird, in eine Ereigniswarteschlange eingibt.

„Instrumentierungsereignisse“ auf Seite 6

Ein Instrumentierungsereignis ist eine logische Kombination von Bedingungen, die ein Warteschlangenmanager oder eine Kanalinstanz feststellt und in eine Ereigniswarteschlange eine spezielle Nachricht, die so genannte *Ereignisnachricht*, einreicht.

## Zugehörige Verweise

C-Programmierung

„Beispielprogramm C zur Überwachung der Ereigniswarteschlange der Protokollfunktion“ auf Seite 50

Verwenden Sie diese Seite, um ein Beispiel-C-Programm anzuzeigen, das die Ereigniswarteschlange der Protokollfunktion für neue Ereignisnachrichten überwacht, diese Nachrichten liest und den Inhalt der Nachricht in stdout einreicht.

## Nachrichtenüberwachung

Die Nachrichtenüberwachung ist der Prozess, bei dem die Route identifiziert wird, die eine Nachricht über ein Warteschlangenmanagernetzwerk entnommen hat. Durch die Angabe der Aktivtypen und die Reihenfolge der Aktivitäten, die im Namen einer Nachricht ausgeführt werden, kann die Nachrichtenroute ermittelt werden.

Wenn eine Nachricht ein Warteschlangenmanagernetzwerk durchläuft, führen verschiedene Prozesse Aktivitäten im Namen der Nachricht aus. Verwenden Sie eine der folgenden Methoden, um eine Nachrichtenroute festzulegen:

- Die IBM MQ-Anwendung zur Routenanzeige (dspmqrte)
- Aktivitätenaufzeichnung
- Trace-Route-Messaging

Diese Techniken generieren alle spezielle Nachrichten, die Informationen zu den Aktivitäten enthalten, die in der Nachricht ausgeführt werden, während sie über ein WS-Manager-Netz übergeben werden. Verwenden Sie die Informationen, die in diesen speziellen Nachrichten zurückgegeben werden, um die folgenden Ziele zu erreichen:

- Nachrichtenaktivität aufzeichnen.
- Bestimmen Sie die letzte bekannte Position einer Nachricht.
- Erkennen Sie Routing-Probleme in Ihrem WS-Manager-Netz.
- Unterstützung bei der Bestimmung der Ursachen von Routing-Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Stellen Sie sicher, dass Ihr Warteschlangenmanagernetzwerk ordnungsgemäß ausgeführt wird.
- Machen Sie sich mit der Ausführung Ihres WS-Manager-Netzes vertraut.
- Trace für veröffentlichte Nachrichten durchführen.

## Zugehörige Verweise

Nachrichtentypen

## Aktivitäten und Operationen

Aktivitäten sind diskrete Aktionen, die eine Anwendung im Namen einer Nachricht ausführt. Aktivitäten bestehen aus Operationen, bei denen es sich um einzelne Arbeitsschritte handelt, die eine Anwendung ausführt.

Die folgenden Aktionen sind Beispiele für Aktivitäten:

- Ein Nachrichtenkanalagent (MCA) sendet eine Nachricht aus einer Übertragungswarteschlange in einen Kanal.
- Ein MCA empfängt eine Nachricht von einem Kanal und versetzt ihn in die Zielwarteschlange.
- Eine Anwendung, die eine Nachricht aus einer Warteschlange erhält und eine Antwortnachricht in eine Antwort eingibt.
- Die Publish/Subscribe-Engine von IBM MQ verarbeitet eine Nachricht.

Aktivitäten bestehen aus einer oder mehreren *Operationen*. Operationen sind einzelne Arbeitsvorgänge, die von einer Anwendung ausgeführt werden. Die Aktivität eines MCA, das eine Nachricht von einer Übertragungswarteschlange aus einem Kanal sendet, besteht beispielsweise aus den folgenden Operationen:

1. Abrufen einer Nachricht aus einer Übertragungswarteschlange (eine *Get* -Operation).
2. Sendet die Nachricht über einen Kanal (eine *Send* -Operation).

In einem Publish/Subscribe-Netz kann die Aktivität der Publish/Subscribe-Engine von IBM MQ, in der eine Nachricht verarbeitet wird, auf den folgenden Mehrfachoperationen bestehen:

1. Eine Nachricht in eine Themenzeichenfolge (eine *Put* -Operation) einreihen.
2. Null oder mehr Operationen für jeden der Subskribenten, die für den Empfang der Nachricht berücksichtigt werden (eine *Publish* -Operation, eine *Discarded Publish* -Operation oder eine *Excluded Publish* -Operation).

## Informationen aus Aktivitäten

Sie können die Reihenfolge der Aktivitäten ermitteln, die in einer Nachricht ausgeführt werden, indem Informationen aufgezeichnet werden, wenn die Nachricht über ein Warteschlangenmanagernetzwerk weitergeleitet wird. Sie können die Route einer Nachricht über das WS-Manager-Netz aus der Abfolge der Aktivitäten ermitteln, die in der Nachricht ausgeführt werden, und kann die folgenden Informationen abrufen:

### Die letzte bekannte Position einer Nachricht

Wenn eine Nachricht nicht das geplante Ziel erreicht, können Sie die letzte bekannte Position der Nachricht von einer vollständigen oder teilweisen Nachrichtenroute bestimmen.

### Konfigurationsprobleme mit einem WS-Manager-Netz

Wenn Sie den Leitweg einer Nachricht über ein Warteschlangenmanagernetzwerk untersuchen, wird möglicherweise angezeigt, dass die Nachricht nicht an die erwartete Position gegangen ist. Es gibt viele Gründe, warum dies eintreten kann, z. B. wenn ein Kanal inaktiv ist, kann die Nachricht eine alternative Route annehmen.

Bei einer Publish/Subscribe-Anwendung können Sie auch die Route einer Nachricht, die zu einem Thema veröffentlicht wird, sowie alle Nachrichten ermitteln, die in einem Warteschlangenmanager-Netz als Ergebnis der Veröffentlichung an Subskribenten gesendet werden.

In solchen Situationen kann ein Systemadministrator feststellen, ob Probleme im Warteschlangenmanagernetzwerk vorhanden sind, und korrigieren Sie gegebenenfalls die Fehler.

## Nachrichtenrouten

Abhängig von Ihrer Ursache für die Bestimmung einer Nachrichtenroute können Sie die folgenden allgemeinen Methoden verwenden:

### Für eine Trace-Route-Nachricht aufgezeichnete Aktivitätsinformationen verwenden

Trace-Route-Nachrichten zeichnen Aktivitätsinformationen für einen bestimmten Zweck auf. Sie können sie verwenden, um Konfigurationsprobleme mit einem WS-Manager-Netz zu bestimmen oder die letzte bekannte Position einer Nachricht zu ermitteln. Wenn eine Trace-Route-Nachricht generiert wird, um die letzte bekannte Position einer Nachricht zu ermitteln, die ihr Ziel nicht erreicht hat, kann sie die ursprüngliche Nachricht imitieren. Dies gibt der Trace-Route-Nachricht die größte Chance, die Route zu verfolgen, die von der ursprünglichen Nachricht übernommen wurde.

Die IBM MQ-Anwendung zur Routenanzeige kann Trace-Route-Nachrichten generieren.

## **Für die ursprüngliche Nachricht aufgezeichnete Aktivitätsinformationen verwenden**

Sie können jede Nachricht für die Aktivitätsaufzeichnung aktivieren und in ihrem Namen aufgezeichnete Aktivitätsinformationen enthalten. Wenn eine Nachricht ihr Ziel nicht erreicht, können Sie die aufgezeichneten Aktivitätsinformationen verwenden, um die letzte bekannte Position der Nachricht zu ermitteln. Durch die Verwendung von Aktivitätsinformationen aus der ursprünglichen Nachricht kann eine möglichst genaue Nachrichtenroute ermittelt werden, die zu der letzten bekannten Position führt. Um diese Methode verwenden zu können, muss die ursprüngliche Nachricht für die Aktivitätsaufzeichnung aktiviert sein.

**Warnung:** Vermeiden Sie es, alle Nachrichten in einem WS-Manager-Netz für die Aktivitätsaufzeichnung zu aktivieren. Nachrichten, die für die Aktivitätsaufzeichnung aktiviert sind, können viele Aktivitätsberichte in ihrem Namen generieren. Wenn jede Nachricht in einem WS-Manager-Netz für die Aktivitätsaufzeichnung aktiviert ist, kann der Datenverkehr des Warteschlangenmanagers auf ein inakzeptables Niveau ansteigen.

## **Zugehörige Konzepte**

„Nachrichtenüberwachung“ auf Seite 59

Die Nachrichtenüberwachung ist der Prozess, bei dem die Route identifiziert wird, die eine Nachricht über ein Warteschlangenmanagernetzwerk entnommen hat. Durch die Angabe der Aktivtypen und die Reihenfolge der Aktivitäten, die im Namen einer Nachricht ausgeführt werden, kann die Nachrichtenroute ermittelt werden.

„Nachrichtenroutetechniken“ auf Seite 61

Bei der Aufzeichnung von Aktivitäten und beim Tracing-Route-Messaging handelt es sich um Verfahren, mit denen Sie Aktivitätsinformationen für eine Nachricht aufzeichnen können, wenn sie über ein WS-Manager-Netz weitergeleitet werden.

„Trace-Route-Messaging“ auf Seite 68

Trace-Route-Messaging ist eine Technik, die *Trace-Route-Nachrichten* verwendet, um Aktivitätsinformationen für eine Nachricht aufzuzeichnen. Bei der Trace-Route-Nachrichtenübertragung wird eine Trace-Route-Nachricht in ein WS-Manager-Netz gesendet.

## **Zugehörige Tasks**

Schreiben eigener Nachrichtenkanalagenten

## **Nachrichtenroutetechniken**

Bei der Aufzeichnung von Aktivitäten und beim Tracing-Route-Messaging handelt es sich um Verfahren, mit denen Sie Aktivitätsinformationen für eine Nachricht aufzeichnen können, wenn sie über ein WS-Manager-Netz weitergeleitet werden.

## **Aktivitätenaufzeichnung**

Wenn eine Nachricht über die entsprechende Berichtsoption verfügt, fordert sie die Anwendungen auf, *Aktivitätsberichte* zu generieren, da sie über ein WS-Manager-Netz weitergeleitet werden. Wenn eine Anwendung eine Aktivität im Namen einer Nachricht ausführt, kann ein Aktivitätsbericht generiert und an eine geeignete Position übergeben werden. Ein Aktivitätsbericht enthält Informationen zu der Aktivität, die für die Nachricht ausgeführt wurde.

Die Aktivitätsinformationen, die mithilfe von Aktivitätsberichten erfasst werden, müssen in der Reihenfolge angeordnet werden, bevor eine Nachrichtenroute ermittelt werden kann.

## **Trace-Route-Messaging**

*Trace-Route-Messaging* ist eine Technik, die das Senden einer *Trace-Route-Nachricht* in ein Warteschlangenmanagernetzwerk umfasst. Wenn eine Anwendung eine Aktivität im Namen der Trace-Route-Nachricht ausführt, können Aktivitätsinformationen in den Nachrichtendaten der Trace-Route-Nachricht akkumuliert werden, oder es können Aktivitätsberichte generiert werden. Wenn die Aktivitätsinformationen in den Nachrichtendaten der Trace-Route-Nachricht akkumuliert werden, kann eine Trace-Route-Antwortnachricht, die alle Informationen aus der Trace-Route-Nachricht enthält, generiert und an eine geeignete Position zugestellt werden.

Da eine Trace-Route-Nachricht dediziert ist, um die Reihenfolge der Aktivitäten aufzuzeichnen, die in ihrem Namen ausgeführt werden, stehen im Vergleich zu normalen Nachrichten, die Aktivitätsberichte anfordern, mehr Verarbeitungsoptionen zur Verfügung.

### Vergleich von Aktivitätsaufzeichnung und Trace-Route-Messaging

Sowohl die Aktivitätsaufzeichnung als auch die Trace-Route-Nachrichtenübertragung kann Aktivitätsinformationen bereitstellen, um die Route zu ermitteln, die eine Nachricht über ein Warteschlangenmanager-Netzwerk hat. Beide Verfahren haben ihre eigenen Vorteile.

Vorteil	Aktivitätenaufzeichnung	Trace-Route-Messaging
Kann die letzte bekannte Position einer Nachricht ermitteln	Ja	Ja
Kann Konfigurationsprobleme mit einem Warteschlangenmanager-Netzwerk ermitteln	Ja	Ja
Kann von jeder Nachricht angefordert werden (ist nicht auf die Verwendung mit Trace-Route-Nachrichten beschränkt)	Ja	Nein
Die Nachrichtendaten werden nicht geändert.	Ja	Nein
Nachricht normal verarbeitet	Ja	Nein
Aktivitätsinformationen können in den Nachrichtendaten kumuliert werden.	Nein	Ja
Optionale Nachrichtenübermittlung an Zielwarteschlange	Nein	Ja
Wenn eine Nachricht in einer Endlosschleife abgefangen wird, kann sie erkannt und bearbeitet werden.	Nein	Ja
Aktivitätsinformationen können zuverlässig in Ordnung gebracht werden	Nein	Ja
Anwendung, die zum Anzeigen der Aktivitätsinformationen bereitgestellt wird	Nein	Ja

### Vollständigkeit der Nachrichtenroute

In einigen Fällen ist es nicht möglich, die vollständige Folge von Aktivitäten zu identifizieren, die im Namen einer Nachricht ausgeführt werden, sodass nur eine Teilnachrichtenroute ermittelt werden kann. Die Vollständigkeit einer Nachrichtenroute wird direkt durch das WS-Manager-Netz beeinflusst, über das die Nachrichten weitergeleitet werden. Die Vollständigkeit einer Nachrichtenroute hängt wie folgt von der Ebene der Warteschlangenmanager im WS-Manager-Netz ab:

#### Warteschlangenmanager in IBM WebSphere MQ 6.0 und nachfolgenden Releases

MCAs und benutzerdefinierte Anwendungen, die mit Warteschlangenmanager in IBM WebSphere MQ 6.0 oder nachfolgenden Releases verbunden sind, können Informationen zu den Aktivitäten aufzeichnen, die im Auftrag einer Nachricht ausgeführt werden. Die Aufzeichnung von Aktivitätsinformationen wird von den WS-Managerattributen ACTIVREC und ROUTEREC gesteuert. Wenn sich ein Warteschlangenmanagernetz nur aus Warteschlangenmanagern in IBM WebSphere MQ 6.0 oder nachfolgenden Releases zusammensetzt, können vollständige Nachrichtenrouten ermittelt werden.

#### IBM WebSphere MQ-Warteschlangenmanager vor IBM WebSphere MQ 6.0

Anwendungen, die mit Warteschlangenmanagern vor IBM WebSphere MQ 6.0 verbunden sind, zeichnen **keine** Aktivitäten auf, die im Auftrag einer Nachricht ausgeführt wurden. Wenn ein Warteschlangenmanagernetz einen Warteschlangenmanager vor IBM WebSphere MQ 6.0 enthält, kann nur ein Teil der Nachrichtenroute ermittelt werden.

## Wie Aktivitätsinformationen gespeichert werden

IBM MQ speichert Aktivitätsinformationen in Aktivitätsberichten, Trace-Route-Nachrichten oder Trace-Route-Antwortnachrichten. In jedem Fall werden die Informationen in einer Struktur gespeichert, die als PCF-Gruppe *Aktivität* bezeichnet wird. Eine Trace-Route-Nachrichten- oder Trace-Route-Antwortnachricht kann viele Aktivitäts-PCF-Gruppen enthalten, abhängig von der Anzahl der Aktivitäten, die in der Nachricht ausgeführt werden. Aktivitätsberichte enthalten eine Aktivität-PCF-Gruppe, da für jede aufgezählte Aktivität ein eigener Aktivitätsbericht generiert wird.

Bei Trace-Route-Messaging können zusätzliche Informationen aufgezeichnet werden. Diese zusätzlichen Informationen werden in einer Struktur gespeichert, die als PCF-Gruppe *TraceRoute* bezeichnet wird. Die PCF-Gruppe 'TraceRoute' enthält eine Reihe von PCF-Strukturen, die zum Speichern zusätzlicher Aktivitätsinformationen und zum Angeben von Optionen verwendet werden, die festlegen, wie die Trace-Route-Nachricht gehandhabt wird, wenn sie über ein WS-Manager-Netz weitergeleitet wird.

### Zugehörige Konzepte

[„Aktivitätenaufzeichnung“ auf Seite 63](#)

Bei der Aktivitätsaufzeichnung handelt es sich um ein Verfahren zum Bestimmen der Routen, die Nachrichten über ein Warteschlangenmanagernetzwerk durchlaufen. Um die Route zu ermitteln, die eine Nachricht genommen hat, werden die Aktivitäten aufgezeichnet, die für die Nachricht ausgeführt werden.

[„Trace-Route-Messaging“ auf Seite 68](#)

Trace-Route-Messaging ist eine Technik, die *Trace-Route-Nachrichten* verwendet, um Aktivitätsinformationen für eine Nachricht aufzuzeichnen. Bei der Trace-Route-Nachrichtenübertragung wird eine Trace-Route-Nachricht in ein WS-Manager-Netz gesendet.

### Zugehörige Verweise

[„PCF-Gruppe "TraceRoute"“ auf Seite 74](#)

Die Attribute in der PCF-Gruppe von *TraceRoute* steuern das Verhalten einer Trace-Route-Nachricht. Die *TraceRoute* -PCF-Gruppe befindet sich in den Nachrichtendaten jeder Trace-Route-Nachricht.

[„Aktivitätsbericht-Nachrichtendaten“ auf Seite 111](#)

Verwenden Sie diese Seite, um die Parameter anzuzeigen, die von der *Activity* -PCF-Gruppe in einer Aktivitätsberichtsnachricht enthalten sind. Einige Parameter werden nur zurückgegeben, wenn bestimmte Operationen ausgeführt wurden.

## Aktivitätenaufzeichnung

Bei der Aktivitätsaufzeichnung handelt es sich um ein Verfahren zum Bestimmen der Routen, die Nachrichten über ein Warteschlangenmanagernetzwerk durchlaufen. Um die Route zu ermitteln, die eine Nachricht genommen hat, werden die Aktivitäten aufgezeichnet, die für die Nachricht ausgeführt werden.

Wenn Sie die Aktivitätsaufzeichnung verwenden, kann jede Aktivität, die für eine Nachricht ausgeführt wird, in einem Aktivitätsbericht aufgezeichnet werden. Ein Aktivitätsbericht ist ein Typ von Berichtsnachricht. Jeder Aktivitätsbericht enthält Informationen über die Anwendung, die die Aktivität im Namen der Nachricht ausgeführt hat, wann die Aktivität stattgefunden hat, sowie Informationen zu den Operationen, die als Teil der Aktivität ausgeführt wurden. Aktivitätsberichte werden in der Regel an eine Warteschlange für Antworten zugestellt, in der sie zusammen erfasst werden. Durch das Studium der Aktivitätsberichte, die sich auf eine Nachricht beziehen, können Sie die Route ermitteln, die die Nachricht über das WS-Manager-Netz genommen hat.

## Verwendung des Aktivitätsberichts

Wenn Nachrichten über ein WS-Manager-Netz weitergeleitet werden, können Aktivitätsberichte generiert werden. Sie können die Informationen zum Aktivitätsbericht auf folgende Arten verwenden:

### Die letzte bekannte Position einer Nachricht ermitteln

Wenn eine Nachricht, die für die Aktivitätsaufzeichnung aktiviert ist, nicht zu ihrem vorgesehenen Ziel gelangt, können Aktivitätsberichte, die für die Nachricht generiert wurden, wie sie durch ein WS-Manager-Netz weitergeleitet wurde, untersucht werden, um die letzte bekannte Position der Nachricht zu ermitteln.

## Konfigurationsprobleme mit einem WS-Manager-Netz bestimmen

Eine Reihe von Nachrichten, die für die Aktivitätsaufzeichnung aktiviert sind, kann in ein WS-Manager-Netz gesendet werden. Durch die Untersuchung der Tätigkeitsberichte, die sich auf die einzelnen Nachrichten beziehen, kann erkennbar werden, dass sie nicht die erwartete Route übernommen haben. Es gibt viele Gründe, warum dies geschehen kann, z. B. ein Kanal hätte gestoppt werden können, wodurch die Nachricht gezwungen wurde, eine alternative Route zu verwenden. In diesen Situationen kann ein Systemadministrator feststellen, ob Probleme im Warteschlangenmanagernetzwerk vorhanden sind, und falls vorhanden, korrigieren Sie diese.

**Anmerkung:** Sie können die Aktivitätsaufzeichnung in Verbindung mit Trace-Route-Nachrichten verwenden, indem Sie die IBM MQ-Anwendung zur Routenanzeige verwenden.

## Aktivitätsberichtsformat

Aktivitätsberichte sind PCF-Nachrichten, die von Anwendungen generiert werden, die eine Aktivität im Namen einer Nachricht ausgeführt haben. Aktivitätsberichte sind standardmäßige IBM MQ-Berichtsnachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten wie folgt enthalten:

### Der Nachrichtendeskriptor

- Eine MQMD-Struktur

### Nachrichtendaten

- Eingebetteter PCF-Header (MQEPH)
- Aktivitätsbericht-Nachrichtendaten

Die Nachrichtendaten des Aktivitätsberichts bestehen aus der *Activity* -PCF-Gruppe und, wenn sie für eine Trace-Route-Nachricht generiert wurde, die *TraceRoute* -PCF-Gruppe.

### Zugehörige Verweise

[MQMD - Nachrichtendeskriptor](#)

[MQEPH - Eingebetteter PCF-Header](#)

## Aktivitätsaufzeichnung steuern

Aktivieren Sie die Aktivitätsaufzeichnung auf WS-Managerebene. Wenn Sie ein vollständiges WS-Manager-Netz aktivieren möchten, aktivieren Sie jeden Warteschlangenmanager im Netz für die Aktivitätsaufzeichnung einzeln. Wenn Sie mehr WS-Manager aktivieren, werden weitere Aktivitätsberichte generiert.

## Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um Aktivitätsberichte für eine Nachricht zu generieren, wenn sie über einen Warteschlangenmanager weitergeleitet wird: Definieren Sie die Nachricht zum Anfordern von Aktivitätsberichten; aktivieren Sie den Warteschlangenmanager für die Aktivitätsaufzeichnung; und stellen Sie sicher, dass Anwendungen, die Aktivitäten in der Nachricht ausführen, in der Lage sind, Aktivitätsberichte zu generieren.

Wenn keine Aktivitätsberichte für eine Nachricht generiert werden sollen, da sie über einen Warteschlangenmanager weitergeleitet wird, *inaktivieren* Sie den Warteschlangenmanager für die Aktivitätsaufzeichnung.

## Vorgehensweise

1. Anforderungsaktivitätsberichte für eine Nachricht
  - a) Geben Sie im Nachrichtendeskriptor der Nachricht MQRO\_ACTIVITY in das Feld *Report* ein.
  - b) Geben Sie im Nachrichtendeskriptor der Nachricht den Namen einer Empfangswarteschlange für Antworten in das Feld *ReplyToQ* an.

**Warnung:** Vermeiden Sie es, alle Nachrichten in einem WS-Manager-Netz für die Aktivitätsaufzeichnung zu aktivieren. Nachrichten, die für die Aktivitätsaufzeichnung aktiviert sind, können viele Aktivitätsberichte in ihrem Namen generieren. Wenn jede Nachricht in einem WS-Manager-Netz für die

Aktivitätsaufzeichnung aktiviert ist, kann der Datenverkehr des Warteschlangenmanagers auf ein inakzeptables Niveau ansteigen.

## 2. Aktivieren oder Inaktivieren des Warteschlangenmanagers für die Aktivitätsaufzeichnung.

Verwenden Sie den MQSC-Befehl `ALTER QMGR` unter Angabe des Parameters `ACTIVREC`, um den Wert des Warteschlangenmanagerattributs zu ändern. Folgende Werte sind möglich:

### **MSG**

Der WS-Manager ist für die Aktivitätsaufzeichnung aktiviert. Alle generierten Aktivitätsberichte werden an die Warteschlange für Antwortnachrichten übergeben, die im Nachrichtendeskriptor der Nachricht angegeben ist. Dies ist der Standardwert.

### **WARTESCHLANGE**

Der WS-Manager ist für die Aktivitätsaufzeichnung aktiviert. Alle generierten Aktivitätsberichte werden an die lokale Systemwarteschlange `SYSTEM.ADMIN.ACTIVITY.QUEUE` übergeben. Die Systemwarteschlange kann auch verwendet werden, um Aktivitätsberichte an eine allgemeine Warteschlange weiterzuleiten.

### **INAKTIVIERT**

Der WS-Manager ist für die Aktivitätsaufzeichnung inaktiviert. Es werden keine Aktivitätsberichte generiert, die im Geltungsbereich dieses Warteschlangenmanagers generiert werden.

Um beispielsweise einen Warteschlangenmanager für die Aktivitätsaufzeichnung zu aktivieren und anzugeben, dass alle generierten Aktivitätsberichte an die lokale Systemwarteschlange `SYSTEM.ADMIN.ACTIVITY.QUEUE` übergeben werden, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QMGR ACTIVREC(Queue)
```

**Hinweis:** Wenn Sie das WS-Manager-Attribut `ACTIVREC` ändern, erkennt ein laufender MCA die Änderung erst, wenn der Kanal erneut gestartet wird.

3. Stellen Sie sicher, dass Ihre Anwendung denselben Algorithmus verwendet wie MCAs, um zu bestimmen, ob ein Aktivitätsbericht für eine Nachricht generiert werden soll:
  - a) Stellen Sie sicher, dass die Nachricht Aktivitätsberichte angefordert hat, die generiert werden sollen.
  - b) Überprüfen Sie, ob der Warteschlangenmanager, in dem die Nachricht derzeit gespeichert ist, für die Aktivitätsaufzeichnung aktiviert ist.
  - c) Den Aktivitätsbericht in die Warteschlange stellen, die vom Warteschlangenmanager des `ACTIVREC` bestimmt wird.

## ***Einrichten einer allgemeinen Warteschlange für Aktivitätsberichte***

Um die Positionen der Aktivitätsberichte zu bestimmen, die sich auf eine bestimmte Nachricht beziehen, wenn die Berichte an die lokale Systemwarteschlange zugestellt werden, ist es effizienter, eine gemeinsame Warteschlange auf einem einzigen Knoten zu verwenden.

## **Vorbereitende Schritte**

Definieren Sie den Parameter **ACTIVREC**, um den Warteschlangenmanager für die Aktivitätsaufzeichnung zu aktivieren, und geben Sie an, dass alle generierten Aktivitätsberichte an die lokale Systemwarteschlange `SYSTEM.ADMIN.ACTIVITY.QUEUE` übergeben werden.

## **Informationen zu diesem Vorgang**

Wenn eine Reihe von Warteschlangenmanagern in einem Warteschlangenmanagernetzwerk so eingestellt sind, dass sie Aktivitätsberichte an die lokale Systemwarteschlange zustellen, kann es zeitaufwendig sein, die Positionen der Aktivitätsberichte zu einer bestimmten Nachricht zu ermitteln. Alternativ können Sie auch einen einzigen Knoten verwenden, der ein Warteschlangenmanager ist, der eine gemeinsame Warteschlange hostet. Alle Warteschlangenmanager in einem WS-Manager-Netz können Aktivitätsberichte an diese allgemeine Warteschlange zustellen. Der Vorteil der Verwendung einer allgemeinen Warteschlange besteht darin, dass WS-Manager keine Aktivitätsberichte an die in einer Nachricht angegebene Warte-

schlange für Antwortnachrichten übergeben müssen und wenn die Positionen der Aktivitätsberichte, die sich auf eine Nachricht beziehen, nur eine Warteschlange abfragen.

Führen Sie die folgenden Schritte aus, um eine allgemeine Warteschlange zu konfigurieren:

## Vorgehensweise

1. Wählen Sie einen Warteschlangenmanager als einen einzigen Knoten aus oder definieren Sie diesen.
2. Wählen Sie auf dem einzelnen Knoten eine Warteschlange für die Verwendung als allgemeine Warteschlange aus oder definieren Sie sie.
3. Auf allen WS-Managern, in denen Aktivitätsberichte an die allgemeine Warteschlange übergeben werden sollen, definieren Sie die lokale Systemwarteschlange `SYSTEM.ADMIN.ACTIVITY.QUEUE` als ferne Warteschlangendefinition erneut:
  - a) Geben Sie den Namen des einzelnen Knotens als Name des fernen Warteschlangenmanagers an.
  - b) Geben Sie den Namen der allgemeinen Warteschlange als Namen der fernen Warteschlange an.

## Nachrichtenrouteninformationen bestimmen

Um eine Nachrichtenroute zu ermitteln, müssen Sie die Informationen aus den erfassten Aktivitätsberichten abrufen. Stellen Sie fest, ob genügend Aktivitätsberichte in der Empfangswarteschlange für Antworten vorhanden sind, damit Sie die erforderlichen Informationen ermitteln und die Aktivitätsberichte in der angegebenen Reihenfolge anordnen können.

## Informationen zu diesem Vorgang

Die Reihenfolge, in der die Aktivitätsberichte in die Warteschlange für Antwortantworten gestellt werden, korreliert nicht unbedingt mit der Reihenfolge, in der die Aktivitäten ausgeführt wurden. Sie müssen Aktivitätsberichte manuell anfordern, sofern sie nicht für eine Trace-Route-Nachricht generiert wurden. In diesem Fall können Sie Aktivitätsberichte mit der IBM MQ-Anwendung zur Routenanzeige anfordern.

Stellen Sie fest, ob genügend Aktivitätsberichte in der Empfangswarteschlange für Antworten enthalten sind, damit Sie die erforderlichen Informationen abrufen können:

## Vorgehensweise

1. Identifizieren Sie alle zugehörigen Aktivitätsberichte in der Warteschlange für Antwortnachrichten, indem Sie die IDs der Aktivitätsberichte und die ursprüngliche Nachricht vergleichen. Stellen Sie sicher, dass Sie die Berichtsoption der ursprünglichen Nachricht so festlegen, dass die Aktivitätsberichte mit der ursprünglichen Nachricht korreliert werden können.
2. Sortieren Sie die angegebenen Aktivitätsberichte aus der Antwortwarteschlange.  
Sie können die folgenden Parameter aus dem Aktivitätsbericht verwenden:

### **OperationType**

Mit den Typen der ausgeführten Operationen können Sie den Aktivitätsbericht, der direkt vor oder nach dem aktuellen Aktivitätsbericht generiert wurde, bestimmen.

Ein Aktivitätsbericht zeigt beispielsweise an, dass ein MCA eine Nachricht von einer Übertragungswarteschlange in einen Kanal gesendet hat. Die letzte Operation, die in dem Aktivitätsbericht detailliert beschrieben wird, enthält einen *OperationType* von `send` und Details, die die Nachricht mit dem Kanal CH1 an den Zielwarteschlangenmanager QM1 gesendet hat. Dies bedeutet, dass die nächste Aktivität, die in der Nachricht ausgeführt wird, auf dem Warteschlangenmanager QM1 aufgetreten ist und mit einer `receive`-Operation aus Kanal CH1 begonnen hat. Wenn Sie diese Informationen verwenden, können Sie den nächsten Aktivitätsbericht angeben, sofern er vorhanden ist und erworben wurde.

### **OperationDate und OperationTime**

Sie können die allgemeine Reihenfolge der Aktivitäten aus den Datums- und Zeitangaben der Operationen in den einzelnen Aktivitätsberichten ermitteln.

**Warnung:** Wenn nicht jeder WS-Manager im WS-Manager-Netz die Systemuhren synchronisiert hat, ist die Sortierung nach Datum und Uhrzeit nicht garantiert, dass die Aktivitätsberichte in der richtigen Reihenfolge enthalten sind. Sie müssen die Reihenfolge manuell erstellen.

Die Reihenfolge der Aktivitätsberichte stellt die Route oder Teilanroute dar, die die Nachricht über das WS-Manager-Netz genommen hat.

3. Fordern Sie die Informationen, die Sie benötigen, aus den Aktivitätsinformationen in den geordneten Aktivitätsberichten an.

Wenn Sie nicht genügend Informationen über die Nachricht haben, können Sie möglicherweise weitere Aktivitätsberichte anfordern.

### **Weitere Aktivitätsberichte abrufen**

Um eine Nachrichtenroute zu ermitteln, müssen in den erfassten Aktivitätsberichten genügend Informationen zur Verfügung stehen. Wenn Sie die Aktivitätsberichte zu einer Nachricht aus der Warteschlange für Antwortnachrichten abrufen, die die angegebene Nachricht enthält, Sie jedoch nicht über die erforderlichen Informationen verfügen, suchen Sie nach weiteren Aktivitätsberichten.

### **Informationen zu diesem Vorgang**

Führen Sie die folgenden Schritte aus, um die Positionen aller weiteren Aktivitätsberichte zu ermitteln:

### **Vorgehensweise**

1. Bei allen WS-Managern im WS-Manager-Netz, die Aktivitätsberichte an eine allgemeine Warteschlange zustellen, rufen Sie Aktivitätsberichte aus der allgemeinen Warteschlange ab, die über einen *CorrelId* verfügen, der mit dem *MsgId* der ursprünglichen Nachricht übereinstimmt.
2. Für alle WS-Manager im WS-Manager-Netz, die keine Aktivitätsberichte an eine allgemeine Warteschlange zustellen, rufen Sie die Aktivitätsberichte wie folgt ab:
  - a) Untersuchen Sie die vorhandenen Aktivitätsberichte, um die Warteschlangenmanager zu identifizieren, über die die Nachricht weitergeleitet wurde.
  - b) Geben Sie für diese WS-Manager die Warteschlangenmanager an, die für die Aktivitätsaufzeichnung aktiviert sind.
  - c) Geben Sie für diese Warteschlangenmanager alle an, die keine Aktivitätsberichte an die angegebene Empfangswarteschlange für Antworten zurückgegeben haben.
  - d) Überprüfen Sie für jeden der Warteschlangenmanager, die Sie identifizieren, die Systemwarteschlange SYSTEM.ADMIN.ACTIVITY.QUEUE, und rufen Sie alle Aktivitätsberichte ab, die einen *CorrelId* aufweisen, der mit dem *MsgId* der ursprünglichen Nachricht übereinstimmt.
  - e) Wenn Sie keine Aktivitätsberichte in der Systemwarteschlange finden, überprüfen Sie die Warteschlange für nicht zustellbare Nachrichten des Warteschlangenmanagers, sofern eine vorhanden ist.

Ein Aktivitätsbericht kann nur dann an eine Warteschlange für nicht zustellbare Nachrichten zugestellt werden, wenn die Berichtsoption MQRO\_DEAD\_LETTER\_Q festgelegt ist.

3. Ordnen Sie alle erfassten Aktivitätsberichte in der angegebenen Reihenfolge an.

Die Reihenfolge der Aktivitätsberichte stellt dann die Route oder die Teilroute dar, die die Nachricht übernommen hat.

4. Fordern Sie die Informationen, die Sie benötigen, aus den Aktivitätsinformationen in den geordneten Aktivitätsberichten an.

In einigen Fällen können die aufgezeichneten Aktivitätsinformationen die angegebene Empfangswarteschlange für Antworten, eine allgemeine Warteschlange oder eine Systemwarteschlange nicht erreichen.

### **Umstände, in denen Aktivitätsinformationen nicht erfasst werden**

Um die vollständige Abfolge der Aktivitäten zu ermitteln, die im Namen einer Nachricht ausgeführt wurden, müssen Informationen zu jeder Aktivität angefordert werden. Wenn die Informationen zu einer

Aktivität nicht aufgezeichnet wurden oder nicht erfasst wurden, können Sie nur eine Teilsequenz von Aktivitäten bestimmen.

Aktivitätsinformationen werden unter den folgenden Umständen nicht erfasst:

- Die Nachricht wird von einem Warteschlangenmanager vor IBM WebSphere MQ 6.0 verarbeitet.
- Die Nachricht wird von einem WS-Manager verarbeitet, der für die Aktivitätsaufzeichnung nicht aktiviert ist.
- Die Anwendung, die die Verarbeitung der Nachricht erwartet hat, ist nicht aktiv.

Die aufgezeichneten Aktivitätsinformationen können die angegebene Warteschlange für Antwortwarteschlangen unter den folgenden Umständen nicht erreichen:

- Es ist kein Kanal definiert, um Aktivitätsberichte an die Warteschlange für Antwortantworten weiterzuleiten.
- Der Kanal, in dem Aktivitätsberichte an die Empfangswarteschlange für Antworten weitergeleitet werden, ist nicht aktiv.
- Die Definition der fernen Warteschlange zur Weiterleitung von Aktivitätsberichten an den Warteschlangenmanager, in dem sich die Warteschlange für Antwortantworten befindet (der Aliasname des Warteschlangenmanagers), ist nicht definiert.
- Der Benutzer, der die ursprüngliche Nachricht generiert hat, hat keine Berechtigung für den Aliasnamen des Warteschlangenmanagers, oder sie hat keine Berechtigung.
- Der Benutzer, der die ursprüngliche Nachricht generiert hat, hat nicht die Berechtigung zum Öffnen oder zum Angeben von Berechtigungen für die Warteschlange für Antwortnachrichten.
- Die Empfangswarteschlange für Antworten wird unterdrückt.

Die aufgezeichneten Aktivitätsinformationen können die Systemwarteschlange oder eine allgemeine Warteschlange unter den folgenden Umständen nicht erreichen:

- Wenn eine allgemeine Warteschlange verwendet werden soll und kein Kanal definiert ist, um Aktivitätsberichte an die allgemeine Warteschlange weiterzuleiten.
- Wenn eine allgemeine Warteschlange verwendet werden soll und der Kanal die Aktivitätsberichte an die allgemeine Warteschlange weiterleiten soll, wird nicht ausgeführt.
- Wenn eine allgemeine Warteschlange verwendet werden soll und die Systemwarteschlange nicht ordnungsgemäß definiert ist.
- Der Benutzer, der die ursprüngliche Nachricht generiert hat, hat keine Berechtigung für die Systemwarteschlange, oder sie hat die Berechtigung für die Systemwarteschlange.
- Die Systemwarteschlange wird unterdrückt.
- Wenn eine allgemeine Warteschlange verwendet werden soll und der Benutzer, der die ursprüngliche Nachricht generiert hat, keine Berechtigung für die allgemeine Warteschlange hat oder diese in die Warteschlange eingibt, ist die Berechtigung für die allgemeine Warteschlange nicht vorhanden.
- Wenn eine gemeinsame Warteschlange verwendet werden soll und die gemeinsame Warteschlange unterdrückt wird.

Unter diesen Umständen verfügt der Aktivitätsbericht nicht über die angegebene Berichtsoption `MQRO_DISCARD_MSG`, der Aktivitätsbericht kann aus einer Warteschlange für nicht zustellbare Nachrichten abgerufen werden, wenn eine Warteschlange auf dem Warteschlangenmanager definiert wurde, in der der Aktivitätsbericht zurückgewiesen wurde. In einem Aktivitätsbericht wird nur diese Berichtsoption angegeben, wenn die ursprüngliche Nachricht, von der aus der Aktivitätsbericht generiert wurde, sowohl `MQRO_PASS_DISCARD_AND_EXPIRY` als auch `MQRO_DISCARD_MSG` im Berichtsfeld des Nachrichten-deskriptors angegeben hat.

## Trace-Route-Messaging

Trace-Route-Messaging ist eine Technik, die *Trace-Route-Nachrichten* verwendet, um Aktivitätsinformationen für eine Nachricht aufzuzeichnen. Bei der Trace-Route-Nachrichtenübertragung wird eine Trace-Route-Nachricht in ein WS-Manager-Netz gesendet.

Da die Trace-Route-Nachricht über das WS-Manager-Netz weitergeleitet wird, werden die Aktivitätsinformationen aufgezeichnet. Diese Aktivitätsinformationen enthalten Informationen zu den Anwendungen, die die Aktivitäten ausgeführt haben, wann sie ausgeführt wurden, und die Operationen, die als Teil der Aktivitäten ausgeführt wurden. Sie können die Informationen, die im Trace-Route-Messaging aufgezeichnet wurden, für die folgenden Zwecke verwenden:

#### **So ermitteln Sie die letzte bekannte Position einer Nachricht:**

Wenn eine Nachricht ihr Ziel nicht erreicht, können Sie die Aktivitätsinformationen, die für eine Trace-Route-Nachricht aufgezeichnet wurden, verwenden, um die letzte bekannte Position der Nachricht zu ermitteln. Es wird eine Trace-Route-Nachricht in ein WS-Manager-Netz mit derselben Zieladresse wie die ursprüngliche Nachricht gesendet, die die gleiche Route verfolgt. Aktivitätsinformationen können in den Nachrichtendaten der Trace-Route-Nachricht kumuliert oder mithilfe von Aktivitätsberichten aufgezeichnet werden. Um die Wahrscheinlichkeit zu erhöhen, dass die Trace-Route-Nachricht auf dieselbe Route wie die ursprüngliche Nachricht folgt, können Sie die Trace-Route-Nachricht ändern, um die ursprüngliche Nachricht zu imitieren.

#### **Gehen Sie wie folgt vor, um Konfigurationsprobleme mit einem Warteschlangenmanagernetz**

Trace-Route-Nachrichten werden in ein WS-Manager-Netz gesendet, und die Aktivitätsinformationen werden aufgezeichnet. Wenn Sie die für eine Trace-Route-Nachricht aufgezeichneten Aktivitätsinformationen untersuchen, kann erkennbar werden, dass die Trace-Route-Nachricht nicht dem erwarteten Leitweg folgt. Es gibt viele Gründe, warum dies zu einem solchen Fall kommen kann, z. B. wenn ein Kanal inaktiv sein könnte, wodurch die Nachricht gezwungen wird, einen alternativen Leitweg zu nehmen. In diesen Situationen kann ein Systemadministrator feststellen, ob Probleme im Warteschlangenmanagernetzwerk vorhanden sind, und falls vorhanden, korrigieren Sie diese.

Mit der IBM MQ-Anwendung zur Routenanzeige können Sie Trace-Route-Nachrichten konfigurieren, generieren und in ein Warteschlangenmanagernetz stellen.

**Warnung:** Wenn Sie eine Trace-Route-Nachricht in eine Verteilerliste stellen, sind die Ergebnisse nicht definiert.

#### **Zugehörige Konzepte**

„Trace-Route-Nachrichtenreferenz“ auf Seite 128

Verwenden Sie diese Seite, um eine Übersicht über das Nachrichtenformat der Traceroute zu erhalten. Die Trace-Route-Nachrichtendaten enthalten Parameter, die die Aktivitäten beschreiben, die von der Trace-Route-Nachricht verursacht wurden.

#### **Wie Aktivitätsinformationen aufgezeichnet werden**

Bei der Trace-Route-Nachrichtenübertragung können Sie Aktivitätsinformationen in den Nachrichtendaten der Trace-Route-Nachricht aufzeichnen oder Aktivitätsberichte verwenden. Alternativ können Sie beide Techniken verwenden.

#### **Akkumulierende Aktivitätsinformationen in den Nachrichtendaten der Trace-Route-Nachricht**

Da eine Trace-Route-Nachricht über ein WS-Manager-Netz weitergeleitet wird, können Informationen zu den Aktivitäten, die im Namen der Trace-Route-Nachricht ausgeführt werden, in den Nachrichtendaten der Trace-Route-Nachricht kumuliert werden. Die Aktivitätsinformationen werden in *Activity*-PCF-Gruppen gespeichert. Für jede Aktivität, die im Namen der Trace-Route-Nachricht ausgeführt wird, wird eine *Activity*-PCF-Gruppe an das Ende des PCF-Blocks in den Nachrichtendaten der Trace-Route-Nachricht geschrieben.

Weitere Aktivitätsinformationen werden im Trace-Route-Messaging in einer PCF-Gruppe aufgezeichnet, die als PCF-Gruppe von *TraceRoute* bezeichnet wird. Die zusätzlichen Aktivitätsinformationen werden in dieser PCF-Gruppe gespeichert und können dazu verwendet werden, die Reihenfolge der aufgezeichneten Aktivitäten zu bestimmen. Dieses Verfahren wird durch den Parameter *Accumulate* in der PCF-Gruppe *TraceRoute* gesteuert.

## Aktivitätsinformationen mithilfe von Aktivitätsberichten aufzeichnen

Da eine Trace-Route-Nachricht über ein WS-Manager-Netz weitergeleitet wird, kann für jede Aktivität, die im Namen der Trace-Route-Nachricht ausgeführt wurde, ein Aktivitätsbericht generiert werden. Die Aktivitätsinformationen werden in der PCF-Gruppe *Activity* gespeichert. Für jede Aktivität, die im Namen einer Trace-Route-Nachricht ausgeführt wird, wird ein Aktivitätsbericht generiert, der eine *Activity*-PCF-Gruppe enthält. Die Aktivitätsaufzeichnung für Trace-Route-Nachrichten funktioniert genauso wie für jede andere Nachricht.

Aktivitätsberichte, die für Trace-Route-Nachrichten generiert werden, enthalten zusätzliche Aktivitätsinformationen im Vergleich zu denen, die für eine andere Nachricht generiert wurden. Die zusätzlichen Informationen werden in einer *TraceRoute*-PCF-Gruppe zurückgegeben. Die in der PCF-Gruppe von *TraceRoute* enthaltenen Informationen sind nur aus dem Zeitpunkt korrekt, zu dem der Aktivitätsbericht generiert wurde. Sie können die zusätzlichen Informationen verwenden, um die Reihenfolge der Aktivitäten zu bestimmen, die im Namen der Trace-Route-Nachricht ausgeführt werden.

## Informationen zu aufgezeichneten Aktivitäten werden abgefragt

Wenn eine Trace-Route-Nachricht das geplante Ziel erreicht hat oder verworfen wird, hängt die Methode, die Sie zum Anfordern der Aktivitätsinformationen verwenden, davon ab, wie diese Informationen aufgezeichnet wurden.

## Vorbereitende Schritte

Wenn Sie mit den Aktivitätsinformationen nicht vertraut sind, lesen Sie die Informationen in [„Wie Aktivitätsinformationen aufgezeichnet werden“](#) auf Seite 69.

## Informationen zu diesem Vorgang

Verwenden Sie die folgenden Methoden, um die Aktivitätsinformationen anzufordern, nachdem die Trace-Route-Nachricht das geplante Ziel erreicht hat oder gelöscht wurde:

## Prozedur

- Rufen Sie die Trace-Route-Nachricht ab.

Der Parameter *Deliver* in der PCF-Gruppe von *TraceRoute* steuert, ob eine Trace-Route-Nachricht bei der Ankunft in die Zielwarteschlange gestellt wird, oder ob sie verworfen wird. Wenn die Trace-Route-Nachricht an die Zielwarteschlange zugestellt wird, können Sie die Trace-Route-Nachricht aus dieser Warteschlange abrufen. Anschließend können Sie mit der IBM MQ-Anwendung für die Routenanzeige die Aktivitätsinformationen anzeigen.

Wenn Sie anfordern möchten, dass die Aktivitätsinformationen in den Nachrichtendaten einer Trace-Route-Nachricht akkumuliert werden, setzen Sie den Parameter *Accumulate* in der PCF-Gruppe von *TraceRoute* auf `MQROUTE_ACCUMULATE_IN_MSG`.

- Verwenden Sie eine Trace-Route-Antwortnachricht.

Wenn eine Trace-Route-Nachricht ihre Zieladresse erreicht oder die Trace-Route-Nachricht nicht weiter in einem WS-Manager-Netz weitergeleitet werden kann, kann eine Antwortnachricht für die Trace-Route generiert werden. Eine Antwortnachricht auf Trace-Route enthält ein Duplikat aller Aktivitätsinformationen aus der Trace-Route-Nachricht und wird entweder an eine angegebene Warteschlange für Antwortnachrichten oder an die Systemwarteschlange `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE` übergeben. Mit der IBM MQ-Anwendung für die Routenanzeige können Sie die Aktivitätsinformationen anzeigen.

Um eine Trace-Route-Antwortnachricht anzufordern, setzen Sie den Parameter *Accumulate* in der PCF-Gruppe von *TraceRoute* auf `MQROUTE_ACCUMULATE_AND_REPLY`.

- Verwenden Sie Aktivitätsberichte.

Wenn Aktivitätsberichte für eine Trace-Route-Nachricht generiert werden, müssen Sie die Aktivitätsberichte suchen, bevor Sie die Aktivitätsinformationen anfordern können. Um die Reihenfolge der Aktivitäten zu bestimmen, müssen Sie die Aktivitätsberichte sortieren.

## **Trace für Trace-Route-Messaging steuern**

Aktivieren Sie das Trace-Route-Messaging auf WS-Managerebene, so dass Anwendungen im Geltungsbereich dieses Warteschlangenmanagers Aktivitätsinformationen in eine Trace-Route-Nachricht schreiben können. Wenn Sie ein vollständiges WS-Manager-Netz aktivieren möchten, aktivieren Sie jeden Warteschlangenmanager im Netz für Trace-Route-Messaging. Wenn Sie mehr WS-Manager aktivieren, werden weitere Aktivitätsberichte generiert.

### **Vorbereitende Schritte**

Wenn Sie Aktivitätsberichte zum Aufzeichnen von Aktivitätsinformationen für eine Trace-Route-Nachricht verwenden, lesen Sie die Informationen in [„Aktivitätsaufzeichnung steuern“](#) auf Seite 64.

### **Informationen zu diesem Vorgang**

Führen Sie die folgenden Schritte aus, um Aktivitätsinformationen für eine Trace-Route-Nachricht zu erfassen, wenn sie über einen Warteschlangenmanager weitergeleitet wird:

### **Prozedur**

- Definieren Sie, wie Aktivitätsinformationen für die Trace-Route-Nachricht aufgezeichnet werden sollen.  
Siehe [„Trace-Route-Nachricht generieren und konfigurieren“](#) auf Seite 73.
- Wenn Sie Aktivitätsinformationen in der Trace-Route-Nachricht akkumulieren möchten, müssen Sie sicherstellen, dass der Warteschlangenmanager für die Trace-Route-Nachrichtenübertragung aktiviert ist.
- Wenn Sie Aktivitätsinformationen in der Trace-Route-Nachricht akkumulieren möchten, stellen Sie sicher, dass Anwendungen, die Aktivitäten in der Trace-Route-Nachricht ausführen, in der Lage sind, Aktivitätsinformationen in die Nachrichtendaten der Trace-Route-Nachricht zu schreiben.

### **Zugehörige Konzepte**

[„Trace-Route-Nachricht generieren und konfigurieren“](#) auf Seite 73

Eine Trace-Route-Nachricht enthält bestimmte Nachrichtendeskriptoren und Nachrichtendatenteile. Zum Generieren einer Trace-Route-Nachricht erstellen Sie die Nachricht manuell oder verwenden die IBM MQ-Anwendung für die Routenanzeige.

### **Zugehörige Tasks**

[„Aktivitätsaufzeichnung steuern“](#) auf Seite 64

Aktivieren Sie die Aktivitätsaufzeichnung auf WS-Managerebene. Wenn Sie ein vollständiges WS-Manager-Netz aktivieren möchten, aktivieren Sie jeden Warteschlangenmanager im Netz für die Aktivitätsaufzeichnung einzeln. Wenn Sie mehr WS-Manager aktivieren, werden weitere Aktivitätsberichte generiert.

#### *WS-Manager für Trace-Route-Messaging aktivieren*

Um zu steuern, ob Warteschlangenmanager für Trace-Route-Messaging aktiviert oder inaktiviert sind, verwenden Sie das WS-Manager-Attribut ROUTEREC.

Verwenden Sie den MQSC-Befehl ALTER QMGR, und geben Sie dabei den Parameter ROUTEREC an, um den Wert des Warteschlangenmanagerattributs zu ändern. Folgende Werte sind möglich:

### **MSG**

Der Warteschlangenmanager ist für Trace-Route-Messaging aktiviert. Anwendungen im Geltungsbereich des Warteschlangenmanagers können Aktivitätsinformationen in die Trace-Route-Nachricht schreiben.

Wenn der Parameter *Accumulate* in der *TraceRoute* -PCF-Gruppe als MQROUTE\_ACCUMULATE\_AND\_REPLY festgelegt ist und die nächste Aktivität in der Trace-Route-Nachricht ausgeführt werden soll, gehen Sie wie folgt vor:

- Ist ein Löszeitpunkt
- ist eine in eine lokale Warteschlange (Zielwarteschlange oder Warteschlange für dead-letter)

- führt dazu, dass die Gesamtzahl der Aktivitäten, die in der Trace-Route-Nachricht ausgeführt werden, den Wert des Parameters *MaxActivities* in der PCF-Gruppe *TraceRoute* überschreitet.

Es wird eine Antwortnachricht für den Trace-Leitweg generiert und an die Warteschlange für Antwortnachrichten übergeben, die im Nachrichtendeskriptor der Trace-Route-Nachricht angegeben ist.

## WARTESCHLANGE

Der Warteschlangenmanager ist für Trace-Route-Messaging aktiviert. Anwendungen im Geltungsbereich des Warteschlangenmanagers können Aktivitätsinformationen in die Trace-Route-Nachricht schreiben.

Wenn der Parameter *Accumulate* in der *TraceRoute* -PCF-Gruppe als *MQRROUTE\_ACCUMULATE\_AND\_REPLY* festgelegt ist und die nächste Aktivität in der Trace-Route-Nachricht ausgeführt werden soll, gehen Sie wie folgt vor:

- Ist ein Löszeitpunkt
- ist eine in eine lokale Warteschlange (Zielwarteschlange oder Warteschlange für dead-letter)
- führt dazu, dass die Gesamtzahl der Aktivitäten, die in der Trace-Route-Nachricht ausgeführt werden, den Wert des Parameters *MaxActivities* in der PCF-Gruppe *TraceRoute* überschreitet.

Es wird eine Antwortnachricht für den Trace-Leitweg generiert und an die lokale Systemwarteschlange *SYSTEM.ADMIN.TRACE.ROUTE.QUEUE* übergeben.

## INAKTIVIERT

Der WS-Manager ist für die Trace-Route-Nachrichtenübertragung inaktiviert. Die Aktivitätsinformationen werden in der Trace-Route-Nachricht nicht kumuliert, die *TraceRoute* -PCF-Gruppe kann jedoch im Geltungsbereich dieses Warteschlangenmanagers aktualisiert werden.

Verwenden Sie beispielsweise den folgenden MQSC-Befehl, um einen Warteschlangenmanager für Trace-Route-Messaging zu inaktivieren:

```
ALTER QMGR ROUTEREC(DISABLED)
```

**Hinweis:** Wenn Sie das WS-Manager-Attribut *ROUTEREC* ändern, erkennt ein laufender MCA die Änderung erst, wenn der Kanal erneut gestartet wird.

### Anwendungen für Trace-Route-Messaging aktivieren

Um das Trace-Route-Messaging für eine Benutzeranwendung zu aktivieren, basieren Sie auf Ihrem Algorithmus auf dem Algorithmus, der von Nachrichtenkanalagenten (MCAs) verwendet wird.

## Vorbereitende Schritte

Wenn Sie mit dem Format einer Trace-Route-Nachricht nicht vertraut sind, lesen Sie die Informationen in [„Trace-Route-Nachrichtenreferenz“](#) auf Seite 128.

## Informationen zu diesem Vorgang

Nachrichtenkanalagenten (MCAs) werden für Trace-Route-Messaging aktiviert. Wenn Sie eine Benutzeranwendung für das Trace-Route-Messaging aktivieren möchten, verwenden Sie die folgenden Schritte aus dem Algorithmus, der von MCAs verwendet wird:

## Vorgehensweise

1. Stellen Sie fest, ob die Nachricht, die verarbeitet wird, eine Trace-Route-Nachricht ist.  
Wenn die Nachricht nicht mit dem Format einer Trace-Route-Nachricht übereinstimmt, wird die Nachricht nicht als Trace-Route-Nachricht verarbeitet.
2. Bestimmen Sie, ob Aktivitätsinformationen aufgezeichnet werden sollen.  
Wenn der Detaillierungsgrad der ausgeführten Aktivität nicht kleiner ist als die vom Parameter *Detail* angegebene Detaillierungsebene, werden die Aktivitätsinformationen unter bestimmten Umständen aufgezeichnet. Diese Informationen werden nur aufgezeichnet, wenn die Trace-Route-Nachrichtenauf-

zeichnung anfordert, und der Warteschlangenmanager für Trace-Route-Messaging aktiviert ist, oder wenn die Trace-Route-Nachricht einen Aktivitätsbericht anfordert und der Warteschlangenmanager für die Aktivitätsaufzeichnung aktiviert ist.

- Wenn Aktivitätsinformationen aufgezeichnet werden sollen, erhöhen Sie den Parameter *RecordedActivities*.
  - Wenn Aktivitätsinformationen nicht aufgezeichnet werden sollen, erhöhen Sie den Parameter *UnrecordedActivities*.
3. Stellen Sie fest, ob die Gesamtzahl der Aktivitäten, die in der Trace-Route-Nachricht ausgeführt werden, den Wert des Parameters *MaxActivities* überschreitet.

Die Gesamtzahl der Aktivitäten ist die Summe aus *RecordedActivities*, *UnrecordedActivities* und *DiscontinuityCount*.

Wenn die Gesamtzahl der Aktivitäten *MaxActivities* überschreitet, weisen Sie die Nachricht mit dem Feedback MQFB\_MAX\_ACTIVITIES zurück.

4. Wenn der Wert von *Accumulate* als MQROUTE\_ACCUMULATE\_IN\_MSG oder MQROUTE\_ACCUMULATE\_AND\_REPLY festgelegt ist und der Warteschlangenmanager für Trace-Route-Messaging aktiviert ist, schreiben Sie eine Aktivität-PCF-Gruppe an das Ende des PCF-Blocks in den Nachrichtendaten einer Trace-Route-Nachricht.
5. Liefern Sie die Trace-Route-Nachricht an eine lokale Warteschlange.
- Wenn der Parameter, *Deliver*, als MQROUTE\_DELIVER\_NO angegeben wird, wird die Trace-Route-Nachricht mit dem Feedback MQFB\_NOT\_DELIVERED zurückgewiesen.
  - Wenn der Parameter *Deliver* als MQROUTE\_DELIVER\_YES angegeben ist, übergeben Sie die Trace-Route-Nachricht an die lokale Warteschlange.
6. Eine Trace-Route-Antwortnachricht generieren, wenn alle folgenden Bedingungen erfüllt sind:
- Die Trace-Route-Nachricht wurde an eine lokale Warteschlange gesendet oder zurückgewiesen.
  - Der Wert des Parameters (*Accumulate*) lautet MQROUTE\_ACCUMULATE\_AND\_REPLY.
  - Der Warteschlangenmanager ist für Trace-Route-Messaging aktiviert.

Die Trace-Route-Antwortnachricht wird in die Warteschlange gestellt, die durch das Attribut ROUTE-REC des Warteschlangenmanagers festgelegt wird.

7. Wenn die Trace-Route-Nachricht einen Aktivitätsbericht angefordert hat und der WS-Manager für die Aktivitätsaufzeichnung aktiviert ist, generieren Sie einen Aktivitätsbericht.

Der Aktivitätsbericht wird in die Warteschlange gestellt, die durch das Warteschlangenmanagerattribut ACTIVREC bestimmt wird.

### **Trace-Route-Nachricht generieren und konfigurieren**

Eine Trace-Route-Nachricht enthält bestimmte Nachrichtendeskriptoren und Nachrichtendatenteile. Zum Generieren einer Trace-Route-Nachricht erstellen Sie die Nachricht manuell oder verwenden die IBM MQ-Anwendung für die Routenanzeige.

Eine Trace-Route-Nachricht besteht aus den folgenden Teilen:

#### **Nachrichtendeskriptor**

Eine MQMD-Struktur, wobei das Feld *Format* auf MQFMT\_ADMIN oder MQFMT\_EMBEDDED\_PCF gesetzt ist.

#### **Nachrichtendaten**

Eine der folgenden Kombinationen:

- Ein PCF-Header (MQCFH) und Trace-Route-Nachrichtendaten, wenn *Format* auf MQFMT\_ADMIN gesetzt ist
- Ein eingebetteter PCF-Header (MQEPH), Trace-Route-Nachrichtendaten und zusätzliche benutzerdefinierte Nachrichtendaten, wenn *Format* auf MQFMT\_EMBEDDED\_PCF gesetzt ist.

Die Trace-Route-Nachrichtendaten bestehen aus der *TraceRoute* -PCF-Gruppe und einer oder mehreren *Activity* -PCF-Gruppen.

## Manuelle Generierung

Wenn Sie eine Trace-Route-Nachricht manuell generieren, ist eine *Activity* -PCF-Gruppe nicht erforderlich. *Activity* -PCF-Gruppen werden in die Nachrichtendaten der Trace-Route-Nachricht geschrieben, wenn eine MCA oder eine benutzerdefinierte Anwendung eine Aktivität in ihrem Namen ausführt.

## Die IBM MQ-Anwendung für die Routenanzeige

Mit der IBM MQ-Anwendung für die Routenanzeige (**dspmqrte**) können Sie eine Trace-Route-Nachricht konfigurieren, generieren und in ein Warteschlangenmanagernetz stellen. Setzen Sie den Parameter *Format* im Nachrichtendeskriptor auf MQFMT\_ADMIN. Sie können der von der IBM MQ-Anwendung für die Routenanzeige Trace-Route-Nachricht keine Benutzerdaten hinzufügen.

**Einschränkung:** dspmqrte kann nicht auf Warteschlangenmanagern vor IBM WebSphere MQ 6.0 oder auf IBM MQ for z/OS-Warteschlangenmanagern ausgegeben werden. Wenn der erste Warteschlangenmanager, durch den die Trace-Route-Nachricht geleitet werden soll, ein Warteschlangenmanager dieses Typs sein soll, stellen Sie die Verbindung zum Warteschlangenmanager als Client der IBM WebSphere MQ 6.0 oder höher mithilfe des optionalen Parameters -c her.

### *Ursprüngliche Nachricht imitieren*

Wenn Sie eine Trace-Route-Nachricht verwenden, um die Route zu ermitteln, die eine andere Nachricht über ein WS-Manager-Netz hat, wird die ursprüngliche Nachricht durch eine Trace-Route-Nachricht, die größer ist als die Wahrscheinlichkeit, dass die Trace-Route-Nachricht die gleiche Route wie die ursprüngliche Nachricht verfolgen wird, besser nachgebildet.

Die folgenden Nachrichtenmerkmale können sich darauf auswirken, dass eine Nachricht in einem Warteschlangenmanagernetz weitergeleitet wird:

### **Priority**

Die Priorität kann im Nachrichtendeskriptor der Nachricht angegeben werden.

### **Permanenz**

Die Persistenz kann im Nachrichtendeskriptor der Nachricht angegeben werden.

### **Ablauf**

Der Verfall kann im Nachrichtendeskriptor der Nachricht angegeben werden.

### **Berichtsoptionen**

Berichtsoptionen können im Nachrichtendeskriptor der Nachricht angegeben werden.

### **Nachrichtengröße**

Um die Größe einer Nachricht zu imitieren, können zusätzliche Daten in die Nachrichtendaten der Nachricht geschrieben werden. Zu diesem Zweck können zusätzliche Nachrichtendaten bedeutungslos sein.

**Tipp:** Die IBM MQ-Anwendung für die Routenanzeige kann keine Nachrichtengröße angeben.

### **Nachrichtendaten**

Einige WS-Manager-Netze verwenden das inhaltsbasierte Routing, um festzustellen, wohin die Nachrichten weitergeleitet werden. In diesen Fällen müssen die Nachrichtendaten der Trace-Route-Nachricht so geschrieben werden, dass sie die Nachrichtendaten der ursprünglichen Nachricht imitieren.

**Tipp:** Die IBM MQ-Anwendung für die Routenanzeige kann keine Nachrichtendaten angeben.

### *PCF-Gruppe "TraceRoute"*

Die Attribute in der PCF-Gruppe von *TraceRoute* steuern das Verhalten einer Trace-Route-Nachricht. Die *TraceRoute* -PCF-Gruppe befindet sich in den Nachrichtendaten jeder Trace-Route-Nachricht.

In der folgenden Tabelle sind die Parameter in der Gruppe *TraceRoute* aufgelistet, die von einem MCA erkannt werden. Weitere Parameter können hinzugefügt werden, falls benutzerdefinierte Anwendungen für ihre Erkennung geschrieben werden (siehe hierzu [„Zusätzliche Aktivitätsinformationen“](#) auf Seite 80).

Tabelle 12. PCF-Gruppe "TraceRoute"

Parameter	Typ
TraceRoute	MQCFGR
Detail	MQCFIN
RecordedActivities	MQCFIN
UnrecordedActivities	MQCFIN
DiscontinuityCount	MQCFIN
MaxActivities	MQCFIN
Kumulieren	MQCFIN
Vorwärts	MQCFIN
Deliver	MQCFIN

Im Folgenden werden die Beschreibungen der einzelnen Parameter in der *TraceRoute* -PCF-Gruppe beschrieben:

### **Detail**

Gibt die Detailstufe der Aktivitätsinformationen an, die aufgezeichnet werden sollen. Folgende Werte sind möglich:

#### **MQROUTE\_DETAIL\_LOW**

Es werden nur Aktivitäten aufgezeichnet, die von der Benutzeranwendung ausgeführt werden.

#### **MQROUTE\_DETAIL\_MEDIUM**

Die in *MQROUTE\_DETAIL\_LOW* angegebenen Aktivitäten sollten aufgezeichnet werden. Darüber hinaus werden Aktivitäten, die von MCAs ausgeführt werden, aufgezeichnet.

#### **MQROUTE\_DETAIL\_HIGH**

Die in '*MQROUTE\_DETAIL\_LOW*' und '*MQROUTE\_DETAIL\_MEDIUM*' angegebenen Aktivitäten sollten aufgezeichnet werden. MCAs zeichnen keine weiteren Aktivitätsinformationen auf dieser Detaillierungsebene auf. Diese Option ist nur für Benutzeranwendungen verfügbar, in denen weitere Aktivitätsinformationen aufgezeichnet werden sollen. Wenn eine Benutzeranwendung beispielsweise die Route einer Nachricht durch die Berücksichtigung bestimmter Nachrichtenmerkmale bestimmt, könnte die Information über die Routing-Logik in diese Detaillierungsebene eingeschlossen werden.

### **RecordedActivities**

Gibt die Anzahl der aufgezeichneten Aktivitäten an, die im Namen der Trace-Route-Nachricht ausgeführt wurden. Eine Aktivität wird als aufgezeichnet betrachtet, wenn Informationen zu ihr in die Trace-Route-Nachricht geschrieben wurden oder wenn ein Aktivitätsbericht generiert wurde. Für jede aufgezeichnete Aktivität wird *RecordedActivities* um eins erhöht.

### **UnrecordedActivities**

Gibt die Anzahl der nicht aufgezeichneten Aktivitäten an, die im Namen der Trace-Route-Nachricht ausgeführt werden. Eine Aktivität wird als nicht erfasst betrachtet, wenn eine Anwendung, die für das Tracing-Route-Messaging aktiviert ist, weder akkumuliert wird noch die zugehörigen Aktivitätsinformationen in einen Aktivitätsbericht schreibt.

Eine Aktivität, die im Namen einer Trace-Route-Nachricht ausgeführt wird, wird unter den folgenden Umständen nicht aufgezeichnet:

- Der Detaillierungsgrad der ausgeführten Aktivität ist kleiner als die Detaillierungsebene, die durch den Parameter *Detail* angegeben wird.
- Die Trace-Route-Nachricht fordert einen Aktivitätsbericht an, aber keine Akkumulation, und der WS-Manager ist nicht für die Aktivitätsaufzeichnung aktiviert.
- Die Trace-Route-Nachricht fordert eine Akkumulation an, aber keinen Aktivitätsbericht, und der Warteschlangenmanager ist für Trace-Route-Messaging nicht aktiviert.
- Die Trace-Route-Nachricht fordert sowohl eine Akkumulation als auch einen Aktivitätsbericht an, und der Warteschlangenmanager ist nicht für die Aufzeichnung von Aktivitäten und die Weiterleitung von Trace-Nachrichten aktiviert.

- Die Trace-Route-Nachricht fordert weder eine Akkumulation noch einen Aktivitätsbericht an.

Für jede nicht aufgezeichnete Aktivität wird der Parameter *UnrecordedActivities* in Inkrementen um eins erhöht.

### **DiscontinuityCount**

Gibt an, wie oft die Trace-Route-Nachricht über einen Warteschlangenmanager mit Anwendungen weitergeleitet wurde, die für die Trace-Route-Nachrichtenübertragung nicht aktiviert waren. Dieser Wert wird vom WS-Manager inkrementiert. Wenn dieser Wert größer als 0 ist, kann nur ein Teilnachrichtenweg ermittelt werden.

### **MaxActivities**

Gibt die maximale Anzahl der Aktivitäten an, die im Namen der Trace-Route-Nachricht ausgeführt werden können.

Die Gesamtzahl der Aktivitäten ist die Summe aus *RecordedActivities*, *UnrecordedActivities* und *DiscontinuityCount*. Die Gesamtzahl der Aktivitäten darf den Wert von *MaxActivities* nicht überschreiten.

Der Wert von *MaxActivities* kann wie folgt sein:

#### **Eine positive Ganzzahl**

Die maximale Anzahl der Aktivitäten.

Wenn die maximale Anzahl an Aktivitäten überschritten wird, wird die Trace-Route-Nachricht mit dem Feedback MQFB\_MAX\_ACTIVITIES zurückgewiesen. Dadurch kann verhindert werden, dass die Trace-Route-Nachricht unbegrenzt weitergeleitet wird, wenn sie in einer Endlosschleife abgefangen wird.

#### **MQROUTE\_UNLIMITED\_ACTIVITIES**

Im Namen der Trace-Route-Nachricht kann eine unbegrenzte Anzahl von Aktivitäten ausgeführt werden.

### **Accumulate**

Gibt die Methode an, mit der Aktivitätsinformationen akkumuliert werden. Folgende Werte sind möglich:

#### **MQROUTE\_ACCUMULATE\_IN\_MSG**

Wenn der Warteschlangenmanager für Trace-Route-Messaging aktiviert ist, werden die Aktivitätsinformationen in den Nachrichtendaten der Trace-Route-Nachricht akkumuliert.

Wenn dieser Wert angegeben wird, besteht die Trace-Route-Nachrichtendaten aus den folgenden Angaben:

- Die *TraceRoute* -PCF-Gruppe.
- Null oder mehr *Activity* -PCF-Gruppen.

#### **MQROUTE\_ACCUMULATE\_AND\_REPLY**

Wenn der Warteschlangenmanager für die Trace-Route-Nachrichtenübertragung aktiviert ist, werden die Aktivitätsinformationen in den Nachrichtendaten der Trace-Route-Nachricht akkumuliert und eine Trace-Route-Antwortnachricht wird generiert, wenn eine der folgenden Ereignisse eintritt:

- Die Trace-Route-Nachricht wird von einem Warteschlangenmanager von IBM WebSphere MQ 6 (oder höher) gelöscht.
- Die Trace-Route-Nachricht wird von einem Warteschlangenmanager von IBM WebSphere MQ 6 (oder höher) in eine lokale Warteschlange (Zielwarteschlange oder Warteschlange für nicht zustellbare Nachrichten) eingereicht.
- Die Anzahl der Aktivitäten, die in der Trace-Route-Nachricht ausgeführt werden, überschreitet den Wert von *MaxActivities*.

Wenn dieser Wert angegeben wird, besteht die Trace-Route-Nachrichtendaten aus den folgenden Angaben:

- Die *TraceRoute* -PCF-Gruppe.
- Null oder mehr *Activity* -PCF-Gruppen.

### **MQROUTE\_ACCUMULATE\_NONE**

Die Aktivitätsinformationen werden in den Nachrichtendaten der Trace-Route-Nachricht nicht kumuliert.

Wenn dieser Wert angegeben wird, besteht die Trace-Route-Nachrichtendaten aus den folgenden Angaben:

- Die *TraceRoute* -PCF-Gruppe.

### **Forward**

Gibt an, an welche Position eine Trace-Route-Nachricht weitergeleitet werden kann. Folgende Werte sind möglich:

### **MQROUTE\_FORWARD\_IF\_SUPPORTED**

Die Trace-Route-Nachricht wird nur an WS-Manager weitergeleitet, die den Wert des Parameters *Deliver* aus der Gruppe *TraceRoute* berücksichtigen.

### **MQROUTE\_FORWARD\_ALL**

Die Trace-Route-Nachricht wird an jeden WS-Manager weitergeleitet, unabhängig davon, ob der Wert des Parameters *Deliver* berücksichtigt wird.

Warteschlangenmanager verwenden den folgenden Algorithmus, wenn Sie festlegen, ob eine Trace-Route-Nachricht an einen fernen Warteschlangenmanager weitergeleitet werden soll:

1. Stellen Sie fest, ob der ferne Warteschlangenmanager in der Lage ist, Trace-Route-Messaging zu unterstützen.
  - Wenn der ferne Warteschlangenmanager in der Lage ist, Trace-Route-Messaging zu unterstützen, fährt der Algorithmus mit Schritt „4“ auf Seite 77 fort.
  - Wenn der ferne Warteschlangenmanager nicht in der Lage ist, Tracing-Route-Messaging zu unterstützen, fährt der Algorithmus mit Schritt „2“ auf Seite 77 fort.
2. Stellen Sie fest, ob der Parameter *Deliver* aus der Gruppe *TraceRoute* alle nicht erkannten Zustelloptionen in der Bitmaske `MQROUTE_DELIVER_REJ_UNSUP_MASK` enthält.
  - Wenn nicht erkannte Zustelloptionen gefunden werden, wird die Trace-Route-Nachricht mit dem Feedback `MQFB_UNSUPPORTED_DELIVERY` zurückgewiesen.
  - Wenn keine nicht erkannten Zustelloptionen gefunden werden, fährt der Algorithmus mit Schritt „3“ auf Seite 77 fort.
3. Ermitteln Sie den Wert des Parameters *Deliver* aus der PCF-Gruppe *TraceRoute* in der Trace-Route-Nachricht.
  - Wenn *Deliver* als `MQROUTE_DELIVER_YES` angegeben wird, wird die Trace-Route-Nachricht an den fernen WS-Manager weitergeleitet.
  - Wenn *Deliver* als `MQROUTE_DELIVER_NO` angegeben wird, wird der Algorithmus mit Schritt „4“ auf Seite 77 fortgesetzt.
4. Stellen Sie fest, ob der Parameter *Forward* aus der Gruppe *TraceRoute* alle nicht erkannten Weiterleitungsoptionen in der Bitmaske `MQROUTE_FORWARDING_REJ_UNSUP_MASK` enthält.
  - Wenn nicht erkannte Weiterleitungsoptionen gefunden werden, wird die Trace-Route-Nachricht mit dem Feedback `MQFB_UNSUPPORTED_FORWARDING` zurückgewiesen.
  - Wenn keine nicht erkannten Weiterleitungsoptionen gefunden werden, fährt der Algorithmus mit Schritt „5“ auf Seite 77 fort.
5. Ermitteln Sie den Wert des Parameters *Forward* aus der PCF-Gruppe *TraceRoute* in der Trace-Route-Nachricht.
  - Wenn *Forward* als `MQROUTE_FORWARD_IF_SUPPORTED` angegeben wird, wird die Trace-Route-Nachricht mit dem Feedback `MQFB_NOT_FORWARDED` zurückgewiesen.
  - Wenn *Forward* als `MQROUTE_FORWARD_ALL` angegeben wird, kann die Trace-Route-Nachricht an den fernen WS-Manager weitergeleitet werden.

## **Deliver**

Gibt die Aktion an, die ausgeführt werden soll, wenn die Trace-Route-Nachricht die beabsichtigte Zieladresse erreicht. Benutzergeschriebene Anwendungen müssen dieses Attribut überprüfen, bevor eine Trace-Route-Nachricht in die Zielwarteschlange gestellt wird. Folgende Werte sind möglich:

### **MQROUTE\_DELIVER\_YES**

Bei der Ankunft wird die Trace-Route-Nachricht in die Zielwarteschlange gestellt. Jede Anwendung, die eine Operation 'get' in der Zielwarteschlange ausführt, kann die Trace-Route-Nachricht abrufen.

### **MQROUTE\_DELIVER\_NO**

Bei der Ankunft wird die Trace-Route-Nachricht nicht an die Zielwarteschlange zugestellt. Die Nachricht wird entsprechend ihren Berichtsoptionen verarbeitet.

## **Eine allgemeine Warteschlange für Antwortnachrichten für Tracetrouten konfigurieren**

Um die Positionen der Trace-Route-Antwortnachrichten zu bestimmen, die sich auf eine bestimmte Nachricht beziehen, wenn die Berichte an die lokale Systemwarteschlange zugestellt werden, ist es effizienter, eine gemeinsame Warteschlange auf einem einzigen Knoten zu verwenden.

## **Vorbereitende Schritte**

Definieren Sie den Parameter **ROUTEREC**, um den Warteschlangenmanager für Trace-Route-Messaging zu aktivieren und anzugeben, dass alle generierten Trace-Route-Antwortnachrichten an die lokale Systemwarteschlange `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE` übergeben werden.

## **Informationen zu diesem Vorgang**

Wenn eine Reihe von Warteschlangenmanagern in einem Warteschlangenmanagernetzwerk so eingestellt sind, dass sie Tracing-Route-Antwortnachrichten an die lokale Systemwarteschlange zustellen, kann es zeitaufwendig sein, die Positionen der Trace-Route-Antwortnachrichten zu ermitteln, die sich auf eine bestimmte Nachricht beziehen. Alternativ können Sie auch einen einzigen Knoten verwenden, der ein Warteschlangenmanager ist, der eine gemeinsame Warteschlange hostet. Alle Warteschlangenmanager in einem WS-Manager-Netz können Antwortnachrichten für die Traceroute an diese allgemeine Warteschlange zustellen. Der Vorteil der Verwendung einer allgemeinen Warteschlange besteht darin, dass Warteschlangenmanager keine Trace-Route-Antwortnachrichten an die in einer Nachricht angegebene Warteschlange für Antwortnachrichten bereitstellen müssen und wenn die Positionen der Antwortnachrichten der Traceroute in Zusammenhang mit einer Nachricht bestimmt werden, wird nur eine Warteschlange abgefragt.

Führen Sie die folgenden Schritte aus, um eine allgemeine Warteschlange zu konfigurieren:

## **Vorgehensweise**

1. Wählen Sie einen Warteschlangenmanager als einen einzigen Knoten aus oder definieren Sie diesen.
2. Wählen Sie auf dem einzelnen Knoten eine Warteschlange für die Verwendung als allgemeine Warteschlange aus oder definieren Sie sie.
3. Auf allen WS-Managern, die Tracing-Route-Antwortnachrichten an die allgemeine Warteschlange weiterleiten, definieren Sie die lokale Systemwarteschlange `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE` als ferne Warteschlangendefinition neu.
  - a) Geben Sie den Namen des einzelnen Knotens als Name des fernen Warteschlangenmanagers an.
  - b) Geben Sie den Namen der allgemeinen Warteschlange als Namen der fernen Warteschlange an.

## **Erfassung und Verwendung von aufgezeichneten Informationen**

Verwenden Sie eine der folgenden Methoden, um aufgezeichnete Aktivitätsinformationen für eine Trace-Route-Nachricht anzufordern.

Beachten Sie, dass die Umstände, unter denen die Aktivitätsinformationen nicht erfasst werden, auch für Trace-Route-Antwortnachrichten gelten.

Die Aktivitätsinformationen werden nicht aufgezeichnet, wenn eine Trace-Route-Nachricht von einem Warteschlangenmanager verarbeitet wird, der sowohl für die Aktivitätsaufzeichnung als auch für die Trace-Route-Nachrichtenübertragung inaktiviert ist.

*Informationen aus Trace-Route-Antwortnachrichten werden abgefragt.*

Informationen zum Anfordern von Aktivitätsinformationen finden Sie in der Antwortnachricht der Trace-Route. Anschließend rufen Sie die Nachricht ab und analysieren die Aktivitätsinformationen.

## Informationen zu diesem Vorgang

Sie können Aktivitätsinformationen aus einer Antwortnachricht für Trace-Routen nur dann anfordern, wenn Sie die Position der Antwortnachricht für die Trace-Route kennen. Suchen Sie die Nachricht und verarbeiten Sie die Aktivitätsinformationen wie folgt:

### Vorgehensweise

1. Überprüfen Sie die Warteschlange für Antwortnachrichten, die im Nachrichtendeskriptor der Trace-Route-Nachricht angegeben wurde. Wenn die Antwortnachricht für die Traceroute nicht in der Warteschlange für Antwortnachrichten enthalten ist, überprüfen Sie die folgenden Positionen:
  - Die lokale Systemwarteschlange SYSTEM.ADMIN.TRACE.ROUTE.QUEUE auf dem Ziel-WS-Manager der Trace-Route-Nachricht
  - Die allgemeine Warteschlange, wenn Sie eine gemeinsame Warteschlange für Antwortnachrichten für Tracerouten konfiguriert haben.
  - Die lokale Systemwarteschlange SYSTEM.ADMIN.TRACE.ROUTE.QUEUE auf einem beliebigen anderen Warteschlangenmanager im WS-Manager-Netz, die auftreten kann, wenn die Trace-Route-Nachricht in eine Warteschlange für einen Deadbuchstaben gestellt wurde oder die maximale Anzahl an Aktivitäten überschritten wurde.
2. Antwortnachricht für Trace-Route abrufen
3. Verwenden Sie die IBM MQ-Anwendung zur Routenanzeige, um die aufgezeichneten Aktivitätsinformationen anzuzeigen.
4. Die Aktivitätsinformationen studieren und die benötigten Informationen abrufen

*Informationen aus Trace-Route-Nachrichten werden abgefragt.*

Informationen zum Anfordern von Aktivitätsinformationen finden Sie in der Trace-Route-Nachricht, die über die entsprechenden Parameter in der PCF-Gruppe von *TraceRoute* verfügen muss. Anschließend rufen Sie die Nachricht ab und analysieren die Aktivitätsinformationen.

## Informationen zu diesem Vorgang

Sie können Aktivitätsinformationen aus einer Trace-Route-Nachricht nur dann anfordern, wenn Sie die Position der Trace-Route-Nachricht kennen und den Parameter *Accumulate* in der *TraceRoute* -PCF-Gruppe entweder als *MQROUTE\_ACCUMULATE\_IN\_MSG* oder als *MQROUTE\_ACCUMULATE\_AND\_REPLY* angegeben hat.

Damit die Trace-Route-Nachricht an die Zielwarteschlange zugestellt werden kann, muss der Parameter *Deliver* in der *TraceRoute* -PCF-Gruppe als *MQROUTE\_DELIVER\_YES* angegeben werden.

### Vorgehensweise

1. Überprüfen Sie die Zielwarteschlange. Wenn die Trace-Route-Nachricht nicht in der Zielwarteschlange enthalten ist, können Sie versuchen, die Trace-Route-Nachricht unter Verwendung einer Trace-Route-Nachricht zu lokalisieren, die für die Aktivitätsaufzeichnung aktiviert ist. Versuchen Sie mit den generierten Aktivitätsberichten, die letzte bekannte Position der Trace-Route-Nachricht zu ermitteln.
2. Trace-Route-Nachricht abrufen
3. Verwenden Sie die IBM MQ-Anwendung zur Routenanzeige, um die aufgezeichneten Aktivitätsinformationen anzuzeigen.
4. Die Aktivitätsinformationen studieren und die benötigten Informationen abrufen

### Informationen aus Aktivitätsberichten werden abgefragt

Informationen zum Anfordern von Aktivitätsinformationen finden Sie im Aktivitätsbericht, der über die Berichtsoption im Nachrichtendeskriptor verfügen muss. Anschließend rufen Sie den Aktivitätsbericht ab und analysieren die Aktivitätsinformationen.

### Informationen zu diesem Vorgang

Sie können Aktivitätsinformationen aus einem Aktivitätsbericht nur dann anfordern, wenn Sie wissen, wo sich der Aktivitätsbericht und die Berichtsoption MQRO\_ACTIVITY im Nachrichtendeskriptor der Trace-Route-Nachricht angegeben haben.

### Vorgehensweise

1. Suchen und ordnen Sie die Aktivitätsberichte, die für eine Trace-Route-Nachricht generiert wurden.  
Wenn Sie die Aktivitätsberichte gefunden haben, können Sie diese manuell sortieren oder die IBM MQ-Anwendung für die Routenanzeige verwenden, um die Aktivitätsinformationen zu sortieren und automatisch anzuzeigen.
2. Die Aktivitätsinformationen studieren und die benötigten Informationen abrufen

### Zusätzliche Aktivitätsinformationen

Da eine Trace-Route-Nachricht über ein WS-Manager-Netz weitergeleitet wird, können Benutzeranwendungen zusätzliche Informationen aufzeichnen, indem sie einen oder mehrere zusätzliche PCF-Parameter enthalten, wenn die *Activity*-Gruppe in die Nachrichtendaten der Trace-Route-Nachricht oder des Aktivitätsberichts geschrieben wird.

Zusätzliche Aktivitätsinformationen können Systemadministratoren helfen, die Route zu identifizieren, die von einer Trace-Route-Nachricht genommen wurde, oder warum diese Route ausgeführt wurde.

Wenn Sie die IBM MQ-Anwendung für die Routenanzeige verwenden, um die aufgezeichneten Informationen für eine Trace-Route-Nachricht anzuzeigen, können alle zusätzlichen PCF-Parameter nur mit einer numerischer ID angezeigt werden, es sei denn, die Parameter-ID der einzelnen Parameter wird von der IBM MQ-Anwendung für die Routenanzeige erkannt. Um eine Parameter-ID zu erkennen, müssen zusätzliche Informationen mit den folgenden PCF-Parametern aufgezeichnet werden. Fügen Sie diese PCF-Parameter an einer geeigneten Position in die PCF-Gruppe von *Activity* ein.

#### GroupName

Tabelle 13. Gruppename	
Beschreibung	Gruppierte Parameter, die die zusätzlichen Informationen angeben.
ID	MQGACF_VALUE_NAMING.
Datentyp	MQCFGR
Parameter in Gruppe	<i>ParameterName</i> <i>ParameterValue</i>

#### ParameterName

Tabelle 14. Parametername	
Beschreibung	Enthält den Namen, der von der IBM MQ-Anwendung für die Routenanzeige angezeigt werden soll, die den Wert von <i>ParameterValue</i> in den Kontext stellt.
ID	MQCA_VALUE_NAME.
Datentyp	MQCFST
In PCF-Gruppe enthalten:	<i>GroupName.</i>

Tabelle 14. Parametername (Forts.)	
<b>Beschreibung</b>	<b>Enthält den Namen, der von der IBM MQ-Anwendung für die Routenanzeige angezeigt werden soll, die den Wert von <i>ParameterValue</i> in den Kontext stellt.</b>
Wert:	Der Name, der angezeigt werden soll.

### **ParameterValue**

Tabelle 15. Parameterwert	
<b>Beschreibung</b>	<b>Enthält den Wert, der von der IBM MQ-Anwendung für die Routenanzeige angezeigt werden soll.</b>
ID:	Die PCF-Struktur-ID für die zusätzlichen Informationen.
Datentyp:	Der PCF-Strukturdatentyp für die zusätzlichen Informationen.
In PCF-Gruppe enthalten:	<i>GroupName</i> .
Wert:	Der Wert, der angezeigt werden soll.

### **Beispiele für die Aufzeichnung zusätzlicher Aktivitätsinformationen**

Die folgenden Beispiele zeigen, wie eine Benutzeranwendung zusätzliche Informationen aufzeichnen kann, wenn eine Aktivität im Namen einer Trace-Route-Nachricht ausgeführt wird. In beiden Beispielen wird mit der IBM MQ-Anwendung für die Routenanzeige eine Trace-Route-Nachricht generiert und es werden die an diese zurückgegebenen Aktivitätsinformationen angezeigt.

#### *Zusätzliche Aktivitätsinformationen aufzeichnen: Beispiel 1*

Zusätzliche Aktivitätsinformationen werden von einer Benutzeranwendung in einem Format aufgezeichnet, in dem die Parameter-ID *nicht* von der IBM MQ-Anwendung für die Routenanzeige erkannt wird.

1. Mit der IBM MQ-Anwendung für die Routenanzeige wird eine Trace-Route-Nachricht generiert und in ein Warteschlangenmanagernetz gestellt. Die erforderlichen Optionen sind so eingestellt, dass Folgendes angefordert wird:
  - Die Aktivitätsinformationen werden in den Nachrichtendaten der Trace-Route-Nachricht akkumuliert.
  - Bei der Ankunft in der Zielwarteschlange wird die Trace-Route-Nachricht verworfen, und eine Trace-Route-Antwortnachricht wird generiert und an eine angegebene Antwortwarteschlange übergeben.
  - Nach Empfang der Trace-Route-Antwortnachricht werden in der IBM MQ-Anwendung für die Routenanzeige die angesammelten Aktivitätsinformationen angezeigt.

Die Trace-Route-Nachricht wird in das WS-Manager-Netz gestellt.

2. Da die Trace-Route-Nachricht über das WS-Manager-Netz weitergeleitet wird, führt eine Benutzeranwendung, die für die Trace-Route-Nachrichtenübertragung aktiviert ist, eine niedrige Detailaktivität im Namen der Nachricht durch. Zusätzlich zum Schreiben der Standardaktivitätsinformationen in die Trace-Route-Nachricht schreibt die Benutzeranwendung den folgenden PCF-Parameter an das Ende der Aktivitätsgruppe:

#### **ColorValue**

**ID**

65536

**Datentyp**

MQCFST

**Wert**

'Rot'

Dieser zusätzliche PCF-Parameter gibt weitere Informationen zu der Aktivität an, ist jedoch in einem Format geschrieben, in dem die Parameter-ID *nicht* von der IBM MQ-Anwendung für die Routenanzeige erkannt wird.

3. Die Trace-Route-Nachricht erreicht die Zielwarteschlange und es wird eine Trace-Route-Antwortnachricht an die IBM MQ-Anwendung für die Routenanzeige zurückgegeben. Die zusätzlichen Aktivitätsinformationen werden wie folgt angezeigt:

```
65536: 'Red'
```

Die IBM MQ-Anwendung für die Routenanzeige erkennt die Parameter-ID des PCF-Parameters nicht und zeigt sie als numerischen Wert an. Der Kontext der zusätzlichen Informationen ist nicht klar.

Ein Beispiel für die Situation, in der die IBM MQ-Anwendung für die Routenanzeige die Parameter-ID des PCF-Parameters nicht erkennt, finden Sie im Abschnitt „Zusätzliche Aktivitätsinformationen aufzeichnen: Beispiel 2“ auf Seite 82.

#### Zusätzliche Aktivitätsinformationen aufzeichnen: Beispiel 2

Zusätzliche Aktivitätsinformationen werden von einer Benutzeranwendung in einem Format aufgezeichnet, in dem die Parameter-ID von der IBM MQ-Anwendung für die Routenanzeige erkannt wird.

1. Mit der IBM MQ-Anwendung für die Routenanzeige wird eine Trace-Route-Nachricht generiert und auf die gleiche Weise in ein Warteschlangenmanagernetz gestellt, wie in „Zusätzliche Aktivitätsinformationen aufzeichnen: Beispiel 1“ auf Seite 81 beschrieben wird.
2. Da die Trace-Route-Nachricht über das WS-Manager-Netz weitergeleitet wird, führt eine Benutzeranwendung, die für die Trace-Route-Nachrichtenübertragung aktiviert ist, eine niedrige Detailaktivität im Namen der Nachricht durch. Zusätzlich zum Schreiben der Standardaktivitätsinformationen in die Trace-Route-Nachricht schreibt die Benutzeranwendung die folgenden PCF-Parameter an das Ende der Aktivitätsgruppe:

#### ColorInfo

Tabelle 16. Farbinformationen	
Beschreibung	Gruppierte Parameter, die Informationen zu einer Farbe angeben.
ID:	MQGACF_VALUE_NAMING.
Datentyp:	MQCFGR.
Parameter in Gruppe:	<i>ColorName</i> <i>ColorValue</i>

#### ColorName

Tabelle 17. Farbname	
Beschreibung	Enthält den Namen, der von der IBM MQ-Anwendung für die Routenanzeige angezeigt werden soll, die den Wert von <i>ColorValue</i> in den Kontext stellt.
ID:	MQCA_VALUE_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>ColorInfo</i> .
Wert:	'Farbe'

## ColorValue

Tabelle 18. Farbwert	
Beschreibung	Enthält den Wert, der von der IBM MQ-Anwendung für die Routenanzeige angezeigt werden soll.
ID:	65536.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	ColorInfo.
Wert:	'Rot'

Diese zusätzlichen PCF-Parameter geben weitere Informationen zu der ausgeführten Aktivität an. Diese PCF-Parameter werden in einem Format geschrieben, in dem die Parameter-ID von der IBM MQ-Anwendung für die Routenanzeige erkannt wird.

- Die Trace-Route-Nachricht erreicht die Zielwarteschlange und es wird eine Trace-Route-Antwortnachricht an die IBM MQ-Anwendung für die Routenanzeige zurückgegeben. Die zusätzlichen Aktivitätsinformationen werden wie folgt angezeigt:

```
Color: 'Red'
```

Die IBM MQ-Anwendung für die Routenanzeige erkennt, dass die Parameter-ID der PCF-Struktur mit dem Wert der zusätzlichen Aktivitätsinformationen einen entsprechenden Namen hat. Der entsprechende Name wird anstelle des numerischen Werts angezeigt.

## IBM MQ-Anwendung für die Routenanzeige

Mit der IBM MQ-Anwendung für die Routenanzeige (**dspmqrte**) können Sie über eine Befehlszeilenschnittstelle mit Trace-Route-Nachrichten und Aktivitätsinformationen, die sich auf eine Trace-Route-Nachricht beziehen, arbeiten.  Die IBM MQ-Anwendung für die Routenanzeige ist nicht im Lieferumfang von IBM MQ for z/OS enthalten, aber Sie können sie in einer verteilten Installation ausführen und als Client mit einem IBM MQ for z/OS-Warteschlangenmanager verbinden, indem Sie den Parameter **-c** beim Absetzen des Befehls **dspmqrte** angeben.

Sie können die IBM MQ **dspmqrte**-Anwendung für die Routenanzeige für die folgenden Zwecke verwenden:

- Zum Konfigurieren, Generieren und Einlegen einer Trace-Route-Nachricht in ein WS-Manager-Netz.

Wenn Sie eine Trace-Route-Nachricht in ein WS-Manager-Netz einreihen, können Aktivitätsinformationen erfasst und verwendet werden, um die Route zu bestimmen, die die Trace-Route-Nachricht genommen hat. Sie können die Merkmale der Trace-Route-Nachrichten wie folgt angeben:

- Die Zieladresse der Trace-Route-Nachricht.
  - Gibt an, wie die Trace-Route-Nachricht eine andere Nachricht imitiert.
  - Wie die Trace-Route-Nachricht behandelt werden soll, wenn sie über ein WS-Manager-Netz weitergeleitet wird.
  - Gibt an, ob die Aktivitäten-Aufzeichnung oder Trace-Route-Messaging zum Aufzeichnen von Aktivitätsinformationen verwendet wird.
- Gehen Sie wie folgt vor, um Aktivitätsinformationen für eine Trace-Route-Nachricht zu sortieren und anzuzeigen.

Wenn die IBM MQ-Anwendung für die Routenanzeige eine Trace-Route-Nachricht in ein Warteschlangenmanagernetz gestellt hat, nachdem die zugehörigen Aktivitätsinformationen zurückgegeben wurden, können die Informationen sofort sortiert und angezeigt werden. Alternativ können mit der IBM

MQ-Anwendung zur Routenanzeige Aktivitätsinformationen zu einer Trace-Route-Nachricht, die zuvor generiert wurde, sortiert und angezeigt werden.

### Zugehörige Verweise

[dspmqrte](#)

### **Parameter für Trace-Route-Nachrichten**

Verwenden Sie diese Seite, um eine Übersicht über die Parameter zu erhalten, die von der IBM MQ-Anwendung für die Routenanzeige **dspmqrte** bereitgestellt werden, um die Merkmale einer Trace-Route-Nachricht zu ermitteln. Dazu gehört auch die Verarbeitung bei der Weiterleitung durch ein Warteschlangenmanagernetz.

### Zugehörige Verweise

[dspmqrte](#)

#### *Warteschlangenmanagerverbindung*

Geben Sie auf dieser Seite den Warteschlangenmanager an, zu dem die IBM MQ-Anwendung für die Routenanzeige eine Verbindung herstellt.

#### **-c**

Gibt an, dass die IBM MQ-Anwendung für die Routenanzeige eine Verbindung als Clientanwendung herstellt.

Wenn Sie diesen Parameter nicht angeben, stellt die IBM MQ-Anwendung für die Routenanzeige keine Verbindung als Clientanwendung her.

#### **-m Warteschlangenmanager-name**

Der Name des Warteschlangenmanagers, zu dem die IBM MQ-Anwendung für die Routenanzeige eine Verbindung herstellt. Der Name kann bis zu 48 Zeichen enthalten.

Wenn Sie diesen Parameter nicht angeben, wird der Standardwarteschlangenmanager verwendet.

#### *Zieladresse*

Verwenden Sie diese Seite, um die Zieladresse einer Trace-Route-Nachricht anzugeben.

#### **-q TargetQName**

Wenn die IBM MQ-Anwendung zur Routenanzeige verwendet wird, um eine Trace-Route-Nachricht in ein Warteschlangenmanagernetz zu senden, gibt *TargetQName* den Namen der Zielwarteschlange an.

#### **-ts TargetTopicString**

Gibt die Themenzeichenfolge an.

#### **-qm TargetQMgr**

Qualifiziert die Ziel-Destination; die normale Auflösung des Warteschlangenmanagers wird dann angewendet. Die Ziel-Destination wird mit *-q TargetQName* oder *-ts TargetTopicString* angegeben.

Wenn Sie diesen Parameter nicht angeben, wird der Warteschlangenmanager, mit dem die IBM MQ-Anwendung für die Routenanzeige verbunden ist, als Zielwarteschlangenmanager verwendet.

#### **-o**

Gibt an, dass die Ziel-Destination nicht an ein bestimmtes Ziel gebunden ist. In der Regel wird dieser Parameter verwendet, wenn die Trace-Route-Nachricht in einen Cluster gestellt werden soll. Die Ziel-Destination wird mit der Option `MQOO_BIND_NOT_FIXED` geöffnet.

Wenn Sie diesen Parameter nicht angeben, wird die Ziel-Destination an ein bestimmtes Ziel gebunden.

#### *Das Veröffentlichungsthema*

Verwenden Sie für Publish/Subscribe-Anwendungen diese Seite, um die Themenzeichenfolge einer Trace-Route-Nachricht für die IBM MQ-Anwendung für die Routenanzeige anzugeben, die veröffentlicht werden soll.

### **-ts TopicName**

Gibt eine Themenzeichenfolge an, für die die IBM MQ-Anwendung für die Routenanzeige eine Trace-Route-Nachricht veröffentlichen soll, und stellt diese Anwendung in den Themenmodus. In diesem Modus verfolgt die Anwendung alle Nachrichten, die sich aus der Veröffentlichungsanforderung ergeben.

Mit der IBM MQ-Anwendung für die Routenanzeige können Sie die Ergebnisse aus einem Aktivitätsbericht anzeigen, der für die Veröffentlichung von Nachrichten generiert wurde.

### *Nachrichtenmimierung*

Verwenden Sie diese Seite, um eine Trace-Route-Nachricht zu konfigurieren, um eine Nachricht zu imitieren, z. B., wenn die ursprüngliche Nachricht nicht das geplante Ziel erreicht hat.

Eine Verwendung des Trace-Route-Messaging ist die Bestimmung der letzten bekannten Position einer Nachricht, die nicht an das geplante Ziel gelangt ist. In der IBM MQ-Anwendung für die Routenanzeige werden Parameter bereitgestellt, mit denen eine Trace-Route-Nachricht konfiguriert werden kann, die die ursprüngliche Nachricht imitiert. Wenn Sie eine Nachricht imitieren, können Sie die folgenden Parameter verwenden:

### **-l Persistence**

Gibt die Persistenz der generierten Trace-Route-Nachricht an. Gültige Werte für *Persistence* sind:

#### **ja**

Die generierte Trace-Route-Nachricht ist persistent. (MQPER\_PERSISTENT).

#### **nein**

Die generierte Trace-Route-Nachricht ist **nicht** persistent. (MQPER\_NOT\_PERSISTENT).

#### **q**

Die generierte Trace-Route-Nachricht übernimmt ihren Persistenzwert von der Destination, die durch *-q TargetQName* oder *-ts TargetTopicString* angegeben wird. (MQPER\_PERSISTENCE\_AS\_Q\_DEF).

Eine Trace-Route-Antwortnachricht oder die zurückgegebenen Berichtsnachrichten verwenden denselben Persistenzwert wie die ursprüngliche Trace-Route-Nachricht.

Wenn *Persistence* als **yes** angegeben ist, müssen Sie den Parameter *-rq ReplyToQ* angeben. Die Empfangswarteschlange für Antworten darf nicht in eine temporäre dynamische Warteschlange aufgelöst werden.

Wenn Sie diesen Parameter nicht angeben, wird die generierte Trace-Route-Nachricht **nicht** permanent angezeigt.

### **-p Priority**

Gibt die Priorität der Trace-Route-Nachricht an. Der Wert von *Priority* ist größer-gleich 0, oder MQPRI\_PRIORITY\_AS\_Q\_DEF. MQPRI\_PRIORITY\_AS\_Q\_DEF gibt an, dass der Prioritätswert von dem Ziel übernommen wird, das durch *-q TargetQName* oder *-ts TargetTopicString* angegeben wird.

Wenn Sie diesen Parameter nicht angeben, wird der Prioritätswert von dem Ziel übernommen, das durch *-q TargetQName* oder *-ts TargetTopicString* angegeben wird.

### **-xs Expiry**

Gibt die Ablaufzeit für die Trace-Route-Nachricht in Sekunden an.

Wenn Sie diesen Parameter nicht angeben, wird die Ablaufzeit als 60 Sekunden angegeben.

### **-ro none | ReportOption**

#### **none**

Gibt an, dass keine Berichtsoptionen festgelegt sind.

#### **ReportOption**

Gibt Berichtsoptionen für die Trace-Route-Nachricht an. Mehrere Berichtsoptionen können mit einem Komma als Trennzeichen angegeben werden. Gültige Werte für *ReportOption* sind:

#### **Aktivität**

Die Berichtsoption MQRO\_ACTIVITY wird festgelegt.

- Die Berichtsoption MQRO\_COA\_WITH\_FULL\_DATA ist festgelegt.

#### **Kabeljau**

Die Berichtsoption MQRO\_COD\_WITH\_FULL\_DATA ist festgelegt.

#### **ausnahmebedingung**

Die Berichtsoption 'MQRO\_EXCEPTION\_WITH\_FULL\_DATA' wird festgelegt.

#### **Ablauf**

Die Berichtsoption 'MQRO\_EXPIRATION\_WITH\_FULL\_DATA' wird festgelegt.

#### **Verwerfen**

Die Berichtsoption MQRO\_DISCARD\_MSG wird festgelegt.

Wenn weder *-ro ReportOption* noch *-ro none* angegeben sind, werden die Berichtsoptionen MQRO\_ACTIVITY und MQRO\_DISCARD\_MSG angegeben.

In der IBM MQ-Anwendung zur Routenanzeige ist es nicht zulässig, Benutzerdaten zur Trace-Route-Nachricht hinzuzufügen. Wenn Benutzerdaten zur Trace-Route-Nachricht hinzugefügt werden müssen, müssen Sie die Trace-Route-Nachricht manuell generieren.

#### *Aufgezeichnete Aktivitätsinformationen*

Verwenden Sie diese Seite, um die Methode anzugeben, mit der aufgezeichnete Aktivitätsinformationen zurückgegeben werden, die Sie dann verwenden können, um die Route zu bestimmen, die eine Trace-Route-Nachricht eingenommen hat.

Aufgezeichnete Aktivitätsinformationen können wie folgt zurückgegeben werden:

- In Aktivitätsberichten
- In einer Trace-Route-Antwortnachricht
- In der Trace-Route-Nachricht selbst (die in die Zielwarteschlange gestellt wurde)

Bei Verwendung von **dspmqrte** wird die Methode zur Rückgabe aufgezeichneter Aktivitätsinformationen mithilfe der folgenden Parameter bestimmt:

#### **-ro activity**

Gibt an, dass Aktivitätsinformationen mithilfe von Aktivitätsberichten zurückgegeben werden. Standardmäßig ist die Aktivitätsaufzeichnung aktiviert.

#### **-ac -ar**

Gibt an, dass die Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden und dass eine Trace-Route-Antwortnachricht generiert werden soll.

#### **-ac**

Gibt an, dass die Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden sollen.

Wenn Sie diesen Parameter nicht angeben, werden die Aktivitätsinformationen in der Trace-Route-Nachricht nicht kumuliert.

#### **-ar**

Fordert an, dass eine Trace-Route-Antwortnachricht mit allen aufgelaufenen Aktivitätsinformationen in den folgenden Situationen generiert wird:

- Die Trace-Route-Nachricht wird von einem IBM MQ-Warteschlangenmanager gelöscht.
- Die Trace-Route-Nachricht wird von einem IBM MQ-Warteschlangenmanager in eine lokale Warteschlange gestellt (Zielwarteschlange oder Warteschlange für nicht zustellbare Nachrichten).
- Die Anzahl der Aktivitäten, die in der Trace-Route-Nachricht ausgeführt werden, überschreitet den in *-s Activities* angegebenen Wert.

#### **-ac -d ja**

Gibt an, dass die Aktivitätsinformationen in der Trace-Route-Nachricht kumuliert werden und dass bei der Ankunft die Trace-Route-Nachricht in die Zielwarteschlange gestellt werden soll.

#### **-ac**

Gibt an, dass die Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden sollen.

Wenn Sie diesen Parameter nicht angeben, werden die Aktivitätsinformationen in der Trace-Route-Nachricht nicht kumuliert.

**-d Ja**

Bei der Ankunft wird die Trace-Route-Nachricht in die Zielwarteschlange gestellt, auch wenn der WS-Manager keine Trace-Route-Messaging unterstützt.

Wenn Sie diesen Parameter nicht angeben, wird die Trace-Route-Nachricht nicht in die Zielwarteschlange gestellt.

Die Trace-Route-Nachricht kann dann aus der Zielwarteschlange abgerufen werden und die erfassten Aktivitätsinformationen werden erfasst.

Sie können diese Methoden nach Bedarf kombinieren.

Darüber hinaus kann die Detailstufe der aufgezeichneten Aktivitätsinformationen mit dem folgenden Parameter angegeben werden:

**-t Detail**

Gibt die Aktivitäten an, die aufgezeichnet werden. Die gültigen Werte für *Detail* sind:

**Niedrig**

Aktivitäten, die von einer vom Benutzer definierten Anwendung ausgeführt werden, werden nur aufgezeichnet.

**Medium**

Die in "low" angegebenen Aktivitäten werden aufgezeichnet. Darüber hinaus werden Veröffentlichungsaktivitäten und Aktivitäten, die von MCAs ausgeführt werden, aufgezeichnet.

**hoch**

Aktivitäten, die in "low" und "medium" angegeben sind, werden aufgezeichnet. MCAs stellen keine weiteren Aktivitätsinformationen auf dieser Detaillierungsebene bereit. Diese Option ist für benutzerdefinierte Anwendungen verfügbar, die nur weitere Aktivitätsinformationen zugänglich machen sollen. Wenn z. B. eine benutzerdefinierte Anwendung die Route einer Nachricht durch Berücksichtigung bestimmter Nachrichtenmerkmale bestimmt, könnte die Routing-Logik in diese Detaillierungsebene eingeschlossen werden.

Wenn Sie diesen Parameter nicht angeben, werden die Aktivitäten auf mittlerer Ebene aufgezeichnet.

Die IBM MQ-Anwendung für die Routenanzeige speichert die zurückgegebenen Nachrichten standardmäßig in einer temporären dynamischen Warteschlange. Wenn die IBM MQ-Anwendung für die Routenanzeige beendet wird, wird die temporäre dynamische Warteschlange geschlossen und alle Nachrichten werden bereinigt. Wenn die zurückgegebenen Nachrichten über die aktuelle Ausführung der IBM MQ-Anwendung für die Routenanzeige hinaus erforderlich sind, muss mit den folgenden Parametern eine permanente Warteschlange angegeben werden:

**-rq ReplyToQ**

Gibt den Namen der Warteschlange für Antwortnachrichten an, an die alle Antworten auf die Trace-Route-Nachricht gesendet werden. Wenn die Trace-Route-Nachricht persistent ist oder wenn der Parameter *-n* angegeben ist, muss eine Empfangswarteschlange für Antworten angegeben werden, die keine temporäre dynamische Warteschlange ist.

Wenn Sie diesen Parameter nicht angeben, wird eine dynamische Empfangswarteschlange für Antworten unter Verwendung der Systemstandardmodellwarteschlange SYSTEM.DEFAULT.MODEL.QUEUE erstellt.

**-rqm ReplyToQMGr**

Gibt den Namen des Warteschlangenmanagers an, in dem sich die Antwortwarteschlange befindet. Der Name kann bis zu 48 Zeichen enthalten.

Wenn Sie diesen Parameter nicht angeben, wird der Warteschlangenmanager, mit dem die IBM MQ-Anwendung für die Routenanzeige verbunden ist, als Warteschlangenmanager für Antwortnachrichten verwendet.

### Handhabung der Trace-Route-Nachricht

Verwenden Sie diese Seite, um zu steuern, wie eine Trace-Route-Nachricht verarbeitet wird, wenn sie über ein Warteschlangenmanagernetzwerk weitergeleitet wird.

Die folgenden Parameter können einschränken, wo die Trace-Route-Nachricht im WS-Manager-Netz weitergeleitet werden kann:

#### **-d Deliver**

Gibt an, ob die Trace-Route-Nachricht bei der Ankunft an die Zielwarteschlange zugestellt werden soll. Gültige Werte für *Deliver* sind:

##### **ja**

Bei der Ankunft wird die Trace-Route-Nachricht in die Zielwarteschlange gestellt, auch wenn der WS-Manager keine Trace-Route-Messaging unterstützt.

##### **nein**

Bei der Ankunft wird die Trace-Route-Nachricht nicht in die Zielwarteschlange gestellt.

Wenn Sie diesen Parameter nicht angeben, wird die Trace-Route-Nachricht nicht in die Zielwarteschlange gestellt.

#### **-f Forward**

Gibt den Typ des Warteschlangenmanagers an, an den die Trace-Route-Nachricht weitergeleitet werden kann. Ausführliche Informationen zu dem Algorithmus, den Warteschlangenmanager verwenden, um zu bestimmen, ob eine Nachricht an einen fernen Warteschlangenmanager weitergeleitet werden soll, finden Sie in „PCF-Gruppe "TraceRoute"“ auf Seite 74. Die gültigen Werte für *Forward* sind:

##### **alle**

Die Trace-Route-Nachricht wird an einen beliebigen WS-Manager weitergeleitet.

**Warnung:** Bei der Weiterleitung an einen Warteschlangenmanager vor IBM WebSphere MQ 6.0 wird die Trace-Route-Nachricht nicht erkannt und kann trotz des Werts des Parameters *-d Deliver* an eine lokale Warteschlange übergeben werden.

##### **unterstützt**

Die Trace-Route-Nachricht wird nur an einen Warteschlangenmanager weitergeleitet, der den Parameter *Deliver* von der *TraceRoute* -PCF-Gruppe berücksichtigt.

Wenn Sie diesen Parameter nicht angeben, wird die Trace-Route-Nachricht nur an einen WS-Manager weitergeleitet, der den Parameter *Deliver* berücksichtigt.

Die folgenden Parameter können verhindern, dass eine Trace-Route-Nachricht in einem Warteschlangenmanagernetzwerk unendlich bleibt:

#### **-s Activities**

Gibt die maximale Anzahl aufgezeichneter Aktivitäten an, die im Namen der Trace-Route-Nachricht ausgeführt werden können, bevor sie gelöscht wird. Dadurch wird verhindert, dass die Trace-Route-Nachricht unbegrenzt weitergeleitet wird, wenn sie in einer Endlosschleife abgefangen wird. Der Wert von *Activities* ist größer-gleich 1, oder MQROUTE\_UNLIMITED\_ACTIVITIES. MQROUTE\_UNLIMITED\_ACTIVITIES gibt an, dass eine unbegrenzte Anzahl von Aktivitäten im Namen der Trace-Route-Nachricht ausgeführt werden kann.

Wenn Sie diesen Parameter nicht angeben, kann im Namen der Trace-Route-Nachricht eine unbegrenzte Anzahl von Aktivitäten ausgeführt werden.

#### **-xs Expiry**

Gibt die Ablaufzeit für die Trace-Route-Nachricht in Sekunden an.

Wenn Sie diesen Parameter nicht angeben, wird die Ablaufzeit als 60 Sekunden angegeben.

#### **-xp PassExpiry**

Gibt an, ob die Ablaufzeit von der Trace-Route-Nachricht an eine Trace-Route-Antwortnachricht übergeben wird. Gültige Werte für *PassExpiry* sind:

**ja**

Die Berichtsoption 'MQRO\_PASS\_DISCARD\_AND\_EXPIRY' wird im Nachrichtendeskriptor der Trace-Route-Nachricht angegeben.

Wenn eine Trace-Route-Antwortnachricht oder Aktivitätsberichte für die Trace-Route-Nachricht generiert werden, wird die MQRO\_DISCARD-Berichtsoption (falls angegeben) und die verbleibende Ablaufzeit an übergeben.

Dies ist der Standardwert.

**nein**

Die Berichtsoption 'MQRO\_PASS\_DISCARD\_AND\_EXPIRY' wurde nicht angegeben.

Wenn eine Trace-Route-Antwortnachricht für die Trace-Route-Nachricht generiert wird, werden die Löschoption und die Ablaufzeit aus der Trace-Route-Nachricht **nicht** weitergegeben.

Wenn Sie diesen Parameter nicht angeben, wird MQRO\_PASS\_DISCARD\_AND\_EXPIRY nicht angegeben.

**-ro discard**

Gibt die Berichtsoption MQRO\_DISCARD\_MSG an. Dadurch kann verhindert werden, dass die im WS-Manager-Netz verbleibende Trace-Route-Nachricht unbegrenzt bleibt.

**Anzeige von Aktivitätsinformationen**

Die IBM MQ-Anwendung für die Routenanzeige kann Aktivitätsinformationen für eine Trace-Route-Nachricht anzeigen, die sie gerade in ein Warteschlangenmanagernetz gestellt hat, oder sie kann Aktivitätsinformationen für eine zuvor generierte Trace-Route-Nachricht anzeigen. Es kann auch zusätzliche Informationen anzeigen, die von benutzerschriftlichen Anwendungen aufgezeichnet wurden.

Geben Sie den folgenden Parameter an, um anzugeben, ob Aktivitätsinformationen, die für eine Trace-Route-Nachricht zurückgegeben werden, angezeigt werden sollen:

**-n**

Gibt an, dass Aktivitätsinformationen, die für die Trace-Route-Nachricht zurückgegeben werden, nicht angezeigt werden sollen.

Wenn dieser Parameter von einer Anforderung für eine Trace-Route-Antwortnachricht ( *-ar* ) oder einen der Berichtsgenerierungsoptionen aus ( *-ro ReportOption* ) begleitet wird, muss eine bestimmte (nicht-Modell-) Antwortwarteschlange angegeben werden, die mit *-rq ReplyToQ* angegeben wird. Standardmäßig werden nur Aktivitätsberichtsrichten angefordert.

Nachdem die Trace-Route-Nachricht in die angegebene Zielwarteschlange gestellt wurde, wird eine 48 Zeichen lange hexadezimale Zeichenfolge angezeigt, die die Nachrichten-ID der Trace-Route-Nachricht enthält. Die Nachrichten-ID kann von der IBM MQ-Anwendung für die Routenanzeige verwendet werden, um die Aktivitätsinformationen für die Trace-Route-Nachricht mit dem Parameter *-i CorrelId* zu einem späteren Zeitpunkt anzuzeigen.

Wenn Sie diesen Parameter nicht angeben, werden die Aktivitätsinformationen, die für die Trace-Route-Nachricht zurückgegeben werden, in dem durch den Parameter *-v* angegebenen Format angezeigt.

Beim Anzeigen von Aktivitätsinformationen für eine Trace-Route-Nachricht, die gerade in ein WS-Manager-Netz gestellt wurde, kann der folgende Parameter angegeben werden:

**-w WaitTime**

Gibt die Dauer in Sekunden an, die die IBM MQ-Anwendung für die Routenanzeige auf Aktivitätsberichte oder auf eine Trace-Route-Antwortnachricht wartet, um diese an die angegebene Empfangswarteschlange für Antworten zurückzugeben.

Wenn Sie diesen Parameter nicht angeben, wird die Wartezeit als Verfallszeit der Trace-Route-Nachricht plus 60 Sekunden angegeben.

Bei der Anzeige der zuvor aufgelaufenen Aktivitätsinformationen müssen die folgenden Parameter festgelegt werden:

### **-q TargetQName**

Wenn die IBM MQ-Anwendung für die Routenanzeige verwendet wird, um vorher zusammengestellte Aktivitätsinformationen anzuzeigen, gibt *TargetQName* den Namen der Warteschlange an, in der die Aktivitätsinformationen gespeichert sind.

### **-i CorrelId**

Dieser Parameter wird verwendet, wenn mit der IBM MQ-Anwendung für die Routenanzeige nur vorher gesammelte Aktivitätsinformationen angezeigt werden. Es können viele Aktivitätsberichte und Trace-Route-Antwortnachrichten in der Warteschlange vorhanden sein, die durch *-q Zielwarteschlangennamenname* angegeben wird. *CorrelId* wird verwendet, um die Aktivitätsberichte oder eine Trace-Route-Antwortnachricht in Bezug auf eine Trace-Route-Nachricht zu identifizieren. Geben Sie die Nachrichten-ID der ursprünglichen Trace-Route-Nachricht in *CorrelId* an.

Das Format von *CorrelId* ist eine hexadezimale Zeichenfolge mit 48 Zeichen.

Die folgenden Parameter können verwendet werden, wenn zuvor aufgelaufene Aktivitätsinformationen angezeigt werden, oder wenn die aktuellen Aktivitätsinformationen für eine Trace-Route-Nachricht angezeigt werden:

### **-b**

Gibt an, dass die IBM MQ-Anwendung für die Routenanzeige nur Aktivitätsberichte oder eine Trace-Route-Antwortnachricht in Bezug auf eine Nachricht durchsucht. Dadurch können die Aktivitätsinformationen zu einem späteren Zeitpunkt erneut angezeigt werden.

Wenn Sie diesen Parameter nicht angeben, führt die IBM MQ-Anwendung für die Routenanzeige für Aktivitätsberichte oder für eine Trace-Route-Antwortnachricht, die sich auf eine Nachricht bezieht, einen Abruf mit Löschen aus.

### **-v summary | all | none | outline DisplayOption**

#### **Zusammenfassung**

Die Warteschlangen, an die die Trace-Route-Nachricht weitergeleitet wurde, werden angezeigt.

#### **alle**

Alle verfügbaren Informationen werden angezeigt.

#### **none**

Es werden keine Informationen angezeigt.

#### **Gliederung DisplayOption**

Gibt die Anzeigeeoptionen für die Trace-Route-Nachricht an. Mehrere Anzeigeeoptionen können mit einem Komma als Trennzeichen angegeben werden.

Wenn keine Werte angegeben werden, wird Folgendes angezeigt:

- Der Anwendungsname
- Der Typ jeder Operation.
- Alle betriebspezifischen Parameter

Gültige Werte für *DisplayOption* sind:

#### **Aktivität**

Alle Nicht-PCF-Gruppenparameter in *Activity* -PCF-Gruppen werden angezeigt.

#### **identifiers**

Es werden Werte mit den Parameter-IDs MQBACF\_MSG\_ID oder MQBACF\_CORREL\_ID angezeigt. Dieser Wert überschreibt *msgdelta*.

#### **Nachricht**

Alle Nicht-PCF-Gruppenparameter in *Message* -PCF-Gruppen werden angezeigt. Wenn dieser Wert angegeben wird, können Sie *msgdelta* nicht angeben.

#### **msgdelta**

Alle Nicht-PCF-Gruppenparameter in *Message* -PCF-Gruppen, die seit der letzten Operation geändert wurden, werden angezeigt. Wenn dieser Wert angegeben wird, können Sie *message* nicht angeben.

## Operation

Alle Nicht-PCF-Gruppenparameter in *Operation* -PCF-Gruppen werden angezeigt.

## Traceroute

Alle Nicht-PCF-Gruppenparameter in *TraceRoute* -PCF-Gruppen werden angezeigt.

Wenn Sie diesen Parameter nicht angeben, wird eine Zusammenfassung der Nachrichtenroute angezeigt.

## Anzeige zusätzlicher Informationen

Da eine Trace-Route-Nachricht über ein WS-Manager-Netz weitergeleitet wird, können benutzerdefinierte Anwendungen zusätzliche Informationen aufzeichnen, indem sie einen oder mehrere zusätzliche PCF-Parameter in die Nachrichtendaten der Trace-Route-Nachricht oder in die Nachrichtendaten eines Aktivitätsberichts schreiben. Damit die IBM MQ-Anwendung für die Routenanzeige zusätzliche Informationen in einem lesbaren Format anzeigen kann, muss sie in einem bestimmten Format aufgezeichnet werden, wie im Abschnitt „Zusätzliche Aktivitätsinformationen“ auf Seite 80 beschrieben wird.

## Beispiele für die IBM MQ-Anwendung für die Routenanzeige

In den folgenden Beispielen wird gezeigt, wie Sie die IBM MQ-Anwendung für die Routenanzeige verwenden können. In jedem Beispiel sind zwei WS-Manager (QM1 und QM2) über zwei Kanäle miteinander verbunden (QM2.TO.QM1 und QM1.TO.QM2).

### Beispiel 1-Aktivitätenberichte anfordern

Aktivitätsinformationen aus einer Trace-Route-Nachricht anzeigen, die an die Zielwarteschlange zugelegt wird

In diesem Beispiel stellt die IBM MQ-Anwendung für die Routenanzeige eine Verbindung zu Warteschlangenmanager QM1 her und wird dazu verwendet, eine Trace-Route-Nachricht zu generieren und an die Zielwarteschlange TARGET.Q auf dem fernen Warteschlangenmanager QM2 zu übergeben. Die erforderliche Berichtsoption wird so angegeben, dass Aktivitätsberichte angefordert werden, wenn die Antwortnachricht für die Trace-Route weitergeleitet wird. Bei der Ankunft in der Zielwarteschlange wird die Trace-Route-Nachricht verworfen. Aktivitätsinformationen, die mithilfe von Aktivitätsberichten an die IBM MQ-Anwendung für die Routenanzeige zurückgegeben werden, werden sortiert und angezeigt.

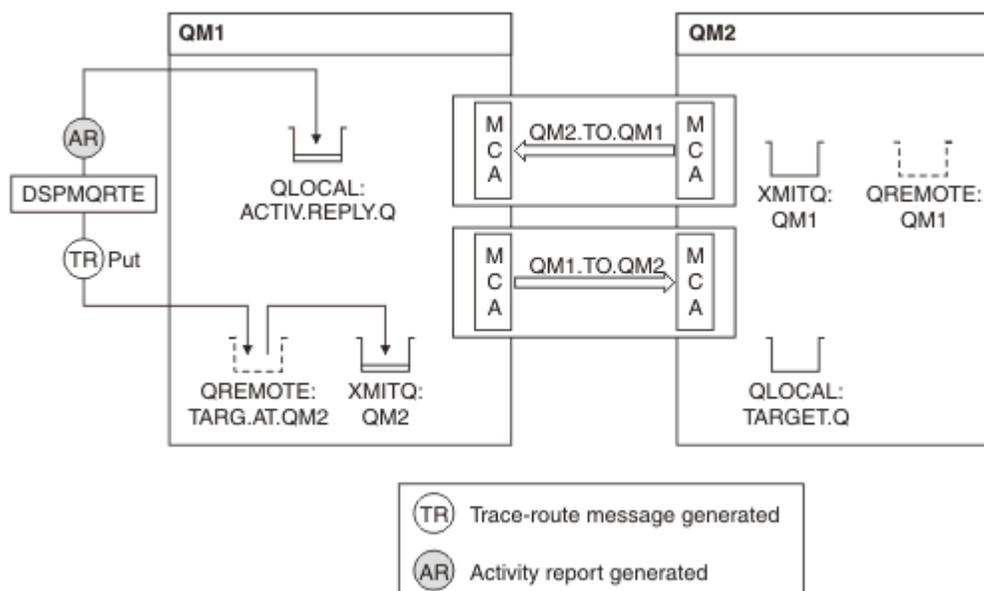


Abbildung 9. Aktivitätenberichte anfordern, Diagramm 1

- Das Attribut ACTIVREC der einzelnen WS-Manager (QM1 und QM2) wird auf MSG gesetzt.
- Der folgende Befehl wird ausgegeben:

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq ACTIV.REPLY.Q
```

QM1 ist der Name des Warteschlangenmanagers, mit dem die IBM MQ-Anwendung für die Routenanzeige eine Verbindung herstellt, TARG.AT.QM2 ist der Name der Zielwarteschlange und ACTIV.REPLY.Q ist der Name der Warteschlange, in die alle Antworten auf die Trace-Route-Nachricht gesendet werden soll.

Für alle nicht angegebenen Optionen werden Standardwerte angenommen, aber insbesondere die Option -f (die Trace-Route-Nachricht wird nur an einen Warteschlangenmanager weitergeleitet, der den Parameter "Deliver" der PCF-Gruppe "TraceRoute" berücksichtigt), die Option -d (bei der Ankunft, die Trace-Route-Nachricht wird nicht in die Zielwarteschlange gestellt), die Option -ro (MQRO\_ACTIVITY und MQRO\_DISCARD\_MSG Berichtsoptionen angegeben) und die Option -t (mittlere Aktivität auf Detailebene wird aufgezeichnet).

- DSPMQRTE generiert die Trace-Route-Nachricht und stellt sie in die ferne Warteschlange TARG.AT.QM2.
- DSPMQRTE sieht dann den Wert des Attributs ACTIVREC des Warteschlangenmanagers QM1 aus. Der Wert ist MSG, daher generiert DSPMQRTE einen Aktivitätsbericht und stellt ihn in die Antwortwarteschlange ACTIV.REPLY.Q.

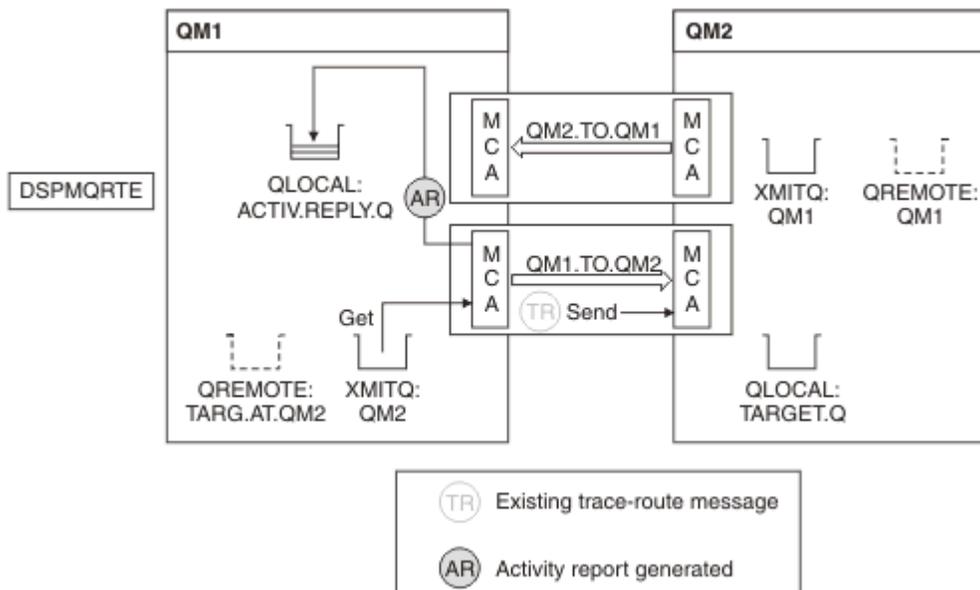


Abbildung 10. Aktivitätenberichte anfordern, Diagramm 2

- Der sendende Nachrichtenkanalagent (MCA) ruft die Trace-Route-Nachricht aus der Übertragungswarteschlange ab. Die Nachricht ist eine Trace-Route-Nachricht, daher beginnt der MCA mit der Aufzeichnung der Aktivitätsinformationen.
- Das Attribut ACTIVREC des Warteschlangenmanagers (QM1) lautet MSG, und die Option MQRO\_ACTIVITY wird im Feld "Bericht" des Nachrichtendeskriptors angegeben. Daher generiert der Nachrichtenkanalmanager später einen Aktivitätsbericht. Der Wert des Parameters "RecordedActivities" in der PCF-Gruppe "TraceRoute" wird um 1 erhöht.
- Der MCA überprüft, ob der Wert für MaxActivities in der PCF-Gruppe "TraceRoute" nicht überschritten wurde.
- Bevor die Nachricht an QM2 weitergeleitet wird, folgt der MCA dem Algorithmus, der in [Weiterleiten](#) beschrieben ist (Schritte „1“ auf Seite 77, „4“ auf Seite 77 und „5“ auf Seite 77), und der Nachrichtenkanalmanager wählt die Nachricht aus.
- Der MCA generiert dann einen Aktivitätsbericht und stellt ihn in die Antwortwarteschlange (ACTIV.REPLY.Q).

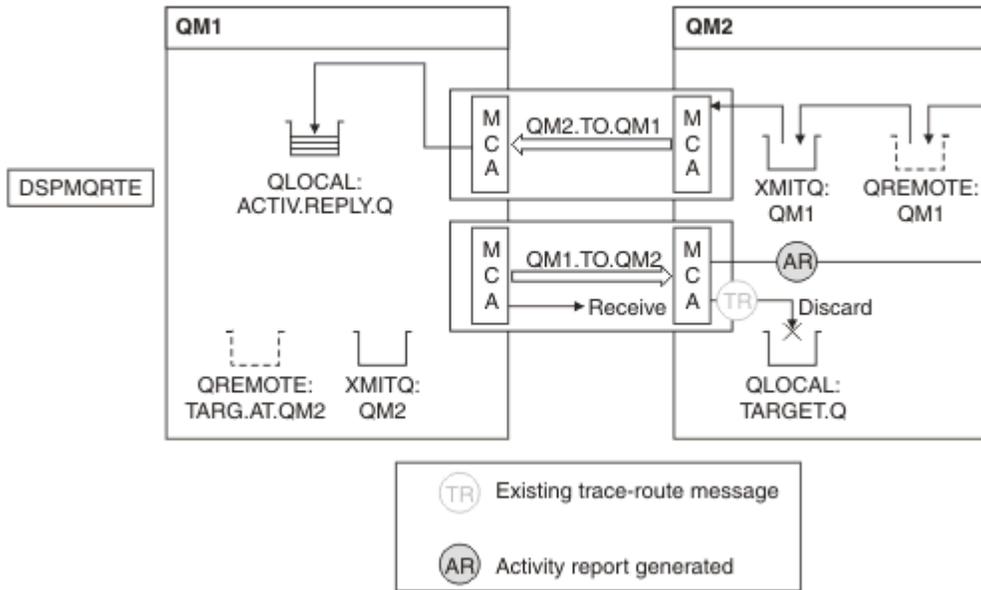


Abbildung 11. Aktivitätenberichte anfordern, Diagramm 3

- Der empfangende MCA empfängt die Trace-Route-Nachricht aus dem Kanal. Die Nachricht ist eine Trace-Route-Nachricht, daher beginnt der MCA mit der Aufzeichnung der Informationen über die Aktivität.
- Wenn der Warteschlangenmanager, von dem die Trace-Route-Nachricht stammt, von Version IBM WebSphere MQ 5.3.1 oder einer früheren Version stammt, erhöht der MCA den Parameter Discontinuity-Count des PCF-Trace-Leitwegs um 1. Dies ist hier nicht der Fall.
- Das Attribut ACTIVREC des Warteschlangenmanagers (QM2) ist MSG, und die Option MQRO\_ACTIVITY wird angegeben, daher generiert der Nachrichtenkanalmanager einen Aktivitätsbericht. Der Wert des Parameters RecordedActivities wird um 1 erhöht.
- Die Zielwarteschlange ist eine lokale Warteschlange, daher wird die Nachricht mit dem Feedback MQFB\_NOT\_DELIVERED in Übereinstimmung mit dem Wert des Parameters Deliver in der PCF-Gruppe TraceRoute gelöscht.
- Der MCA generiert dann den endgültigen Aktivitätsbericht und stellt ihn in die Antwortwarteschlange. Dies wird in die Übertragungswarteschlange aufgelöst, die dem WS-Manager QM1 zugeordnet ist, und der Aktivitätsbericht wird an Warteschlangenmanager QM1 (ACTIV.REPLY.Q) zurückgegeben.

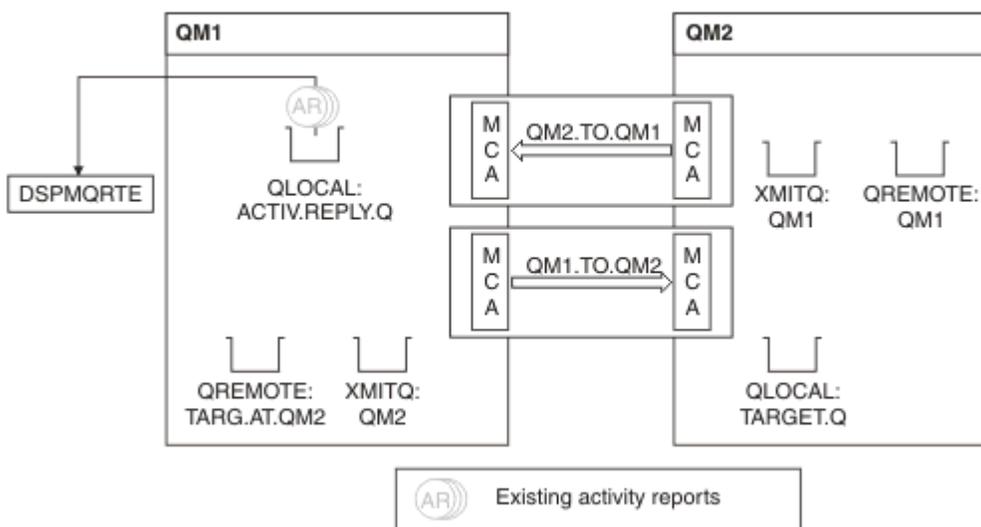


Abbildung 12. Aktivitätenberichte anfordern, Diagramm 4

- In der Zwischenzeit hat DSPMQRTE MQGETs in der Antwortwarteschlange (ACTIV.REPLY.Q) ausgeführt, die auf Aktivitätsberichte wartet. Es wartet bis zu 120 Sekunden (60 Sekunden länger als die Ablaufzeit der Trace-Route-Nachricht), da -w nicht angegeben wurde, als DSPMQRTE gestartet wurde.
- DSPMQRTE ruft die 3-Aktivität ab, die von der Antwortwarteschlange abgemeldet wird.
- Die Aktivitätsberichte werden unter Verwendung der Parameter "RecordedActivities", "UnrecordedActivities" und "DiscontinuityCount" in der PCF-Gruppe "TraceRoute" für jede der Aktivitäten geordnet. Der einzige Wert, der in diesem Beispiel ungleich null ist, ist RecordedActivities. Daher ist dieser Parameter der einzige tatsächlich verwendete Parameter.
- Das Programm wird beendet, sobald die Löschoption angezeigt wird. Obwohl die letzte Operation eine Löschoption war, wird sie so behandelt, als sei eine Nachricht in der Form, dass die Rückmeldung MQFB\_NOT\_DELIVERED ist.

Die Ausgabe, die angezeigt wird, folgt:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

### Beispiel 2-Anforderung einer Trace-Route-Antwortnachricht anfordern

Generieren und Bereitstellung einer Trace-Route-Nachricht an die Zielwarteschlange

In diesem Beispiel stellt die IBM MQ-Anwendung für die Routenanzeige eine Verbindung zu Warteschlangenmanager QM1 her und wird dazu verwendet, eine Trace-Route-Nachricht zu generieren und an die Zielwarteschlange TARGET.Q auf dem fernen Warteschlangenmanager QM2 zu übergeben. Die erforderliche Option wird so angegeben, dass die Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden. Bei der Ankunft in der Zielwarteschlange wird eine Trace-Route-Antwortnachricht angefordert, und die Trace-Route-Nachricht wird gelöscht.

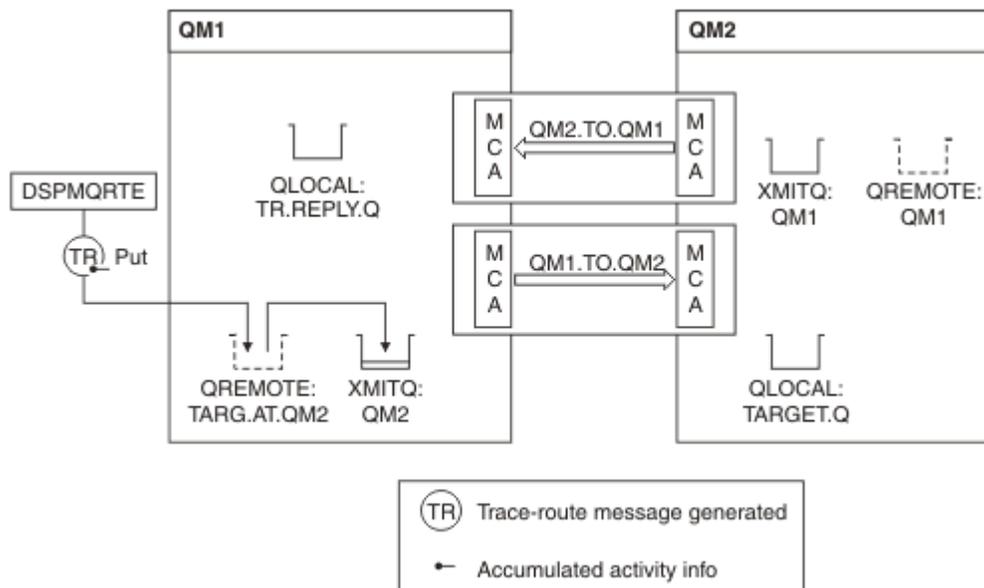


Abbildung 13. Anfordern einer Trace-Route-Antwortnachricht, Diagramm 1

- Das Attribut ROUTEREC der einzelnen WS-Manager (QM1 und QM2) wird auf MSG gesetzt.
- Der folgende Befehl wird ausgegeben:



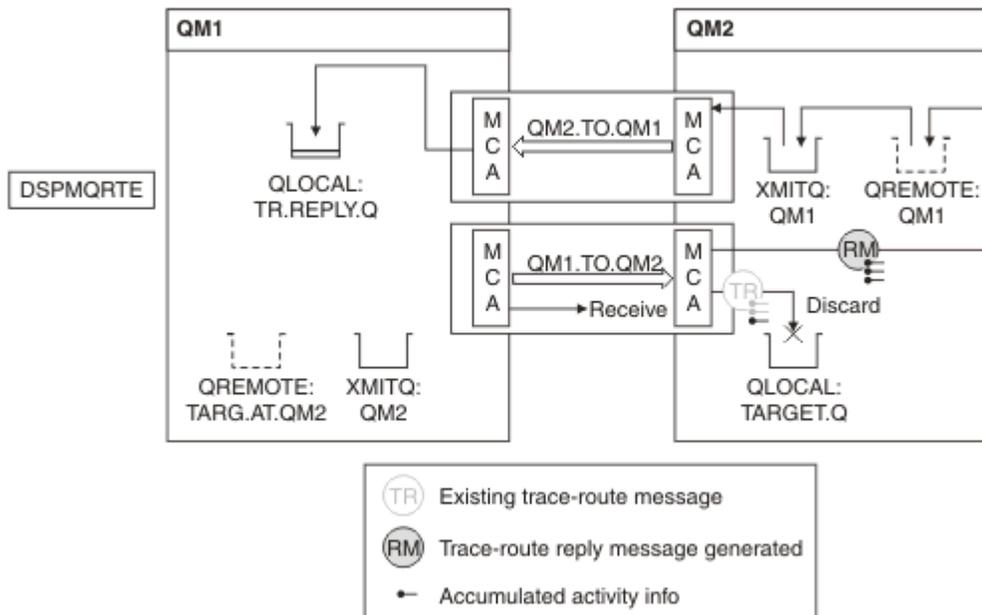


Abbildung 15. Antwort auf Trace-Route-Antwortnachricht anfordern, Diagramm 3

- Die Nachricht ist eine Trace-Route-Nachricht, daher beginnt der empfangende MCA mit dem Aufzeichnen von Informationen über die Aktivität.
- Das WS-Manager-Attribut ROUTEREC auf WSM2 ist nicht DISABLED, daher sammelt der MCA die Informationen in der Nachricht.
- Die Zielwarteschlange ist eine lokale Warteschlange, daher wird die Nachricht mit dem Feedback MQFB\_NOT\_DELIVERED in Übereinstimmung mit dem Wert des Parameters Deliver in der PCF-Gruppe TraceRoute gelöscht.
- Dies ist die letzte Aktivität, die in der Nachricht ausgeführt wird, und weil das WS-Manager-Attribut ROUTEREC auf QM1 nicht DISABLED ist, generiert der MCA eine Trace-Route-Antwortnachricht entsprechend dem Wert für "Accumulate". Der Wert von ROUTEREC ist MSG, daher wird die Antwortnachricht in die Antwortwarteschlange gestellt. Die Antwortnachricht enthält alle aufgelaufenen Aktivitätsinformationen aus der Trace-Route-Nachricht.

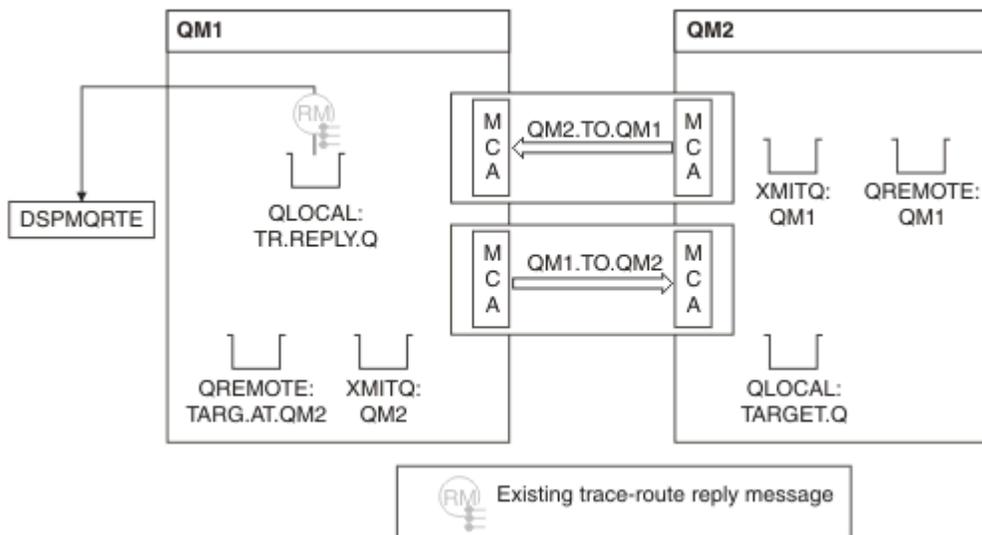


Abbildung 16. Antwort auf Trace-Route-Antwortnachricht anfordern, Diagramm 4

- In der Zwischenzeit wartet DSPMQRTE auf die Antwort auf die Trace-Route-Antwortnachricht, um in die Antwortwarteschlange zurückzukehren. Wenn er zurückkehrt, analysiert DSPMQRTE jede Aktivität, die er enthält, und druckt sie aus. Die letzte Operation ist eine Löschoperation. DSPMQRTE beendet, nachdem er gedruckt wurde.

Die Ausgabe, die angezeigt wird, folgt:

```

AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
TR.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.

```

### Beispiel 3-Übergabe von Aktivitätsberichten an die Systemwarteschlange

Es wird erkannt, wenn Aktivitätsberichte an andere Warteschlangen als die Warteschlange für zu beantwortende Nachrichten übergeben werden, und mit der IBM MQ-Anwendung für die Routenanzeige können Aktivitätsberichte aus der anderen Warteschlange gelesen werden.

Dieses Beispiel ist identisch mit dem „Beispiel 1-Aktivitätenberichte anfordern“ auf Seite 91, mit der Ausnahme, dass WSM2 jetzt den Wert des Attributs ACTIVREC queue management auf QUEUE gesetzt hat. Der Kanal QM1.TO.QM2 muss neu gestartet worden sein, damit diese Funktion wirksam wird.

In diesem Beispiel wird gezeigt, wie erkannt wird, wenn Aktivitätsberichte an andere Warteschlangen als die Warteschlange für Antwortantworten zugestellt werden. Nach dem Erkennen werden Aktivitätsberichte aus der anderen Warteschlange mit der IBM MQ-Anwendung zur Routenanzeige gelesen.

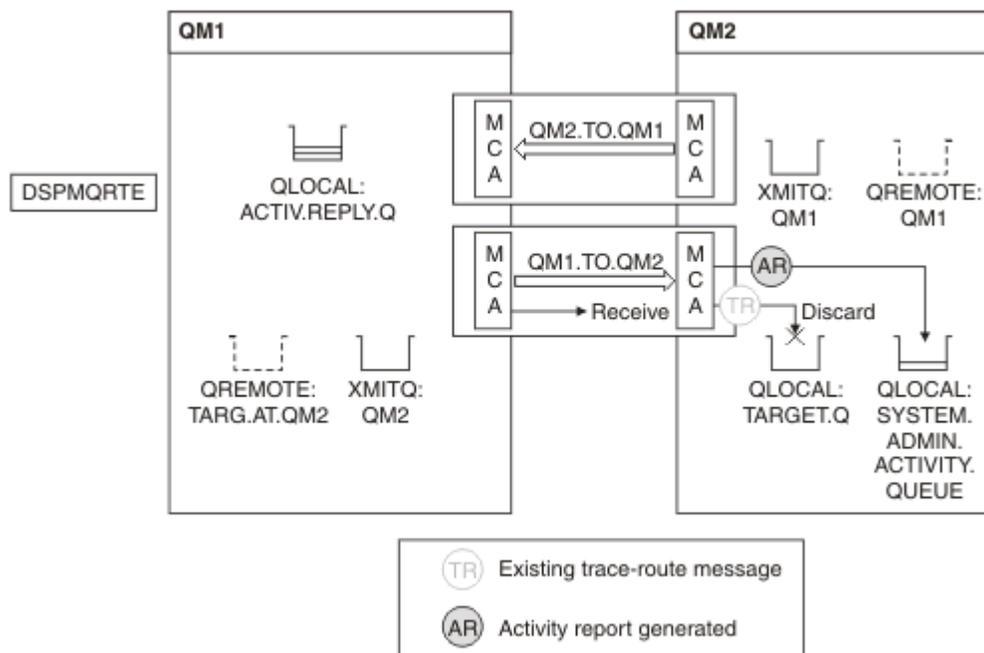


Abbildung 17. Aktivitätsberichte an die Systemwarteschlange übergeben, Diagramm 1

- Die Nachricht ist eine Trace-Route-Nachricht, daher beginnt der empfangende MCA mit dem Aufzeichnen von Informationen über die Aktivität.
- Der Wert des Warteschlangenmanagerattributs ACTIVREC auf QM2 ist jetzt QUEUE, daher generiert der Nachrichtenkanalmanager einen Aktivitätsbericht, stellt ihn jedoch in die Systemwarteschlange (SYSTEM.ADMIN.ACTIVITY.QUEUE) und nicht in die Antwortwarteschlange (ACTIV.REPLY.Q).

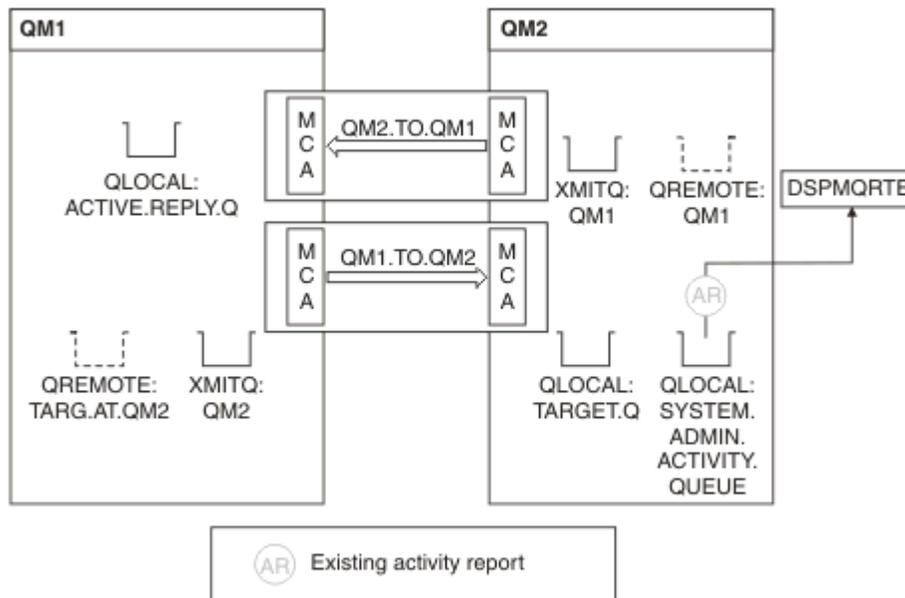


Abbildung 18. Aktivitätsberichte an die Systemwarteschlange übergeben, Diagramm 2

- In der Zwischenzeit hat DSPMQRTE auf Aktivitätsberichte gewartet, um in ACTIV.REPLY.Q. einzutreffen. Nur zwei kommen. DSPMQRTE wartet weiterhin 120 Sekunden, da es den Anschein hat, dass die Route noch nicht abgeschlossen ist.

Die Ausgabe, die angezeigt wird, folgt:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
ACTIV.REPLY.Q -v outline identifiers'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
```

```
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'
```

```
Operation:
  OperationType: Put
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'TARG.AT.QM2'
  ResolvedQName: 'QM2'
  RemoteQName: 'TARGET.Q'
  RemoteQMgrName: 'QM2'
```

```
-----
Activity:
  ApplName: 'cann\output\bin\runmqchl.EXE'
```

```
Operation:
  OperationType: Get
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001505'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001502'
```

```
EmbeddedMQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'QM2'
  ResolvedQName: 'QM2'
```

```
Operation:
  OperationType: Send
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  RemoteQMgrName: 'QM2'
  ChannelName: 'QM1.TO.QM2'
  ChannelType: Sender
  XmitQName: 'QM2'
```

```
-----
AMQ8652: DSPMQRTE command has finished.
```

- Die letzte Operation, die DSPMQRTE beobachtet hat, war eine Sendung, daher wird der Kanal ausgeführt. Jetzt müssen wir herausfinden, warum wir keine weiteren Aktivitätsberichte vom WS-Manager QM2 erhalten haben (wie in RemoteQMgrName angegeben).
- Um zu überprüfen, ob Aktivitätsinformationen in der Systemwarteschlange vorhanden sind, starten Sie DSPMQRTE auf QM2, um zu versuchen, weitere Aktivitätsberichte zu erfassen. Verwenden Sie den folgenden Befehl, um DSPMQRTE zu starten:

```
dspmqrte -m QM2 -q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C9154220001502 -v outline
```

Dabei gilt: 414D51204C41524745512020202020A3C9154220001502 ist die MsgId der Trace-Route-Nachricht, die in die Datei gestellt wurde.

- DSPMQRTE führt dann erneut eine Folge von MQGETs aus und wartet auf Antworten in der Systemaktivitätswarteschlange, die sich auf die Trace-Route-Nachricht mit der angegebenen ID bezieht.

- DSPMQRTE ruft einen weiteren Aktivitätsbericht ab, den er anzeigt. DSPMQRTE stellt fest, dass die vorangegangenen Aktivitätsberichte fehlen, und zeigt eine Nachricht an, die dies angibt. Wir wissen aber schon über diesen Teil der Strecke.

Die Ausgabe, die angezeigt wird, folgt:

```

AMQ8653: DSPMQRTE command started with options '-m QM2
-q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C915420001502 -v outline'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----

Activity:
  Activity information unavailable.

-----

Activity:
  ApplName: 'cann\output\bin\AMQRMPPA.EXE'

  Operation:
    OperationType: Receive
    QMgrName: 'QM2'
    RemoteQMGrName: 'QM1'
    ChannelName: 'QM1.TO.QM2'
    ChannelType: Receiver

  Operation:
    OperationType: Discard
    QMgrName: 'QM2'
    QName: 'TARGET.Q'
    Feedback: NotDelivered

-----

AMQ8652: DSPMQRTE command has finished.

```

- Dieser Aktivitätsbericht zeigt an, dass die Routeninformationen jetzt abgeschlossen sind. Es ist kein Fehler aufgetreten.
- Nur weil Leitweginformationen nicht verfügbar sind oder weil DSPMQRTE nicht die gesamte Route anzeigen kann, bedeutet dies nicht, dass die Nachricht nicht zugestellt wurde. Beispielsweise können die WS-Manager-Attribute unterschiedlicher WS-Manager unterschiedlich sein, oder es ist möglicherweise keine Antwortwarteschlange definiert, um die Antwort zurückerhalten zu können.

#### *Beispiel 4-Kanalfehler diagnostizieren*

Diagnostizieren eines Problems, bei dem die Trace-Route-Nachricht die Zielwarteschlange nicht erreicht

In diesem Beispiel stellt die IBM MQ-Anwendung für die Routenanzeige eine Verbindung zu Warteschlangenmanager QM1 her, generiert eine Trace-Route-Nachricht und versucht anschließend, die Nachricht an die Zielwarteschlange TARGET.Q auf dem fernen Warteschlangenmanager QM2 zu übergeben. In diesem Beispiel erreicht die Trace-Route-Nachricht nicht die Zielwarteschlange. Der verfügbare Aktivitätsbericht wird zur Diagnose des Problems verwendet.

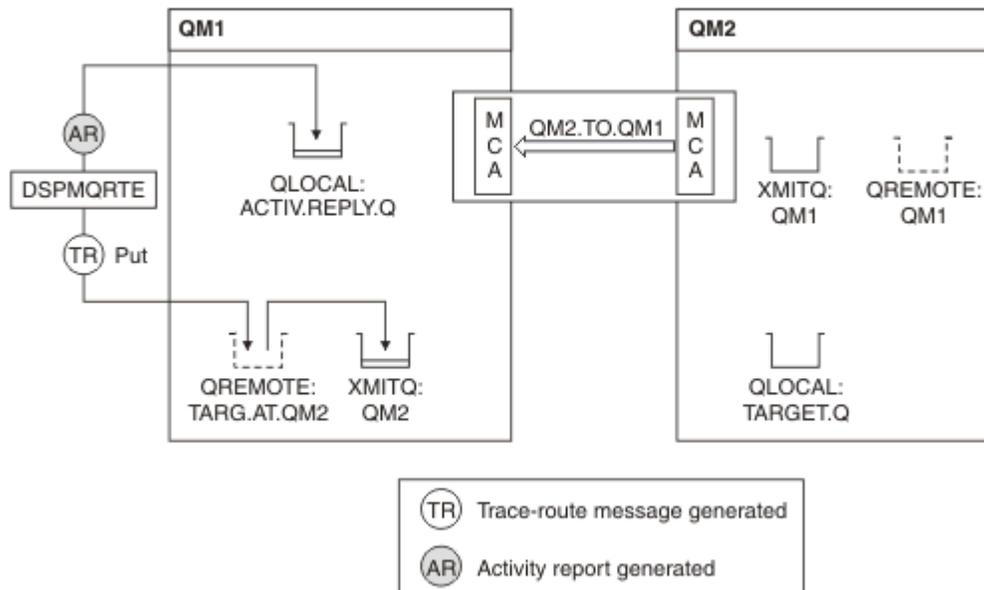


Abbildung 19. Kanalfehler diagnostizieren

- In diesem Beispiel wird der Kanal QM1.TO.QM2 nicht ausgeführt.
- Mit dem Befehl DSPMQRTE wird eine Trace-Route-Nachricht (wie in Beispiel 1) in die Zielwarteschlange gestellt und ein Aktivitätsbericht generiert.
- Es gibt keinen MCA zum Abrufen der Nachricht aus der Übertragungswarteschlange (WSM2). Daher ist dies der einzige Aktivitätsbericht, den DSPMQRTE aus der Antwortwarteschlange zurückerhält. Diesmal deutet die Tatsache, dass die Route nicht vollständig ist, auf ein Problem hin. Der Administrator kann die in ResolvedQName gefundene Übertragungswarteschlange verwenden, um zu ermitteln, warum die Übertragungswarteschlange nicht gewartet wird.

Die Ausgabe, die angezeigt wird, folgt:

```

AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q -v outline'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'

  Operation:
    OperationType: Put
    QMgrName: 'QM1'
    QName: 'TARG.AT.QM2'
    ResolvedQName: 'QM2'
    RemoteQName: 'TARGET.Q'
    RemoteQMgrName: 'QM2'
-----
AMQ8652: DSPMQRTE command has finished.

```

## Aktivitätsberichtsreferenz

Verwenden Sie diese Seite, um eine Übersicht über das Nachrichtenformat des Aktivitätsberichts zu erhalten. Die Nachrichtendaten des Aktivitätsberichts enthalten die Parameter, die die Aktivität beschreiben.

### Aktivitätsberichtsformat

Aktivitätsberichte sind standardmäßige IBM MQ-Berichtsnachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Aktivitätsberichte sind PCF-Nachrichten, die von Anwendungen gene-

riert werden, die eine Aktivität im Namen einer Nachricht ausgeführt haben, da sie über ein Warteschlangenmanagernetzwerk weitergeleitet wurde.

Aktivitätsberichte enthalten die folgenden Informationen:

**Ein Nachrichtendeskriptor**

Eine MQMD-Struktur

**Nachrichtendaten**

Befasst sich wie folgt:

- Ein eingebetteter PCF-Header (MQEPH).
- Nachrichtendaten des Aktivitätsberichts.

Die Nachrichtendaten des Aktivitätsberichts bestehen aus der *Activity* -PCF-Gruppe und, wenn sie für eine Trace-Route-Nachricht generiert wurde, die *TraceRoute* -PCF-Gruppe.

Tabelle 19 auf Seite 103 zeigt die Struktur dieser Berichte, einschließlich der Parameter, die nur unter bestimmten Bedingungen zurückgegeben werden.

Tabelle 19. Aktivitätsberichtsformat

MQMD-Struktur	Eingebetteter PCF-Header MQEPH-Struktur	Aktivitätsbericht-Nachrichtendaten
Struktur-ID Strukturversion Berichtsoptionen Nachrichtentyp Verfallszeit Feedback Encoding ID des codierten Zeichensatzes (CCSID). Nachrichtenformat Priority Permanenz Nachrichten-ID Korrelations-ID Rücksetzungszähler Empfangswarteschlange für Antworten Warteschlangenmanager für Antwortwarteschlange Benutzer-ID Abrechnung Anwendungsidentitätsdaten Anwendungstyp Anwendungsname Aufnahmezeit Ursprungsdaten der Anwendung Gruppen-ID Nachrichtenfolgennummer Offset Nachrichtenmarkierungen Ursprüngliche Länge	Struktur-ID Strukturversion Strukturlänge Encoding ID des codierten Zeichensatzes (CCSID). Nachrichtenformat Flags PCF-Header (MQCFH) Strukturtyp Strukturlänge Strukturversion Befehls-ID Nachrichtenfolgennummer Steueroptionen Beendigungscode Ursachencode Parameteranzahl	Aktivität Anwendungsname der Aktivität Aktivitätsanwendungstyp Aktivitätsbeschreibung Operation Operationstyp Operationstermin Operationszeit Nachricht Nachrichtenlänge MQMD <sup>8</sup> EmbeddedMQMD Name des Warteschlangenmanagers Name der Gruppe mit gemeinsamer Warteschlange Warteschlangenname <sup>1</sup> <sup>2</sup> <sup>3</sup> <sup>7</sup> Aufgelöster Warteschlangenname <sup>1</sup> <sup>3</sup> <sup>7</sup> Name der fernen Warteschlange <sup>3</sup> <sup>7</sup> Ferner Warteschlangenmanagername <sup>2</sup> <sup>3</sup> <sup>4</sup> <sup>5</sup> <sup>7</sup> Subskriptionsebene <sup>9</sup> Subskriptionskennung <sup>9</sup> Feedback <sup>2</sup> <sup>10</sup> Kanalname <sup>4</sup> <sup>5</sup> Kanaltyp <sup>4</sup> <sup>5</sup> Name der Übertragungswarteschlange <sup>5</sup> TraceRoute <sup>6</sup> Detail Aufgezeichnete Aktivitäten Nicht aufgezeichnete Aktivitäten Diskontinuitätszähler Max. Aktivitäten Kumulieren Deliver

**Anmerkungen:**

1. Wird für Get- und Browse-Operationen zurückgegeben.
2. Wird für Discard-Operationen zurückgegeben.
3. Zurückgegeben für Put-, Put-Reply- und Put-Report-Operationen.

4. Zurückgegeben für Empfangsoperationen.
5. Wird für Sendeoperationen zurückgegeben.
6. Wird für Trace-Route-Nachrichten zurückgegeben.
7. Nicht zurückgegeben für 'Put' -Operationen in ein Thema, das in Publish-Aktivitäten enthalten ist.
8. Für ausgeschlossene Publish-Operationen nicht zurückgegeben. Für Publish- und Discarded Publish-Operationen, die eine Untergruppe von Parametern enthalten.
9. Zurückgegeben für Publish, Discarded Publish und Excluded Publish-Operationen.
10. Zurückgegeben bei Veröffentlichen von Veröffentlichen und Ausgeschlossenen Publish-Operationen.

### **MQMD des Aktivitätsberichts (Nachrichtendeskriptor)**

Verwenden Sie diese Seite, um die Werte anzuzeigen, die in der MQMD-Struktur für einen Aktivitätsbericht enthalten sind.

#### **StrucId**

Struktur-ID:

#### **Datentyp**

MQCHAR4

#### **Wert**

MQMD\_STRUC\_ID.

#### **Version**

Strukturversionsnummer

#### **Datentyp**

MQLONG

#### **Werte**

Kopiert aus dem ursprünglichen Nachrichtendeskriptor. Mögliche Werte:

#### **MQMD\_VERSION\_1**

Version-1 Nachrichtendeskriptorstruktur, die in allen Umgebungen unterstützt wird.

#### **MQMD\_VERSION\_2**

Version 2 der Nachrichtendeskriptorstruktur, die in den folgenden Umgebungen unterstützt wird:

-  AIX
-  IBM i
-  Linux
-  Solaris
-  Windows
-  z/OS

und auf allen IBM MQ MQI clients, die mit diesen Systemen verbunden sind.

#### **Report**

Optionen für weitere Berichtsnachrichten

#### **Datentyp**

MQLONG

#### **Wert**

Wenn MQRO\_PASS\_DISCARD\_AND\_EXPIRY oder MQRO\_DISCARD\_MSG im Feld *Report* des ursprünglichen Nachrichtendeskriptors angegeben wurden:

#### **MQRO\_DISCARD**

Der Bericht wird gelöscht, wenn er nicht an die Zielwarteschlange übergeben werden kann.

Andernfalls:

**MQRO\_NONE**

Kein Bericht erforderlich

**MsgType**

Gibt den Typ der Nachricht an.

**Datentyp**

MQLONG

**Wert**

MQMT\_REPORT

**Expiry**

Lebensdauer der Berichtsnachricht

**Datentyp**

MQLONG

**Wert**

Wenn das Feld *Report* im ursprünglichen Nachrichtendeskriptor als MQRO\_PASS\_DISCARD\_AND\_EXPIRY angegeben wird, wird die verbleibende Ablaufzeit von der ursprünglichen Nachricht verwendet.

Andernfalls:

**MQEI\_UNLIMITED**

Der Bericht hat keine Ablaufzeit.

**Feedback**

Beschreibung: Rückmeldungs- oder Ursachencode.

Datentyp: MQLONG.

Wert: **MQFB\_ACTIVITY**  
Tätigkeitsbericht.

**Encoding**

Beschreibung: Numerische Codierung von Berichtsnachrichtendaten.

Datentyp: MQLONG.

Wert: MQENC\_NATIVE.

**CodedCharSetId**

Beschreibung: Zeichensatzes-ID der Berichtsnachrichtendaten.

Datentyp: MQLONG.

Wert: Legen Sie die entsprechende Gruppe fest.

**Format**

Beschreibung: Formatname der Berichtsnachrichtendaten

Datentyp: MQCHAR8.

Wert: **MQFMT\_EMBEDDED\_PCF**  
Eingebettete PCF-Nachricht

**Priority**

Beschreibung: Nachrichtenpriorität melden.

Datentyp: MQLONG.  
Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **Persistence**

Beschreibung: Nachrichtenpersistenz berichten.  
Datentyp: MQLONG.  
Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **MsgId**

Beschreibung: Nachrichten-ID.  
Datentyp: MQBYTE24.  
Werte: Wenn das Feld *Report* im ursprünglichen Nachrichtendeskriptor als 'MQRO\_PASS\_MSG\_ID' angegeben ist, wird die Nachrichten-ID aus der ursprünglichen Nachricht verwendet.  
Andernfalls wird vom WS-Manager ein eindeutiger Wert generiert.

### **CorrelId**

Beschreibung: Korrelations-ID.  
Datentyp: MQBYTE24.  
Wert: Wenn das Feld *Report* im ursprünglichen Nachrichtendeskriptor als 'MQRO\_PASS\_CORREL\_ID' angegeben ist, wird die Korrelations-ID aus der ursprünglichen Nachricht verwendet.  
Andernfalls wird die Nachrichten-ID aus der ursprünglichen Nachricht kopiert.

### **BackoutCount**

Beschreibung: Zurücksetzungszähler.  
Datentyp: MQLONG.  
Wert: 0.

### **ReplyToQ**

Beschreibung: Name der Antwortwarteschlange.  
Datentyp: MQCHAR48.  
Werte: Leer.

### **ReplyToQMgr**

Beschreibung: Antwort-Warteschlangenmanagername  
Datentyp: MQCHAR48.  
Wert: Der Name des Warteschlangenmanagers, der die Berichtsnachricht generiert hat.

### **UserIdentifier**

Beschreibung: Die Benutzer-ID der Anwendung, die die Berichtsnachricht generiert hat.  
Datentyp: MQCHAR12.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **AccountingToken**

Beschreibung: Abrechnungstoken, das es einer Anwendung ermöglicht, die Arbeit zu berechnen, die als Ergebnis der Nachricht ausgeführt wurde.

Datentyp: MQBYTE32.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **ApplIdentityData**

Beschreibung: Identitätsbezogene Anwendungsdaten.

Datentyp: MQCHAR32.

Werte: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **PutApplType**

Beschreibung: Typ der Anwendung, die die Berichtsnachricht eingibt.

Datentyp: MQLONG.

Wert: **MQAT\_QMGR**  
Nachricht des vom WS-Manager generierten Nachrichten.

### **PutApplName**

Beschreibung: Name der Anwendung, die die Berichtsnachricht eingibt.

Datentyp: MQCHAR28.

Wert: Entweder die ersten 28 Byte des Warteschlangenmanagernamens oder der Name des MCA, der die Berichtsnachricht generiert hat.

### **PutDate**

Beschreibung: Datum, an dem die Nachricht eingereicht wurde.

Datentyp: MQCHAR8.

Wert: Wie vom WS-Manager generiert.

### **PutTime**

Beschreibung: Zeitpunkt, zu dem die Nachricht eingereicht wurde.

Datentyp: MQCHAR8.

Wert: Wie vom WS-Manager generiert.

### **ApplOriginData**

Beschreibung: Anwendungsdaten zum Ursprung

Datentyp: MQCHAR4.

Wert: Leer.

Wenn *Version* MQMD\_VERSION\_2 ist, sind die folgenden zusätzlichen Felder vorhanden:

### **GroupId**

Beschreibung:	Gibt an, zu welcher Nachrichtengruppe oder logischen Nachricht die physische Nachricht gehört.
Datentyp:	MQBYTE24.
Wert:	Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **MsgSeqNumber**

Beschreibung:	Folgenummer der logischen Nachricht in einer Gruppe
Datentyp:	MQLONG.
Wert:	Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **Offset**

Beschreibung:	Relative Adresse von Daten in einer physischen Nachricht ab dem Anfang der logischen Nachricht.
Datentyp:	MQLONG.
Wert:	Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **MsgFlags**

Beschreibung:	Nachrichtenmarkierungen, die Attribute der Nachricht angeben oder ihre Verarbeitung steuern.
Datentyp:	MQLONG.
Wert:	Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **OriginalLength**

Beschreibung:	Länge der ursprünglichen Nachricht
Datentyp:	MQLONG.
Wert:	Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

### **Aktivitätenbericht MQEPH (eingebetteter PCF-Header)**

Verwenden Sie diese Seite, um die Werte anzuzeigen, die in der MQEPH-Struktur für einen Aktivitätsbericht enthalten sind.

Die MQEPH-Struktur enthält eine Beschreibung der PCF-Informationen, die die Nachrichtendaten eines Aktivitätsberichts enthalten, sowie die Anwendungsnachrichtendaten, die auf sie folgen.

Für einen Aktivitätsbericht enthält die MQEPH-Struktur die folgenden Werte:

### **StrucId**

Beschreibung:	Struktur-ID
Datentyp:	MQCHAR4.
Wert:	MQEPH_STRUC_ID.

### **Version**

Beschreibung:	Strukturversionsnummer.
Datentyp:	MQLONG.
Werte:	MQEPH_VERSION_1.

### **StrucLength**

Beschreibung:	Strukturlänge.
Datentyp:	MQLONG.
Wert:	Gesamtlänge der Struktur einschließlich der PCF-Parameterstrukturen, die auf diese Struktur folgen.

### **Encoding**

Beschreibung:	Numerische Codierung der Nachrichtendaten, die auf die letzte PCF-Parameterstruktur folgen.
Datentyp:	MQLONG.
Wert:	Wenn Daten aus den ursprünglichen Anwendungsnachrichtendaten in der Berichtsnachricht enthalten sind, wird der Wert aus dem Feld <i>Encoding</i> des ursprünglichen Nachrichtendeskriptors kopiert. Andernfalls 0.

### **CodedCharSetId**

Beschreibung:	Zeichensatzes-ID der Nachrichtendaten, die auf die letzte PCF-Parameterstruktur folgen.
Datentyp:	MQLONG.
Wert:	Wenn Daten aus den ursprünglichen Anwendungsnachrichtendaten in der Berichtsnachricht enthalten sind, wird der Wert aus dem Feld <i>CodedCharSetId</i> des ursprünglichen Nachrichtendeskriptors kopiert. Andernfalls MQCCSI_UNDEFINED.

### **Format**

Beschreibung:	Formatname der Nachrichtendaten, die auf die letzte PCF-Parameterstruktur folgen.
Datentyp:	MQCHAR8.
Wert:	Wenn Daten aus den ursprünglichen Anwendungsnachrichtendaten in der Berichtsnachricht enthalten sind, wird der Wert aus dem Feld <i>Format</i> des ursprünglichen Nachrichtendeskriptors kopiert. Andernfalls MQFMT_NONE.

### **Flags**

Beschreibung:	Flags, die Attribute der Struktur angeben oder ihre Verarbeitung steuern.
Datentyp:	MQLONG.
Wert:	<b>MQEPH_CCSID_EMBEDDED</b> Gibt an, dass der Zeichensatz der Parameter, die Zeichendaten enthalten, einzeln im Feld <i>CodedCharSetId</i> in jeder Struktur angegeben wird.

### **PCFHeader**

Beschreibung:	Header für Programmierbares Befehlsformat
Datentyp:	MQCFH.
Wert:	Siehe <u>„Aktivitätenbericht MQCFH (PCF-Header)“</u> auf Seite 110.

## **Aktivitätenbericht MQCFH (PCF-Header)**

Verwenden Sie diese Seite, um die PCF-Werte anzuzeigen, die in der MQCFH-Struktur für einen Aktivitätsbericht enthalten sind.

Für einen Aktivitätsbericht enthält die MQCFH-Struktur die folgenden Werte:

### **Type**

Beschreibung:	Strukturtyp, der den Inhalt der Berichtsnachricht identifiziert.
Datentyp:	MQLONG.
Wert:	<b>MQCFT_REPORT</b> Nachricht ist ein Bericht.

### **StrucLength**

Beschreibung:	Strukturlänge.
Datentyp:	MQLONG.
Wert:	<b>MQCFH_STRUC_LENGTH</b> Länge der MQCFH-Struktur in Byte.

### **Version**

Beschreibung:	Strukturversionsnummer.
Datentyp:	MQLONG.
Werte:	MQCFH_VERSION_3

### **Command**

Beschreibung:	Befehls-ID. Gibt die Kategorie der Nachricht an.
Datentyp:	MQLONG.
Werte:	<b>MQCMD_ACTIVITY_MSG</b> Nachrichtenaktivität.

### **MsgSeqNumber**

Beschreibung:	Nachrichtenfolgennummer. Dies ist die Folgennummer der Nachricht in einer Gruppe zusammengehöriger Nachrichten.
Datentyp:	MQLONG.
Werte:	1.

### **Control**

Beschreibung:	Steueroptionen.
Datentyp:	MQLONG.
Werte:	MQCFC_LAST.

### **CompCode**

Beschreibung:	Beendigungscode.
Datentyp:	MQLONG.
Werte:	MQCC_OK.

## **Reason**

Beschreibung:	Beendigungscode für Ursachencode.
Datentyp:	MQLONG.
Werte:	MQRC_NONE.

## **ParameterCount**

Beschreibung:	Anzahl der Parameterstrukturen. Dies ist die Anzahl der Parameterstrukturen, die auf die MQCFH-Struktur folgen. Eine Gruppenstruktur (MQCFGR) und die darin enthaltenen Parameterstrukturen werden nur als eine einzige Struktur gezählt.
Datentyp:	MQLONG.
Werte:	1 oder größer.

## **Aktivitätsbericht-Nachrichtendaten**

Verwenden Sie diese Seite, um die Parameter anzuzeigen, die von der *Activity* -PCF-Gruppe in einer Aktivitätsberichtsrichtnachricht enthalten sind. Einige Parameter werden nur zurückgegeben, wenn bestimmte Operationen ausgeführt wurden.

Die Nachrichtendaten des Aktivitätsberichts bestehen aus der *Activity* -PCF-Gruppe und, wenn sie für eine Trace-Route-Nachricht generiert wurde, die *TraceRoute* -PCF-Gruppe. Die *Activity* -PCF-Gruppe ist in diesem Abschnitt ausführlich beschrieben.

Einige Parameter, die als operationsspezifische Aktivitätsberichtsrichtnachrichtendaten beschrieben werden, werden nur zurückgegeben, wenn bestimmte Operationen ausgeführt wurden.

Für einen Aktivitätsbericht enthält die Aktivitätsberichtsrichtnachrichtendaten die folgenden Parameter:

### **Activity**

Beschreibung:	Gruppierte Parameter, die die Aktivität beschreiben.
ID:	MQGACF_ACTIVITY.
Datentyp:	MQCFGR.
In PCF-Gruppe enthalten:	Keine.
Parameter in PCF-Gruppe:	<i>ActivityApplName</i> <i>ActivityApplType</i> <i>ActivityDescription</i> <i>Operation</i> <i>TraceRoute</i>
Zurückgegeben:	Immer.

### **ActivityApplName**

Beschreibung:	Name der Anwendung, die die Aktivität ausgeführt hat.
ID:	MQCACF_APPL_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Activity</i> .
Maximale Länge:	MQ_APPL_NAME_LENGTH.
Zurückgegeben:	Immer.

### **ActivityApplType**

Beschreibung:	Typ der Anwendung, die die Aktivität ausgeführt hat.
ID:	MQIA_APPL_TYPE.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Activity</i> .
Zurückgegeben:	Immer.

### **ActivityDescription**

Beschreibung:	Beschreibung der Aktivität, die von der Anwendung ausgeführt wird.
ID:	MQCACF_ACTIVITY_DESCRIPTION.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Activity</i> .
Maximale Länge:	64
Zurückgegeben:	Immer.

### **Operation**

Beschreibung:	Gruppierte Parameter, die eine Operation der Aktivität beschreiben.
ID:	MQGACF_OPERATION.
Datentyp:	MQCFGR.
In PCF-Gruppe enthalten:	<i>Activity</i> .
Parameter in PCF- Gruppe:	<i>OperationType</i> <i>OperationDate</i> <i>OperationTime</i> <i>Message</i> <i>QMgrName</i> <i>QSGName</i>

**Anmerkung:** Je nach Operationstyp werden in dieser Gruppe zusätzliche Parameter zurückgegeben. Diese zusätzlichen Parameter werden als Operation-spezifische Aktivitätsberichtsnachrichtendaten beschrieben.

Zurückgegeben: Eine *Operation* -PCF-Gruppe pro Operation in der Aktivität.

### **OperationType**

Beschreibung:	Typ der ausgeführten Operation.
ID:	MQIACF_OPERATION_TYPE.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation</i> .
Werte:	MQOPER_ *.
Zurückgegeben:	Immer.

### **OperationDate**

Beschreibung:	Datum, an dem die Operation ausgeführt wurde.
ID:	MQCACF_OPERATION_DATE.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_DATE_LENGTH.
Zurückgegeben:	Immer.

### **OperationTime**

Beschreibung:	Zeit, zu der die Operation ausgeführt wurde.
ID:	MQCACF_OPERATION_TIME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_TIME_LENGTH.
Zurückgegeben:	Immer.

### **Message**

Beschreibung:	Gruppierte Parameter, die die Nachricht beschreiben, die die Aktivität verursacht hat.
ID:	MQGACF_MESSAGE.
Datentyp:	MQCFGR.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Parameter in Gruppe:	<i>MsgLength</i> <i>MQMD</i> <i>EmbeddedMQMD</i>
Zurückgegeben:	Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

### **MsgLength**

Beschreibung:	Länge der Nachricht, die die Aktivität verursacht hat, bevor die Aktivität auftrat.
ID:	MQIACF_MSG_LENGTH.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Message.</i>
Zurückgegeben:	Immer.

### **MQMD**

Beschreibung:	Gruppierte Parameter, die sich auf den Nachrichtendeskriptor der Nachricht beziehen, die die Aktivität verursacht hat.
ID:	MQGACF_MQMD.

Datentyp:	MQCFGR.
In PCF-Gruppe enthalten:	<i>Message.</i>
Parameter in Gruppe:	<i>StrucId</i> <i>Version</i> <i>Report</i> <i>MsgType</i> <i>Expiry</i> <i>Feedback</i> <i>Encoding</i> <i>CodedCharSetId</i> <i>Format</i> <i>Priority</i> <i>Persistence</i> <i>MsgId</i> <i>CorrelId</i> <i>BackoutCount</i> <i>ReplyToQ</i> <i>ReplyToQMgr</i> <i>UserIdentifier</i> <i>AccountingToken</i> <i>ApplIdentityData</i> <i>PutApplType</i> <i>PutApplName</i> <i>PutDate</i> <i>PutTime</i> <i>ApplOriginData</i> <i>GroupId</i> <i>MsgSeqNumber</i> <i>Offset</i> <i>MsgFlags</i> <i>OriginalLength</i>

Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

### ***EmbeddedMQMD***

Beschreibung: Gruppierete Parameter, die den Nachrichtendeskriptor beschreiben, der in eine Nachricht in einer Übertragungswarteschlange eingebettet ist.

ID: MQGACF\_EMBEDDED\_MQMD.

Datentyp: MQCFGR.

In PCF-Gruppe  
enthalten: *Message.*

Parameter in Gruppe:	<i>StrucId</i> <i>Version</i> <i>Report</i> <i>MsgType</i> <i>Expiry</i> <i>Feedback</i> <i>Encoding</i> <i>CodedCharSetId</i> <i>Format</i> <i>Priority</i> <i>Persistence</i> <i>MsgId</i> <i>CorrelId</i> <i>BackoutCount</i> <i>ReplyToQ</i> <i>ReplyToQMgr</i> <i>UserIdentifier</i> <i>AccountingToken</i> <i>ApplIdentityData</i> <i>PutApplType</i> <i>PutApplName</i> <i>PutDate</i> <i>PutTime</i> <i>ApplOriginData</i> <i>GroupId</i> <i>MsgSeqNumber</i> <i>Offset</i> <i>MsgFlags</i> <i>OriginalLength</i>
----------------------	---

Zurückgegeben: Für Get-Operationen, bei denen die Warteschlange in eine Übertragungswarteschlange aufgelöst wird.

### **StrucId**

Beschreibung:	Struktur-ID
ID:	MQCACF_STRUC_ID.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Maximale Länge:	4.
Zurückgegeben:	Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **Version**

Beschreibung:	Strukturversionsnummer.
ID:	MQIACF_VERSION.
Datentyp:	MQCFIN.

In PCF-Gruppe enthalten: *MQMD* oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **Report**

Beschreibung: Optionen für Berichtsnachrichten.  
ID: MQIACF\_REPORT.  
Datentyp: MQCFIN.  
In PCF-Gruppe enthalten: *MQMD* oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **MsgType**

Beschreibung: Gibt den Typ der Nachricht an.  
ID: MQIACF\_MSG\_TYPE.  
Datentyp: MQCFIN.  
In PCF-Gruppe enthalten: *MQMD* oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **Expiry**

Beschreibung: Lebensdauer der Nachricht  
ID: MQIACF\_EXPIRY.  
Datentyp: MQCFIN.  
In PCF-Gruppe enthalten: *MQMD* oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **Feedback**

Beschreibung: Rückmeldungs- oder Ursachencode.  
ID: MQIACF\_FEEDBACK.  
Datentyp: MQCFIN.  
In PCF-Gruppe enthalten: *MQMD* oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **Encoding**

Beschreibung: Numerische Codierung von Nachrichtendaten.  
ID: MQIACF\_ENCODING.

Datentyp: MQCFIN.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **CodedCharSetId**

Beschreibung: Zeichensatz-ID der Nachrichtendaten.  
ID: MQIA\_CODED\_CHAR\_SET\_ID.  
Datentyp: MQCFIN.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **Format**

Beschreibung: Name des Formats von Nachrichtendaten  
ID: MQCACH\_FORMAT\_NAME.  
Datentyp: MQCFST.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Maximale Länge: MQ\_FORMAT\_LENGTH.  
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

### **Priority**

Beschreibung: Nachrichtenpriorität  
ID: MQIACF\_PRIORITY.  
Datentyp: MQCFIN.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

### **Persistence**

Beschreibung: Nachrichtenpersistenz.  
ID: MQIACF\_PERSISTENCE.  
Datentyp: MQCFIN.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

### **MsgId**

Beschreibung: Nachrichten-ID.  
ID: MQBACF\_MSG\_ID.

Datentyp: MQCFBS.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Maximale Länge: MQ\_MSG\_ID\_LENGTH.  
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

### **CorrelId**

Beschreibung: Korrelations-ID.  
ID: MQBACF\_CORREL\_ID.  
Datentyp: MQCFBS.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Maximale Länge: MQ\_CORREL\_ID\_LENGTH.  
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

### **BackoutCount**

Beschreibung: Zurücksetzungszähler.  
ID: MQIACF\_BACKOUT\_COUNT.  
Datentyp: MQCFIN.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **ReplyToQ**

Beschreibung: Name der Antwortwarteschlange.  
ID: MQCACF\_REPLY\_TO\_Q.  
Datentyp: MQCFST.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Maximale Länge: MQ\_Q\_NAME\_LENGTH.  
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

### **ReplyToQMGr**

Beschreibung: Antwort-Warteschlangenmanagername  
ID: MQCACF\_REPLY\_TO\_Q\_MGR.  
Datentyp: MQCFST.  
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.  
Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH.  
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

### **UserIdentifier**

Beschreibung:	Die Benutzer-ID der Anwendung, die die Nachricht erstellt hat.
ID:	MQCACF_USER_IDENTIFIER.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Maximale Länge:	MQ_USER_ID_LENGTH.
Zurückgegeben:	Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen'.

### **AccountingToken**

Beschreibung:	Abrechnungstoken, das es einer Anwendung ermöglicht, die Arbeit zu berechnen, die als Ergebnis der Nachricht ausgeführt wurde.
ID:	MQBACF_ACCOUNTING_TOKEN.
Datentyp:	MQCFBS.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Maximale Länge:	MQ_ACCOUNTING_TOKEN_LENGTH.
Zurückgegeben:	Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen'.

### **ApplIdentityData**

Beschreibung:	Identitätsbezogene Anwendungsdaten.
ID:	MQCACF_APPL_IDENTITY_DATA.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Maximale Länge:	MQ_APPL_IDENTITY_DATA_LENGTH.
Zurückgegeben:	Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen'.

### **PutApplType**

Beschreibung:	Typ der Anwendung, die die Nachricht eingereicht hat.
ID:	MQIA_APPL_TYPE.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Zurückgegeben:	Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

### **PutApplName**

Beschreibung:	Name der Anwendung, die die Nachricht einreicht.
ID:	MQCACF_APPL_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .

Maximale Länge: MQ\_APPL\_NAME\_LENGTH.  
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

### **PutDate**

Beschreibung: Datum, an dem die Nachricht eingereicht wurde.  
ID: MQCACF\_PUT\_DATE.  
Datentyp: MQCFST.  
In PCF-Gruppe enthalten: MQMD oder EmbeddedMQMD.  
Maximale Länge: MQ\_PUT\_DATE\_LENGTH.  
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

### **PutTime**

Beschreibung: Zeitpunkt, zu dem die Nachricht eingereicht wurde.  
ID: MQCACF\_PUT\_TIME.  
Datentyp: MQCFST.  
In PCF-Gruppe enthalten: MQMD oder EmbeddedMQMD.  
Maximale Länge: MQ\_PUT\_TIME\_LENGTH.  
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

### **ApplOriginData**

Beschreibung: Anwendungsdaten zum Ursprung  
ID: MQCACF\_APPL\_ORIGIN\_DATA.  
Datentyp: MQCFST.  
In PCF-Gruppe enthalten: MQMD oder EmbeddedMQMD.  
Maximale Länge: MQ\_APPL\_ORIGIN\_DATA\_LENGTH.  
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

### **GroupId**

Beschreibung: Gibt an, zu welcher Nachrichtengruppe oder logischen Nachricht die physische Nachricht gehört.  
ID: MQBACF\_GROUP\_ID.  
Datentyp: MQCFBS.  
In PCF-Gruppe enthalten: MQMD oder EmbeddedMQMD.  
Maximale Länge: MQ\_GROUP\_ID\_LENGTH.

Zurückgegeben: Wenn der *Version* als MQMD\_VERSION\_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

### **MsgSeqNumber**

Beschreibung: Folgenummer der logischen Nachricht in einer Gruppe

ID: MQIACH\_MSG\_SEQUENCE\_NUMBER.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.

Zurückgegeben: Wenn *Version* als MQMD\_VERSION\_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

### **Offset**

Beschreibung: Relative Adresse von Daten in einer physischen Nachricht ab dem Anfang der logischen Nachricht.

ID: MQIACF\_OFFSET.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.

Zurückgegeben: Wenn *Version* als MQMD\_VERSION\_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

### **MsgFlags**

Beschreibung: Nachrichtenmarkierungen, die Attribute der Nachricht angeben oder ihre Verarbeitung steuern.

ID: MQIACF\_MSG\_FLAGS.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.

Zurückgegeben: Wenn *Version* als MQMD\_VERSION\_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

### **OriginalLength**

Beschreibung: Länge der ursprünglichen Nachricht

ID: MQIACF\_ORIGINAL\_LENGTH.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.

Zurückgegeben: Wenn *Version* als MQMD\_VERSION\_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

### **QMgrName**

Beschreibung: Der Name des Warteschlangenmanagers, in dem die Aktivität ausgeführt wurde.  
ID: MQCA\_Q\_MGR\_NAME.  
Datentyp: MQCFST.  
In PCF-Gruppe enthalten: *Operation*.  
Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH  
Zurückgegeben: Immer.

### **QSGName**

Beschreibung: Der Name der Gruppe mit gemeinsamer Warteschlange, zu der der Warteschlangenmanager gehört, auf dem die Aktivität ausgeführt wurde.  
ID: MQCA\_QSG\_NAME.  
Datentyp: MQCFST.  
In PCF-Gruppe enthalten: *Operation*.  
Maximale Länge: MQ\_QSG\_NAME\_LENGTH  
Zurückgegeben: Wenn die Aktivität in einem IBM MQ for z/OS-Warteschlangenmanager ausgeführt wurde.

### **TraceRoute**

Beschreibung: Gruppierte Parameter, die Attribute der Trace-Route-Nachricht angeben.  
ID: MQGACF\_TRACE\_ROUTE.  
Datentyp: MQCFGR.  
Enthalten in PCF-Gruppe: *Activity*.  
Parameter in Gruppe: *Detail*  
*RecordedActivities*  
*UnrecordedActivities*  
*DiscontinuityCount*  
*MaxActivities*  
*Accumulate*  
*Forward*  
*Deliver*  
Zurückgegeben: Wenn die Aktivität im Namen der Trace-Route-Nachricht ausgeführt wurde.

Die Werte der Parameter in der PCF-Gruppe von *TraceRoute* sind die Werte aus der Trace-Route-Nachricht zu dem Zeitpunkt, zu dem der Aktivitätsbericht generiert wurde.

### **Nachrichtendaten für operationsspezifische Aktivitätsberichte**

Verwenden Sie diese Seite, um die zusätzlichen PCF-Parameter anzuzeigen, die in einem Aktivitätsbericht in der PCF-Gruppe *Operation* zurückgegeben werden können, abhängig vom Wert des Parameters *OperationType*.

Die zusätzlichen Parameter variieren in Abhängigkeit von den folgenden Operationstypen:

### *Abrufen/Durchsuchen (MQOPER\_GET/MQOPER\_BROWSE)*

Die zusätzlichen Aktivitätsberichtsnachrichtendatenparameter, die in der PCF-Gruppe *Operation* für den Operationstyp Get/Browse (MQOPER\_GET/MQOPER\_BROWSE) zurückgegeben werden (eine Nachricht in einer Warteschlange wurde erhalten oder durchsucht).

#### **QName**

Beschreibung:	Der Name der Warteschlange, die geöffnet wurde.
ID:	MQCA_Q_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Immer.

#### **ResolvedQName**

Beschreibung:	Der Name, in den die geöffnete Warteschlange aufgelöst wird.
ID:	MQCACF_RESOLVED_Q_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Immer.

### *Discard (MQOPER\_DISCARD)*

Die zusätzlichen Nachrichtendatenparameter für Aktivitäten, die in der PCF-Gruppe *Operation* für den Operationstyp 'Discard (MQOPER\_DISCARD)' zurückgegeben werden (eine Nachricht wurde gelöscht).

#### **Feedback**

Beschreibung:	Die Ursache für die Nachricht, die gelöscht wird.
ID:	MQIACF_FEEDBACK.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Immer.

#### **QName**

Beschreibung:	Der Name der Warteschlange, die geöffnet wurde.
ID:	MQCA_Q_NAME.
Datentyp:	MQCFST.
Maximale Länge:	MQ_Q_NAME_LENGTH
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Wenn die Nachricht gelöscht wurde, weil sie nicht erfolgreich in eine Warteschlange gestellt wurde.

### **RemoteQMgrName**

Beschreibung:	Der Name des Warteschlangenmanagers, für den die Nachricht bestimmt war.
ID:	MQCA_REMOTE_Q_MGR_NAME.
Datentyp:	MQCFST.
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Wenn der Wert von <i>Feedback</i> MQFB_NOT_FORWARDED ist.

*Publish/Discarded Publish/Excluded Publish/Excluded Publish (MQOPER\_PUBLISH/MQOPER\_DISCARDED\_PUBLISH/MQOPER\_EXCLUDED\_PUBLISH)*

Die zusätzlichen Aktivitätsberichtsdatenparameter, die in der PCF-Gruppe *Operation* für den Operationstyp *Publish/Discarded Publish/Excluded Publish (MQOPER\_PUBLISH/MQOPER\_DISCARDED\_PUBLISH/MQOPER\_EXCLUDED\_PUBLISH)* zurückgegeben werden (eine *Publish/Subscribe*-Nachricht wurde zugestellt, verworfen oder ausgeschlossen).

### **SubId**

Beschreibung:	Die Subskriptionskennung.
ID:	MQBACF_SUB_ID.
Datentyp:	MQCFBS.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Immer.

### **SubLevel**

Beschreibung:	Die Subskriptionsebene.
ID:	MQIACF_SUB_LEVEL.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Immer.

### **Feedback**

Beschreibung:	Der Grund für das Löschen der Nachricht.
ID:	MQIACF_FEEDBACK.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Wenn die Nachricht gelöscht wurde, weil sie nicht an einen Subskribenten zugestellt wurde, oder die Nachricht nicht zugestellt wurde, weil der Subskribent ausgeschlossen wurde.

Die Veröffentlichungsoperation 'MQOPER\_PUBLISH' enthält Informationen zu einer Nachricht, die an einen bestimmten Subskribenten zugestellt wird. Diese Operation beschreibt die Elemente der einseitigen Nachricht, die sich möglicherweise von der in der zugeordneten Put-Operation beschriebenen Nachricht geändert haben. Ähnlich wie bei einer Put-Operation enthält sie eine Nachrichtengruppe MQGACF\_MES-

SAGE und innerhalb dieser eine MQMD-Gruppe MQGACF\_MQMD. Diese MQMD-Gruppe enthält jedoch die folgenden Felder, die von einem Subskribenten überschrieben werden können: *Format, Priority, Persistence, MsgId, CorrelId, UserIdentifier, AccountingToken, ApplIdentityData*.

Die *SubId* und die *SubLevel* des Subskribenten sind in den Operationsinformationen enthalten. Sie können die *Sub-ID* mit dem PCF-Befehl MQCMD\_INQUIRE\_SUBSCRIBER verwenden, um alle anderen Attribute für einen Subskribenten abzurufen.

Die Operation MQOPER\_DISCARDED\_PUBLISH der Veröffentlichungsfunktion ist analog zu der Operation 'Discard', die verwendet wird, wenn eine Nachricht nicht in Punkt-zu-Punkt-Messaging zugestellt wird. Eine Nachricht wird nicht an einen Subskribenten zugestellt, wenn die Nachricht explizit angefordert wurde, dass sie nicht an eine lokale Destination zugestellt werden soll, und dieser Subskribent gibt ein lokales Ziel an. Eine Nachricht wird auch als nicht zugestellt betrachtet, wenn ein Problem beim Abrufen der Nachricht in die Zielwarteschlange besteht, z. B. weil die Warteschlange voll ist.

Die Informationen in einer Discarded Publish-Operation sind dieselben wie bei einer Publizierungsoperation mit dem Hinzufügen eines *Feedback* -Felds, das die Gründe angibt, warum die Nachricht nicht zugestellt wurde. Dieses Feedback-Feld enthält die Werte MQFB\_\* oder MQRC\_\*, die mit der Operation MQOPER\_DISCARD gemeinsam sind. Der Grund für das Verwerfen einer Veröffentlichung im Gegensatz zu Ausschließen ist mit den Gründen für das Löschen eines Abschlusses identisch.

Die Operation 'Ausgeschlossene Publizierung' MQOPER\_EXCLUDED\_PUBLISH stellt Informationen zu einem Subskribenten bereit, der für die Zustellung der Nachricht in Betracht gezogen wurde, da das Thema, für das der Subskribent subskribiert, mit der zugeordneten Put-Operation übereinstimmt, die Nachricht jedoch nicht an den Subskribenten zugestellt wurde, weil andere Auswahlkriterien nicht mit der Nachricht übereinstimmen, die in das Thema gestellt wird. Wie bei einer Discarded Publish-Operation enthält das Feld *Feedback* Informationen zu der Ursache, warum diese Subskription ausgeschlossen wurde. Anders als bei der Veröffentlichung von Veröffentlichungsoperationen werden jedoch keine nachrichtenbezogenen Informationen bereitgestellt, da für diesen Subskribenten keine Nachricht generiert wurde.

*Bericht 'Put/Put Reply/Put Report' (MQOPER\_PUT/MQOPER\_PUT\_REPLY/MQOPER\_PUT\_REPORT)*

Die zusätzlichen Aktivitätsberichtsdatenparameter, die in der PCF-Gruppe *Operation* für den Operationstyp 'Put/Put Reply/Put Report' (MQOPER\_PUT/MQOPER\_PUT\_REPLY/MQOPER\_PUT\_REPORT) zurückgegeben werden, (eine Nachricht, eine Antwortnachricht oder eine Berichtsnachricht wurde in eine Warteschlange gestellt).

### **QName**

Beschreibung:	Der Name der Warteschlange, die geöffnet wurde.
ID:	MQCA_Q_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation</i> .
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Es wird immer, abgesehen von einer Ausnahme, nicht zurückgegeben, wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

### **ResolvedQName**

Beschreibung:	Der Name, in den die geöffnete Warteschlange aufgelöst wird.
ID:	MQCACF_RESOLVED_Q_NAME.
Datentyp:	MQCFST.

In PCF-Gruppe enthalten: *Operation.*

Maximale Länge: MQ\_Q\_NAME\_LENGTH

Zurückgegeben: Wenn die geöffnete Warteschlange aufgelöst werden konnte. Wird nicht zurückgegeben, wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

### **RemoteQName**

Beschreibung: Der Name der geöffneten Warteschlange, wie sie auf dem fernen Warteschlangenmanager bekannt ist.

ID: MQCA\_REMOTE\_Q\_NAME.

Datentyp: MQCFST.

In PCF-Gruppe enthalten: *Operation.*

Maximale Länge: MQ\_Q\_NAME\_LENGTH

Zurückgegeben: Wenn es sich bei der geöffneten Warteschlange um eine ferne Warteschlange handelt. Wird nicht zurückgegeben, wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

### **RemoteQMgrName**

Beschreibung: Der Name des fernen Warteschlangenmanagers, auf dem die ferne Warteschlange definiert ist.

ID: MQCA\_REMOTE\_Q\_MGR\_NAME.

Datentyp: MQCFST.

In PCF-Gruppe enthalten: *Operation.*

Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH

Zurückgegeben: Wenn es sich bei der geöffneten Warteschlange um eine ferne Warteschlange handelt. Wird nicht zurückgegeben, wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

### **TopicString**

Beschreibung: Die vollständige Themenzeichenfolge, in die die Nachricht gestellt wird.

ID: MQCA\_TOPIC\_STRING.

Datentyp: MQCFST.

In PCF-Gruppe enthalten: *Operation.*

Zurückgegeben: Wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

### **Feedback**

Beschreibung: Der Grund für die Nachricht, die in die Warteschlange für dead-letter gestellt wird.

ID: MQIACF\_FEEDBACK.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Wenn die Nachricht in die Warteschlange für nicht zustellbare Nachrichten gestellt wurde.

#### *Empfangen (MQOPER\_RECEIVE)*

Die zusätzlichen Aktivitätsberichtsdatenparameter, die in der PCF-Gruppe *Operation* für den Operationstyp 'Empfangen (MQOPER\_RECEIVE)' zurückgegeben werden (eine Nachricht wurde auf einem Kanal empfangen).

#### **ChannelName**

Beschreibung:	Der Name des Kanals, auf dem die Nachricht empfangen wurde.
ID:	MQCACH_CHANNEL_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_CHANNEL_NAME_LENGTH
Zurückgegeben:	Immer.

#### **ChannelType**

Beschreibung:	Der Typ des Kanals, auf dem die Nachricht empfangen wurde.
ID:	MQIACH_CHANNEL_TYPE.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Immer.

#### **RemoteQMgrName**

Beschreibung:	Der Name des Warteschlangenmanagers, von dem die Nachricht empfangen wurde.
ID:	MQCA_REMOTE_Q_MGR_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH
Zurückgegeben:	Immer.

#### *Senden (MQOPER\_SEND)*

Die zusätzlichen Nachrichtendatenparameter für Aktivitäten, die in der PCF-Gruppe *Operation* für den Operationstyp 'Senden (MQOPER\_SEND)' zurückgegeben werden (eine Nachricht wurde auf einem Kanal gesendet).

#### **ChannelName**

Beschreibung:	Der Name des Kanals, an den die Nachricht gesendet wurde.
ID:	MQCACH_CHANNEL_NAME.
Datentyp:	MQCFST.

In PCF-Gruppe  
enthalten: *Operation.*

Maximale Länge: MQ\_CHANNEL\_NAME\_LENGTH.

Zurückgegeben: Immer.

### **ChannelType**

Beschreibung: Der Typ des Kanals, an den die Nachricht gesendet wurde.

ID: MQIACH\_CHANNEL\_TYPE.

Datentyp: MQCFIN.

In PCF-Gruppe  
enthalten: *Operation.*

Zurückgegeben: Immer.

### **XmitQName**

Beschreibung: Die Übertragungswarteschlange, aus der die Nachricht abgerufen wurde.

ID: MQCACH\_XMIT\_Q\_NAME.

Datentyp: MQCFST.

In PCF-Gruppe  
enthalten: *Operation.*

Maximale Länge: MQ\_Q\_NAME\_LENGTH.

Zurückgegeben: Immer.

### **RemoteQMgrName**

Beschreibung: Der Name des fernen Warteschlangenmanagers, an den die Nachricht gesendet wurde.

ID: MQCA\_REMOTE\_Q\_MGR\_NAME.

Datentyp: MQCFST.

In PCF-Gruppe  
enthalten: *Operation.*

Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH

Zurückgegeben: Immer.

## **Trace-Route-Nachrichtenreferenz**

Verwenden Sie diese Seite, um eine Übersicht über das Nachrichtenformat der Traceroute zu erhalten. Die Trace-Route-Nachrichtendaten enthalten Parameter, die die Aktivitäten beschreiben, die von der Trace-Route-Nachricht verursacht wurden.

### **Nachrichtenformat für Trace-Route**

Trace-Route-Nachrichten sind IBM MQ-Standardnachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Die Nachrichtendaten enthalten Informationen zu den Aktivitäten, die in einer Trace-Route-Nachricht ausgeführt wurden, da sie über ein WS-Manager-Netz weitergeleitet wurde.

Trace-Route-Nachrichten enthalten die folgenden Informationen:

#### **Ein Nachrichtendeskriptor**

Eine MQMD-Struktur, wobei das Feld *Format* auf MQFMT\_ADMIN oder MQFMT\_EMBEDDED\_PCF gesetzt ist.

## Nachrichtendaten

Besteht aus:

- Ein PCF-Header (MQCFH) und Trace-Route-Nachrichtendaten, wenn *Format* auf MQFMT\_ADMIN gesetzt ist, oder
- Ein eingebetteter PCF-Header (MQEPH), Trace-Route-Nachrichtendaten und zusätzliche benutzerdefinierte Nachrichtendaten, wenn *Format* auf MQFMT\_EMBEDDED\_PCF gesetzt ist.

Wenn Sie mit der IBM MQ-Anwendung zur Routenanzeige eine Trace-Route-Nachricht generieren, wird *Format* auf MQFMT\_ADMIN gesetzt.

Der Inhalt der Trace-Route-Nachrichtendaten wird durch den Parameter *Accumulate* aus der *TraceRoute*-PCF-Gruppe wie folgt bestimmt:

- Wenn *Accumulate* auf MQROUTE\_ACCUMULATE\_NONE gesetzt ist, enthält die Trace-Route-Nachrichtendaten die *TraceRoute*-PCF-Gruppe.
- Wenn *Accumulate* auf MQROUTE\_ACCUMULATE\_IN\_MSG oder MQROUTE\_ACCUMULATE\_AND\_REPLY gesetzt ist, enthält die Trace-Route-Nachrichtendaten die *TraceRoute*-PCF-Gruppe und null oder mehr *Activity*-PCF-Gruppen.

Tabelle 20 auf Seite 130 zeigt die Struktur einer Trace-Route-Nachricht.

Tabelle 20. Nachrichtenformat für Trace-Route

MQMD-Struktur	Eingebetteter PCF-Header MQEPH-Struktur	Trace-Route-Nachrichtendaten
Struktur-ID Strukturversion Berichtsoptionen Nachrichtentyp Verfallszeit Feedback Encoding ID des codierten Zeichensatzes (CCSID). Nachrichtenformat Priority Permanenz Nachrichten-ID Korrelations-ID Rücksetzungszähler Empfangswarteschlange für Antworten Warteschlangenmanager für Antwortwarteschlange Benutzer-ID Abrechnung Anwendungsidentitätsdaten Anwendungstyp Anwendungsname Aufnahmedatum Aufnahmezeit Ursprungsdaten der Anwendung Gruppen-ID Nachrichtenfolgennummer Offset Nachrichtenmarkierungen Ursprüngliche Länge	Struktur-ID Strukturversion Strukturlänge Encoding ID des codierten Zeichensatzes (CCSID). Nachrichtenformat Flags PCF-Header (MQCFH) Strukturtyp Strukturlänge Strukturversion Befehls-ID Nachrichtenfolgennummer Steueroptionen Beendigungscode Ursachencode Parameteranzahl	TraceRoute Detail Aufgezeichnete Aktivitäten Nicht aufgezeichnete Aktivitäten Diskontinuitätszähler Max. Aktivitäten Kumulieren Deliver

### Trace-Route-Nachricht MQMD (Nachrichtendeskriptor)

Verwenden Sie diese Seite, um die Werte anzuzeigen, die in der MQMD-Struktur für eine Trace-Route-Nachricht enthalten sind.

#### StrucId

Beschreibung: Struktur-ID  
 Datentyp: MQCHAR4.  
 Wert: MQMD\_STRUC\_ID.

#### Version

Beschreibung: Strukturversionsnummer.  
 Datentyp: MQLONG.  
 Werte: **MQMD\_VERSION\_1.**

## **Bericht**

Beschreibung: Optionen für Berichtsnachrichten.  
Datentyp: MQLONG.  
Wert: Stellen Sie sie entsprechend den Anforderungen ein. Es folgen allgemeine Berichtsoptionen:

### **MQRO\_DISCARD\_MSG**

Die Nachricht wird bei der Ankunft in eine lokale Warteschlange verworfen.

### **MQRO\_PASS\_DISCARD\_AND\_EXPIRY**

Jede Antwort (Aktivitätsberichte oder Tracing-Route-Antwortnachricht) enthält die Berichtsoption MQRO\_DISCARD\_MSG und die verbleibende Ablaufzeit wurde übergeben. Dadurch wird sichergestellt, dass die Antworten nicht unbegrenzt im Warteschlangenmanagernetzwerk verbleiben.

## **MsgType**

Beschreibung: Typ der Nachricht.  
Datentyp: MQLONG.  
Wert: Wenn der Parameter *Accumulate* in der Gruppe 'TraceRoute' als 'MQROUTE\_ACCUMULATE\_AND\_REPLY' angegeben wird, lautet der Nachrichtentyp 'MQMT\_REQUEST'.

Andernfalls:

### **MQMT\_DATAGRAM.**

## **Verfall**

Beschreibung: Lebensdauer der Nachricht  
Datentyp: MQLONG.  
Wert: Stellen Sie sie entsprechend den Anforderungen ein. Dieser Parameter kann verwendet werden, um sicherzustellen, dass Nachrichten der Traceroute nicht unbegrenzt in einem Warteschlangenmanager-Netz zurückgelassen werden.

## **Feedback**

Beschreibung: Rückmeldungs- oder Ursachencode.  
Datentyp: MQLONG.  
Wert: **MQFB\_NONE.**

## **Encoding**

Beschreibung: Numerische Codierung von Nachrichtendaten.  
Datentyp: MQLONG.  
Wert: Legen Sie die entsprechende Gruppe fest.

## **CodedCharSetId**

Beschreibung: Zeichensatz-ID der Nachrichtendaten.  
Datentyp: MQLONG.  
Wert: Legen Sie die entsprechende Gruppe fest.

### **Format**

Beschreibung:	Name des Formats von Nachrichtendaten
Datentyp:	MQCHAR8.
Wert:	<b>MQFMT_ADMIN</b> Admin-Nachricht. Es folgen keine Benutzerdaten der <i>TraceRoute</i> -PCF-Gruppe. <b>MQFMT_EMBEDDED_PCF</b> Eingebettete PCF-Nachricht Die Benutzerdaten folgen der <i>TraceRoute</i> -PCF-Gruppe.

### **Priorität**

Beschreibung:	Nachrichtenpriorität
Datentyp:	MQLONG.
Wert:	Stellen Sie sie entsprechend den Anforderungen ein.

### **Persistenz**

Beschreibung:	Nachrichtenpersistenz.
Datentyp:	MQLONG.
Wert:	Stellen Sie sie entsprechend den Anforderungen ein.

### **MsgId**

Beschreibung:	Nachrichten-ID.
Datentyp:	MQBYTE24.
Wert:	Stellen Sie sie entsprechend den Anforderungen ein.

### **CorrelId**

Beschreibung:	Korrelations-ID
Datentyp:	MQBYTE24.
Wert:	Stellen Sie sie entsprechend den Anforderungen ein.

### **BackoutCount**

Beschreibung:	Zurücksetzungszähler.
Datentyp:	MQLONG.
Wert:	0.

### **ReplyToQ**

Beschreibung:	Name der Antwortwarteschlange.
Datentyp:	MQCHAR48.
Werte:	Stellen Sie sie entsprechend den Anforderungen ein. Wenn <i>MsgType</i> auf MQMT_REQUEST gesetzt ist oder wenn <i>Report</i> über Berichtsgenerierungsoptionen verfügt, muss dieser Parameter nicht leer sein.

**ReplyToQMgr**

Beschreibung: Antwort-Warteschlangenmanagername  
Datentyp: MQCHAR48.  
Wert: Stellen Sie sie entsprechend den Anforderungen ein.

**UserIdentifier**

Beschreibung: Die Benutzer-ID der Anwendung, die die Nachricht erstellt hat.  
Datentyp: MQCHAR12.  
Wert: Als normal definieren.

**AccountingToken**

Beschreibung: Abrechnungstoken, das es einer Anwendung ermöglicht, die Arbeit zu berechnen, die als Ergebnis der Nachricht ausgeführt wurde.  
Datentyp: MQBYTE32.  
Wert: Als normal definieren.

**ApplIdentityData**

Beschreibung: Identitätsbezogene Anwendungsdaten.  
Datentyp: MQCHAR32.  
Werte: Als normal definieren.

**PutApplType**

Beschreibung: Typ der Anwendung, die die Nachricht eingereicht hat.  
Datentyp: MQLONG.  
Wert: Als normal definieren.

**PutApplName**

Beschreibung: Name der Anwendung, die die Nachricht einreicht.  
Datentyp: MQCHAR28.  
Wert: Als normal definieren.

**PutDate**

Beschreibung: Datum, an dem die Nachricht eingereicht wurde.  
Datentyp: MQCHAR8.  
Wert: Als normal definieren.

**PutTime**

Beschreibung: Zeitpunkt, zu dem die Nachricht eingereicht wurde.  
Datentyp: MQCHAR8.  
Wert: Als normal definieren.

**ApplOriginData**

Beschreibung: Anwendungsdaten zum Ursprung

Datentyp: MQCHAR4.  
Wert: Als normal definieren ..

### ***Trace-Route-Nachricht MQEPH (eingebetteter PCF-Header)***

Verwenden Sie diese Seite, um die Werte anzuzeigen, die in der MQEPH-Struktur für eine Trace-Route-Nachricht enthalten sind.

Die MQEPH-Struktur enthält eine Beschreibung der PCF-Informationen, die die Nachrichtendaten einer Trace-Route-Nachricht begleitet, und die darauf folgenden Anwendungsnachrichtendaten. Eine MQEPH-Struktur wird nur verwendet, wenn zusätzliche Benutzernachrichtendaten der PCF-Gruppe 'TraceRoute' folgen.

Für eine Trace-Route-Nachricht enthält die MQEPH-Struktur die folgenden Werte:

#### ***StrucId***

Beschreibung: Struktur-ID  
Datentyp: MQCHAR4.  
Wert: MQEPH\_STRUC\_ID.

#### ***Version***

Beschreibung: Strukturversionsnummer.  
Datentyp: MQLONG.  
Werte: MQEPH\_VERSION\_1.

#### ***StrucLength***

Beschreibung: Strukturlänge.  
Datentyp: MQLONG.  
Wert: Gesamtlänge der Struktur einschließlich der PCF-Parameterstrukturen, die auf diese Struktur folgen.

#### ***Encoding***

Beschreibung: Numerische Codierung der Nachrichtendaten, die auf die letzte PCF-Parameterstruktur folgen.  
Datentyp: MQLONG.  
Wert: Die Codierung der Nachrichtendaten.

#### ***CodedCharSetId***

Beschreibung: Zeichensatzes-ID der Nachrichtendaten, die auf die letzte PCF-Parameterstruktur folgen.  
Datentyp: MQLONG.  
Wert: Der Zeichensatz der Nachrichtendaten.

#### ***Format***

Beschreibung: Formatname der Nachrichtendaten, die auf die letzte PCF-Parameterstruktur folgen.  
Datentyp: MQCHAR8.  
Wert: Der Formatname der Nachrichtendaten.

## **Flags**

Beschreibung:	Flags, die Attribute der Struktur angeben oder ihre Verarbeitung steuern.
Datentyp:	MQLONG.
Wert:	<b>MQEPH_NONE</b> Es wurden keine Markierungen angegeben. <b>MQEPH_CCSID_EMBEDDED</b> Gibt an, dass der Zeichensatz der Parameter, die Zeichendaten enthalten, einzeln im Feld <i>CodedCharSetId</i> in jeder Struktur angegeben wird.

## **PCFHeader**

Beschreibung:	Header für programmierbares Befehlsformat
Datentyp:	MQCFH.
Wert:	Siehe <a href="#">„Trace-Route-Nachricht MQCFH (PCF-Header)“</a> auf Seite 135.

## **Trace-Route-Nachricht MQCFH (PCF-Header)**

Verwenden Sie diese Seite, um die PCF-Werte anzuzeigen, die in der MQCFH-Struktur für eine Trace-Route-Nachricht enthalten sind.

Für eine Trace-Route-Nachricht enthält die MQCFH-Struktur die folgenden Werte:

### **Type**

Beschreibung:	Strukturtyp, der den Inhalt der Nachricht identifiziert.
Datentyp:	MQLONG.
Wert:	<b>MQCFT_TRACE_ROUTE</b> Nachricht ist eine Trace-Route-Nachricht.

### **StrucLength**

Beschreibung:	Strukturlänge.
Datentyp:	MQLONG.
Wert:	<b>MQCFH_STRUC_LENGTH</b> Länge der MQCFH-Struktur in Byte.

### **Version**

Beschreibung:	Strukturversionsnummer.
Datentyp:	MQLONG.
Werte:	MQCFH_VERSION_3

### **Command**

Beschreibung:	Befehls-ID. Gibt die Kategorie der Nachricht an.
Datentyp:	MQLONG.
Werte:	<b>MQCMD_TRACE_ROUTE</b> Trace-Route-Nachricht.

### **MsgSeqNumber**

Beschreibung:	Nachrichtenfolgennummer. Dies ist die Folgennummer der Nachricht in einer Gruppe zusammengehöriger Nachrichten.
Datentyp:	MQLONG.
Werte:	1.

### **Control**

Beschreibung:	Steuroptionen.
Datentyp:	MQLONG.
Werte:	MQCFC_LAST.

### **CompCode**

Beschreibung:	Beendigungscode.
Datentyp:	MQLONG.
Werte:	MQCC_OK.

### **Reason**

Beschreibung:	Beendigungscode für Ursachencode.
Datentyp:	MQLONG.
Werte:	MQRC_NONE.

### **ParameterCount**

Beschreibung:	Anzahl der Parameterstrukturen. Dies ist die Anzahl der Parameterstrukturen, die auf die MQCFH-Struktur folgen. Eine Gruppenstruktur (MQCFGR) und die darin enthaltenen Parameterstrukturen werden nur als eine einzige Struktur gezählt.
Datentyp:	MQLONG.
Werte:	1 oder größer.

### **Trace-Route-Nachrichtendaten**

Verwenden Sie diese Seite, um die Parameter anzuzeigen, aus denen die *TraceRoute* -PCF-Gruppe Teil der Trace-Route-Nachrichtendaten ist.

Der Inhalt der Trace-Route-Nachrichtendaten hängt vom Parameter *Accumulate* in der *TraceRoute* -PCF-Gruppe ab. Trace-Route-Nachrichtendaten bestehen aus der *TraceRoute* -PCF-Gruppe und null oder mehr *Activity* -PCF-Gruppen. Die *TraceRoute* -PCF-Gruppe ist in diesem Abschnitt ausführlich beschrieben. Weitere Informationen zu der *Activity* -PCF-Gruppe finden Sie in den zugehörigen Informationen.

Trace-Route-Nachrichtendaten enthalten die folgenden Parameter:

#### **TraceRoute**

Beschreibung:	Gruppierte Parameter, die Attribute der Trace-Route-Nachricht angeben. Für eine Trace-Route-Nachricht können einige dieser Parameter geändert werden, um zu steuern, wie die Route verarbeitet wird.
ID:	MQGACF_TRACE_ROUTE.
Datentyp:	MQCFGR.
Enthalten in PCF-Gruppe:	Keine.

Parameter in Gruppe:	<i>Detail</i> <i>RecordedActivities</i> <i>UnrecordedActivities</i> <i>DiscontinuityCount</i> <i>MaxActivities</i> <i>Accumulate</i> <i>Forward</i> <i>Deliver</i>
----------------------	---

### **Detail**

Beschreibung:	Die Detailstufe, die für die Aktivität aufgezeichnet wird.
ID:	MQIACF_ROUTE_DETAIL.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .
Werte:	<p><b>MQROUTE_DETAIL_LOW</b> Aktivitäten, die von einer vom Benutzer geschriebenen Anwendung ausgeführt werden, werden aufgezeichnet.</p> <p><b>MQROUTE_DETAIL_MEDIUM</b> Die in MQROUTE_DETAIL_LOW angegebenen Aktivitäten werden aufgezeichnet. Darüber hinaus werden Aktivitäten, die von MCAs ausgeführt werden, aufgezeichnet.</p> <p><b>MQROUTE_DETAIL_HIGH</b> Die in 'MQROUTE_DETAIL_LOW' und 'MQROUTE_DETAIL_MEDIUM' angegebenen Aktivitäten werden aufgezeichnet. MCAs zeichnen keine weiteren Aktivitätsinformationen auf dieser Detaillierungsebene auf. Diese Option ist nur für benutzerdefinierte Anwendungen verfügbar, die weitere Aktivitätsinformationen aufzeichnen sollen.</p>

### **RecordedActivities**

Beschreibung:	Die Anzahl der Aktivitäten, die von der Trace-Route-Nachricht verursacht wurden, wo Informationen aufgezeichnet wurden.
ID:	MQIACF_RECORDED_ACTIVITIES.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .

### **UnrecordedActivities**

Beschreibung:	Die Anzahl der Aktivitäten, die von der Trace-Route-Nachricht verursacht wurden, wo keine Informationen aufgezeichnet wurden.
ID:	MQIACF_UNRECORDED_ACTIVITIES.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .

### ***DiscontinuityCount***

Beschreibung:	Die Häufigkeit, mit der eine Trace-Route-Nachricht von einem WS-Manager empfangen wurde, der keine Trace-Route-Messaging unterstützt.
ID:	MQIACF_DISCONTINUITY_COUNT.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .

### ***MaxActivities***

Beschreibung:	Die maximale Anzahl an Aktivitäten, an der die Trace-Route-Nachricht beteiligt sein kann, bevor sie nicht mehr verarbeitet wird.
ID:	MQIACF_MAX_ACTIVITIES.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .
Wert:	<b>Eine positive Ganzzahl</b> Die maximale Anzahl der Aktivitäten. <b>MQROUTE_UNLIMITED_ACTIVITIES</b> Eine unbegrenzte Anzahl an Aktivitäten.

### ***Accumulate***

Beschreibung:	Gibt an, ob Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden und ob eine Antwortnachricht, die die aufgelaufenen Aktivitätsinformationen enthält, generiert wird, bevor die Trace-Route-Nachricht gelöscht wird oder in eine Nicht-Übertragungswarteschlange gestellt wird.
ID:	MQIACF_ROUTE_ACCUMULATION.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .
Wert:	<b>MQROUTE_ACCUMULATE_NONE</b> Die Aktivitätsinformationen werden in den Nachrichtendaten der Trace-Route-Nachricht nicht kumuliert. <b>MQROUTE_ACCUMULATE_IN_MSG</b> Die Aktivitätsinformationen werden in den Nachrichtendaten der Trace-Route-Nachricht akkumuliert. <b>MQROUTE_ACCUMULATE_AND_REPLY</b> Die Aktivitätsinformationen werden in den Nachrichtendaten der Trace-Route-Nachricht kumuliert, und es wird eine Antwortnachricht für die Trace-Route generiert.

### ***Forward***

Beschreibung:	Gibt die Warteschlangenmanager an, an die die Trace-Route-Nachricht weitergeleitet werden kann. Bei der Entscheidung, ob eine Nachricht an einen fernen Warteschlangenmanager weitergeleitet werden soll, verwenden Warteschlangenmanager den Algorithmus, der in <a href="#">Weiterleiten</a> beschrieben ist.
ID:	MQIACF_ROUTE_FORWARDING.

Datentyp: MQCFIN.

Enthalten in PCF-Gruppe: *TraceRoute*.

Wert: **MQROUTE\_FORWARD\_IF\_SUPPORTED**

Die Trace-Route-Nachricht wird nur an WS-Manager weitergeleitet, die den Wert des Parameters *Deliver* aus der Gruppe *TraceRoute* berücksichtigen.

**MQROUTE\_FORWARD\_ALL**

Die Trace-Route-Nachricht wird an jeden WS-Manager weitergeleitet, unabhängig davon, ob der Wert des Parameters *Deliver* berücksichtigt wird.

### ***Deliver***

Beschreibung: Gibt die Aktion an, die ausgeführt werden soll, wenn die Trace-Route-Nachricht erfolgreich an der Zielwarteschlange ankommt.

ID: MQIACF\_ROUTE\_DELIVERY.

Datentyp: MQCFIN.

Enthalten in PCF-Gruppe: *TraceRoute*.

Wert: **MQROUTE\_DELIVER\_YES**

Bei der Ankunft wird die Trace-Route-Nachricht in die Zielwarteschlange gestellt. Jede Anwendung, die eine zerstörerische Verbindung in der Zielwarteschlange ausführt, kann die Trace-Route-Nachricht empfangen.

**MQROUTE\_DELIVER\_NO**

Bei der Ankunft wird die Trace-Route-Nachricht gelöscht.

## **Trace-Route-Antwortnachrichtenreferenz**

Verwenden Sie diese Seite, um eine Übersicht über das Antwortnachrichtenformat für Trace-Routen zu erhalten. Die Trace-Route-Antwortnachrichtendaten sind ein Duplikat der Trace-Route-Nachrichtendaten aus der Trace-Route-Nachricht, für die sie generiert wurde.

### ***Antwortnachrichtenformat für Trace-Route***

Trace-Route-Antwortnachrichten sind IBM MQ-Standardnachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Die Nachrichtendaten enthalten Informationen zu den Aktivitäten, die in einer Trace-Route-Nachricht ausgeführt wurden, da sie über ein WS-Manager-Netz weitergeleitet wurde.

Tracing-Route-Antwortnachrichten enthalten die folgenden Informationen:

#### **Ein Nachrichtendeskriptor**

Eine MQMD-Struktur

#### **Nachrichtendaten**

Ein PCF-Header (MQCFH) und Trace-Route-Antwortnachrichtendaten

Die Trace-Route-Antwortnachrichtendaten bestehen aus einer oder mehreren *Activity* -PCF-Gruppen.

Wenn eine Trace-Route-Nachricht ihre Zielwarteschlange erreicht, kann eine Trace-Route-Antwortnachricht generiert werden, die eine Kopie der Aktivitätsinformationen aus der Trace-Route-Nachricht enthält. Die Antwortnachricht für die Trace-Route wird an eine Empfangswarteschlange für Antworten oder an eine Systemwarteschlange zugestellt.

Tabelle 21 auf Seite 140 zeigt die Struktur einer Antwortnachricht für die Trace-Route, einschließlich der Parameter, die nur unter bestimmten Bedingungen zurückgegeben werden.

Tabelle 21. Antwortnachrichtenformat für Trace-Route

MQMD-Struktur	PCF-Header MQCFH-Struktur	Trace-Route-Antwortnachrichtendaten
Struktur-ID Strukturversion Berichtsoptionen Nachrichtentyp Verfallszeit Feedback Encoding ID des codierten Zeichensatzes (CCSID). Nachrichtenformat Priority Permanenz Nachrichten-ID Korrelations-ID Rücksetzungszähler Empfangswarteschlange für Antworten Warteschlangenmanager für Antwortwarteschlange Benutzer-ID Abrechnung Anwendungsidentitätsdaten Anwendungstyp Anwendungsname Aufnahmezeit Ursprungsdaten der Anwendung Gruppen-ID Nachrichtenfolgennummer Offset Nachrichtenmarkierungen Ursprüngliche Länge	PCF-Header (MQCFH) Strukturtyp Strukturgröße Strukturversion Befehls-ID Nachrichtenfolgennummer Steueroptionen Beendigungscode Ursachencode Parameteranzahl	Aktivität Anwendungsname der Aktivität Aktivitätsanwendungstyp Aktivitätsbeschreibung Operation Operationstyp Operationstermin Operationszeit Nachricht Nachrichtenlänge MQMD EmbeddedMQMD Name des Warteschlangenmanagers Name der Gruppe mit gemeinsamer Warteschlange Warteschlangenname <sup>1 2 3</sup> Aufgelöster Warteschlangenname <sup>1 3</sup> Name der fernen Warteschlange <sup>3</sup> Ferner Warteschlangenmanager-Name <sup>2 3 4 5</sup> Feedback <sup>2</sup> Kanalname <sup>4 5</sup> Kanaltyp <sup>4 5</sup> Name der Übertragungswarteschlange <sup>5</sup> TraceRoute Detail Aufgezeichnete Aktivitäten Nicht aufgezeichnete Aktivitäten Diskontinuitätszähler Max. Aktivitäten Kumulieren Deliver

**Anmerkung:**

1. Wird für Get- und Browse-Operationen zurückgegeben.
2. Wird für Discard-Operationen zurückgegeben.
3. Zurückgegeben für Put-, Put-Reply- und Put-Report-Operationen.
4. Zurückgegeben für Empfangsoperationen.
5. Wird für Sendeoperationen zurückgegeben.

**Trace-Route-Antwortnachricht MQMD (Nachrichtendeskriptor)**

Verwenden Sie diese Seite, um die Werte anzuzeigen, die in der MQMD-Struktur für eine Trace-Route-Antwortnachricht enthalten sind.

Für eine Antwortnachricht auf Trace-Route enthält die MQMD-Struktur die Parameter, die im Nachrichtendeskriptor für Aktivitätsberichte beschrieben sind. Einige der Parameterwerte in einem Antwortnach-

richtendeskriptor einer Trace-Route unterscheiden sich von denen in einem Nachrichtendeskriptor für Aktivitätsberichte wie folgt:

### ***MsgType***

Beschreibung: Typ der Nachricht.  
Datentyp: MQLONG.  
Wert: **MQMT\_REPLY**

### ***Feedback***

Beschreibung: Rückmeldungs- oder Ursachencode.  
Datentyp: MQLONG.  
Wert: **MQFB\_NONE**

### ***Encoding***

Beschreibung: Numerische Codierung von Nachrichtendaten.  
Datentyp: MQLONG.  
Wert: Kopiert aus Trace-Route-Nachrichtendeskriptor.

### ***CodedCharSetId***

Beschreibung: Zeichensatz-ID der Nachrichtendaten.  
Datentyp: MQLONG.  
Wert: Kopiert aus Trace-Route-Nachrichtendeskriptor.

### ***Format***

Beschreibung: Name des Formats von Nachrichtendaten  
Datentyp: MQCHAR8.  
Wert: **MQFMT\_ADMIN**  
Admin-Nachricht.

### ***Trace-Route-Antwortnachricht MQCFH (PCF-Header)***

Verwenden Sie diese Seite, um die PCF-Werte anzuzeigen, die in der MQCFH-Struktur für eine Trace-Route-Antwortnachricht enthalten sind.

Der PCF-Header (MQCFH) für eine Trace-Route-Antwortnachricht ist die gleiche wie bei einer Trace-Route-Nachricht.

### ***Trace-Route-Antwortnachrichtendaten***

Die Trace-Route-Antwortnachrichtendaten sind ein Duplikat der Trace-Route-Nachrichtendaten aus der Trace-Route-Nachricht, für die sie generiert wurde.

Die Antwortnachrichtendaten der Traceroute enthalten mindestens eine *Activity* -Gruppe. Die Parameter werden in „[Aktivitätsbericht-Nachrichtendaten](#)“ auf Seite 111 beschrieben.

## **Abrechnungs- und Statistiknachrichten**

Warteschlangenmanager generieren Abrechnungs- und Statistiknachrichten, um Informationen zu den von IBM MQ-Anwendungen ausgeführten MQI-Operationen oder zu den Aktivitäten in einem IBM MQ-System aufzuzeichnen.

## Abrechnungsnachrichten

Mit Abrechnungsnachrichten werden Informationen zu den MQI-Operationen aufgezeichnet, die von IBM MQ-Anwendungen ausgeführt werden (siehe [„Abrechnungsnachrichten“](#) auf Seite 142).

## Statistiknachrichten

Mit Statistiknachrichten werden Informationen zu den Aktivitäten aufgezeichnet, die in einem IBM MQ-System auftreten (siehe [„Statistiknachrichten“](#) auf Seite 146). Einige in Statistiknachrichten aufgezeichnete Aktivitäten beziehen sich auf interne Warteschlangenmanageroperationen.

 Die hier beschriebenen Abrechnungsnachrichten und Statistiknachrichten sind nicht unter IBM MQ for z/OS verfügbar, aber über SMF (System Management Facility, Systemverwaltungsfunktion) ist eine ähnliche Funktion verfügbar.

Abrechnungs- und Statistiknachrichten werden an eine von zwei Systemwarteschlangen zugestellt. Benutzeranwendungen können die Nachrichten aus diesen Systemwarteschlangen abrufen und die aufgezeichneten Informationen zu verschiedenen Zwecken verwenden:

- Konto für Anwendungsressourcenverwendung.
- Die Anwendungsaktivität erfassen.
- Kapazitätsplanung.
- Erkennen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Unterstützung bei der Bestimmung der Ursachen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Verbessern Sie die Effizienz Ihres Warteschlangenmanagernetzes.
- Machen Sie sich mit der Ausführung Ihres WS-Manager-Netzes vertraut.
- Stellen Sie sicher, dass Ihr Warteschlangenmanagernetzwerk ordnungsgemäß ausgeführt wird.

## Zugehörige Konzepte

[„Systemverwaltungsfunktion verwenden“](#) auf Seite 335

Sie können SMF verwenden, um Statistikdaten und Abrechnungsdaten zu erfassen. Für die Verwendung von SMF müssen bestimmte Parameter in z/OS und IBM MQ festgelegt werden.

## Abrechnungsnachrichten

Abrechnungsnachrichten zeichnen Informationen zu den MQI-Operationen auf, die von IBM MQ-Anwendungen ausgeführt werden. Eine Abrechnungsnachricht ist eine PCF-Nachricht, die eine Reihe von PCF-Strukturen enthält.

Wenn eine Anwendung die Verbindung zu einem WS-Manager trennt, wird eine Abrechnungsnachricht generiert und an die Systemabrechnungswarteschlange (SYSTEM.ADMIN.ACCOUNTING.QUEUE) übergeben. Für IBM MQ-Anwendungen mit langer Laufzeit werden temporäre Abrechnungsnachrichten folgendermaßen generiert:

- Wenn die Zeit seit dem Zeitpunkt der Verbindungseinbindung überschritten wurde, überschreitet das konfigurierte Intervall.
- Wenn die Zeit seit der letzten Zwischenabrechnungsnachricht das konfigurierte Intervall überschreitet.

Abrechnungsnachrichten befinden sich in den folgenden Kategorien:

### MQI-Abrechnungsnachrichten

MQI-Abrechnungsnachrichten enthalten Informationen, die sich auf die Anzahl der MQI-Aufrufe beziehen, die über eine Verbindung zu einem Warteschlangenmanager hergestellt wurden.

### Nachrichten zur Warteschlangenabrechnung

Warteschlangenabrechnungsnachrichten enthalten Informationen, die sich auf die Anzahl der MQI-Aufrufe beziehen, die unter Verwendung von Verbindungen zu einem Warteschlangenmanager (gruppiert nach Warteschlange) hergestellt wurden.

Jede Warteschlangenabrechnungsnachricht kann bis zu 100 Datensätze enthalten, wobei jeder Datensatz sich auf eine Aktivität bezieht, die von der Anwendung in Bezug auf eine bestimmte Warteschlange ausgeführt wird.

Abrechnungsnachrichten werden nur für lokale Warteschlangen aufgezeichnet. Wenn eine Anwendung einen MQI-Aufruf für eine Aliaswarteschlange aufruft, werden die Abrechnungsdaten für die Basiswarteschlange erfasst und die Abrechnungsdaten für eine ferne Warteschlange werden für die Übertragungswarteschlange aufgezeichnet.

### **Zugehörige Verweise**

„MQI-Abrechnungsnachrichtendaten“ auf Seite 161

Verwenden Sie diese Seite, um die Struktur einer MQI-Abrechnungsnachricht anzuzeigen.

„Nachrichtendaten der Warteschlangenabrechnung“ auf Seite 173

Auf dieser Seite können Sie die Struktur einer Warteschlangenabrechnungsnachricht anzeigen.

### **Abrechnungsnachrichtenformat**

Abrechnungsnachrichten umfassen eine Gruppe von PCF-Feldern, die aus einem Nachrichtendeskriptor und Nachrichtendaten bestehen.

#### **Nachrichtendeskriptor**

- Eine Abrechnungsnachricht MQMD (Nachrichtendeskriptor)

#### **Abrechnungsnachrichtendaten**

- Eine Abrechnungsnachricht MQCFH (PCF-Header)
- Accounting-Nachrichtendaten, die immer zurückgegeben werden
- Abrechnungs-Nachrichtendaten, die zurückgegeben werden, falls

Die Abrechnungsnachricht MQCFH (PCF-Header) enthält Informationen zu der Anwendung und das Intervall, in dem die Abrechnungsdaten aufgezeichnet wurden.

Abrechnungsnachrichtendaten enthalten PCF-Parameter, in denen die Abrechnungsdaten gespeichert werden. Der Inhalt der Abrechnungsnachrichten hängt wie folgt von der Nachrichtenkategorie ab:

#### **MQI-Abrechnungsnachricht**

Die MQI-Abrechnungsnachrichtendaten bestehen aus einer Reihe von PCF-Parametern, aber keine PCF-Gruppen.

#### **Warteschlangenabrechnungsnachricht**

Die Daten der Warteschlangenabrechnungsnachricht bestehen aus einer Reihe von PCF-Parametern und im Bereich von 1 bis 100 *QAccountingData* -PCF-Gruppen.

Es gibt eine *QAccountingData* -PCF-Gruppe für jede Warteschlange, für die Abrechnungsdaten erfasst wurden. Wenn eine Anwendung auf mehr als 100 Warteschlangen zugreift, werden mehrere Abrechnungsnachrichten generiert. Jede Nachricht hat die *SeqNumber* im MQCFH (PCF-Header) entsprechend aktualisiert, und die letzte Nachricht in der Sequenz hat den Parameter *Control* in der MQCFH-Datei als MQCFC\_LAST angegeben.

### **Erfassung von Abrechnungsdaten**

Verwenden Sie die Attribute der Warteschlange und des Warteschlangenmanagers, um die Erfassung von Abrechnungsdaten zu steuern. Sie können MQCONNX-Optionen auch verwenden, um die Erfassung auf Verbindungsebene zu steuern.

#### *Erfassung von MQI-Abrechnungsdaten steuern*

Verwenden Sie das WS-Manager-Attribut ACCTMQI, um die Erfassung von MQI-Abrechnungsdaten zu steuern.

Wenn Sie den Wert dieses Attributs ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter ACCTMQI an. Abrechnungsnachrichten werden nur für Verbindungen generiert, die nach der Aktivierung der Abrechnung beginnen. Der Parameter **ACCTMQI** kann die folgenden Werte haben:

**ON**

Die MQI-Abrechnungsdaten werden für jede Verbindung zum WS-Manager erfasst.

**OFF**

Die MQI-Abrechnungsdaten werden nicht erfasst. Dies ist der Standardwert.

Wenn Sie beispielsweise die Erfassung von MQI-Abrechnungsdaten aktivieren möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QMGR ACCTMQI(ON)
```

*Informationen zur Warteschlangenabrechnung*

Verwenden Sie das Warteschlangenattribut ACCTQ und das WS-Manager-Attribut ACCTQ, um die Erfassung von Warteschlangenabrechnungsdaten zu steuern.

Wenn Sie den Wert des Warteschlangenattributs ändern möchten, verwenden Sie den MQSC-Befehl ALTER QLOCAL und geben Sie den Parameter ACCTQ an. Abrechnungsnachrichten werden nur für Verbindungen generiert, die nach der Aktivierung der Abrechnung beginnen. Beachten Sie, dass Änderungen an diesem Wert nur für Verbindungen zum WS-Manager wirksam sind, die nach der Änderung des Attributs auftreten.

Das Warteschlangenattribut ACCTQ kann die folgenden Werte haben:

**ON**

Die Warteschlangenabrechnungsdaten für diese Warteschlange werden für jede Verbindung zu dem Warteschlangenmanager erfasst, der die Warteschlange öffnet.

**OFF**

Die Warteschlangenabrechnungsdaten für diese Warteschlange werden nicht erfasst.

**QMGR**

Die Erfassung der Warteschlangenabrechnungsdaten für diese Warteschlange wird entsprechend dem Wert des Warteschlangenmanagerattributs ACCTQ gesteuert. Dies ist der Standardwert.

Wenn Sie den Wert des Warteschlangenmanagerattributs ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter ACCTQ an. Das WS-Manager-Attribut ACCTQ kann die folgenden Werte haben:

**ON**

Die Warteschlangenabrechnungsdaten werden für Warteschlangen erfasst, die das Warteschlangenattribut ACCTQ als QMGR festgelegt haben.

**OFF**

Es werden keine Warteschlangenabrechnungsdaten für Warteschlangen erfasst, für die das Warteschlangenattribut ACCTQ als QMGR festgelegt ist. Dies ist der Standardwert.

**KEINE**

Die Erfassung von Warteschlangenabrechnungsdaten ist für alle Warteschlangen inaktiviert, unabhängig vom Warteschlangenattribut ACCTQ.

Wenn das WS-Managerattribut ACCTQ auf NONE gesetzt ist, ist die Erfassung von Warteschlangenabrechnungsinformationen für alle Warteschlangen inaktiviert, unabhängig vom Warteschlangenattribut ACCTQ.

Wenn Sie beispielsweise die Erfassung von Abrechnungsdaten für die Warteschlange Q1 aktivieren möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QLOCAL(Q1) ACCTQ(ON)
```

Verwenden Sie den folgenden MQSC-Befehl, um die Erfassung von Abrechnungsdaten für alle Warteschlangen zu aktivieren, in denen das Warteschlangenattribut ACCTQ als QMGR angegeben ist:

```
ALTER QMGR ACCTQ(ON)
```

### *MQCONN-Optionen für die Steuerung der Erfassung von Abrechnungsdaten*

Verwenden Sie den Parameter **ConnectOpts** im Aufruf MQCONN, um die Erfassung von MQI- und Warteschlangenabrechnungsdaten auf Verbindungsebene zu ändern, indem die effektiven Werte der WS-Manager-Attribute ACCTMQI und ACCTQ überschrieben werden.

Der Parameter **ConnectOpts** kann die folgenden Werte haben:

#### **MQCNO\_ACCOUNTING\_MQI\_ENABLED**

Wenn der Wert des WS-Managerattributs ACCTMQI als OFF angegeben ist, wird die MQI-Abrechnung für diese Verbindung aktiviert. Dies entspricht dem WS-Managerattribut ACCTMQI, das als ON angegeben wird.

Wenn der Wert des WS-Managerattributs ACCTMQI nicht als OFF angegeben ist, hat dieses Attribut keine Auswirkungen.

#### **MQCNO\_ACCOUNTING\_MQI\_DISABLED**

Wenn der Wert des WS-Managerattributs ACCTMQI als ON angegeben wird, ist die MQI-Abrechnung für diese Verbindung inaktiviert. Dies entspricht dem WS-Managerattribut ACCTMQI, das als OFF angegeben wird.

Wenn der Wert des WS-Managerattributs ACCTMQI nicht als ON angegeben ist, hat dieses Attribut keine Auswirkungen.

#### **MQCNO\_ACCOUNTING\_Q\_ENABLED**

Wenn der Wert des Warteschlangenmanagerattributs ACCTQ als OFF angegeben ist, wird die Warteschlangenabrechnung für diese Verbindung aktiviert. Alle Warteschlangen mit ACCTQ, die als QMGR angegeben sind, werden für die Warteschlangenabrechnung aktiviert. Dies entspricht dem WS-Managerattribut ACCTQ, das als ON angegeben wird.

Wenn der Wert des WS-Managerattributs ACCTQ nicht als OFF angegeben ist, hat dieses Attribut keine Auswirkungen.

#### **MQCNO\_ACCOUNTING\_Q\_DISABLED**

Wenn der Wert des Warteschlangenmanagerattributs ACCTQ als ON angegeben ist, ist die Warteschlangenabrechnung für diese Verbindung inaktiviert. Dies entspricht dem WS-Managerattribut ACCTQ, das als OFF angegeben wird.

Wenn der Wert des Warteschlangenmanagerattributs ACCTQ nicht als ON angegeben ist, hat dieses Attribut keine Auswirkungen.

Diese Überschreibungen sind standardmäßig inaktiviert. Wenn Sie sie aktivieren möchten, setzen Sie das WS-Managerattribut ACCTCONO auf ENABLED. Verwenden Sie den folgenden MQSC-Befehl, um Abrechnungsüberschreibungen für einzelne Verbindungen zu aktivieren:

```
ALTER QMGR ACCTCONO(ENABLED)
```

### *Generierung von Accounting-Nachrichten bei Verbindung zu Anwendungen*

Abrechnungsnachrichten werden generiert, wenn eine Anwendung die Verbindung zum WS-Manager trennt. Temporäre Abrechnungsnachrichten werden auch für lang aktive IBM MQ-Anwendungen geschrieben.

Abrechnungsnachrichten werden auf eine der folgenden Arten generiert, wenn eine Anwendung die Verbindung trennt:

- Die Anwendung gibt einen MQDISC-Aufruf aus.
- Der WS-Manager erkennt, dass die Anwendung beendet wurde.

Temporäre Abrechnungsnachrichten werden für lange aktive IBM MQ-Anwendungen geschrieben, wenn das Intervall seit dem Zeitpunkt der Verbindungsherstellung oder seit der letzten temporären Abrechnungsnachricht, die geschrieben wurde, das konfigurierte Intervall überschreitet. Das WS-Manager-Attribut (ACCTINT) gibt die Zeit in Sekunden an, nach der temporäre Abrechnungsnachrichten automatisch geschrieben werden können. Abrechnungsnachrichten werden nur generiert, wenn die Anwendung mit dem Warteschlangenmanager interagiert, so dass Anwendungen, die lange Zeiträume ohne Ausführung

von MQI-Anforderungen mit dem Warteschlangenmanager verbunden bleiben, keine Abrechnungsnachrichten generieren, bis die Ausführung der ersten MQI-Anforderung nach Abschluss des Abrechnungsintervalls erfolgt ist.

Das Standardabrechnungsintervall beträgt 1800 Sekunden (30 Minuten). Wenn Sie z. B. das Accounting-Intervall auf 900 Sekunden (15 Minuten) ändern möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QMGR ACCTINT(900)
```

## Statistiknachrichten

Informationen zu den Datensätzen für Statistiknachrichten über die Aktivitäten in einem IBM MQ-System. Bei einer Statistiknachricht handelt es sich um eine PCF-Nachricht, die eine Reihe von PCF-Strukturen enthält.

Statistiknachrichten werden in den konfigurierten Intervallen an die Systemwarteschlange (SYSTEM.ADMIN.STATISTICS.QUEUE) zugestellt, wenn eine Aktivität vorhanden ist.

Statistiknachrichten befinden sich in den folgenden Kategorien:

### MQI-Statistiknachrichten

MQI-Statistiknachrichten enthalten Informationen, die sich auf die Anzahl der MQI-Aufrufe während eines konfigurierten Intervalls beziehen. Die Informationen können z. B. die Anzahl der MQI-Aufrufe enthalten, die von einem Warteschlangenmanager ausgegeben werden.

### Warteschlangenstatistiknachrichten

Warteschlangenstatistiknachrichten enthalten Informationen, die sich auf die Aktivität einer Warteschlange während eines konfigurierten Intervalls beziehen. Zu den Informationen gehören die Anzahl der Nachrichten, die in die Warteschlange gestellt und aus der Warteschlange abgerufen wurden, sowie die Gesamtzahl der Byte, die von einer Warteschlange verarbeitet wurden.

Jede Warteschlangenstatistiknachricht kann bis zu 100 Datensätze enthalten, wobei jeder Datensatz sich auf die Aktivität pro Warteschlange bezieht, für die Statistikdaten erfasst wurden.

Statistiknachrichten werden nur für lokale Warteschlangen aufgezeichnet. Wenn eine Anwendung einen MQI-Aufruf für eine Aliaswarteschlange aufruft, werden die Statistikdaten für die Basiswarteschlange aufgezeichnet, und für eine ferne Warteschlange werden die Statistikdaten für die Übertragungswarteschlange aufgezeichnet.

### Kanalstatistiknachrichten

Kanalstatistiknachrichten enthalten Informationen, die sich auf die Aktivität eines Kanals während eines konfigurierten Intervalls beziehen. Beispielsweise kann es sich bei den Informationen um die Anzahl der vom Kanal übertragenen Nachrichten oder um die Anzahl der Byte handeln, die vom Kanal übertragen wurden.

Jede Kanalstatistiknachricht enthält bis zu 100 Datensätze, wobei jeder Datensatz sich auf die Aktivität pro Kanal bezieht, für die Statistikdaten erfasst wurden.

### Zugehörige Verweise

[„Steuerung der MQI-Statistikerfassung“ auf Seite 147](#)

Verwenden Sie das WS-Manager-Attribut STATMQI, um die Erfassung von MQI-Statistikinformationen zu steuern.

[„Erfassung von Warteschlangenstatistikdaten steuern“ auf Seite 148](#)

Verwenden Sie das Warteschlangenattribut STATQ und das Warteschlangenmanagerattribut STATQ, um die Erfassung von Warteschlangenstatistikdaten zu steuern.

[„Erfassung von Kanalstatistikdaten steuern“ auf Seite 149](#)

Verwenden Sie das Kanalattribut STATCHL, um die Erfassung von Kanalstatistikdaten zu steuern. Sie können auch WS-Manager-Attribute festlegen, um die Informationssammlung zu steuern.

## **Format der Statistknachrichten**

Statistknachrichten umfassen eine Gruppe von PCF-Feldern, die aus einem Nachrichtendeskriptor und Nachrichtendaten bestehen.

### **Nachrichtendeskriptor**

- Eine Statistknachricht MQMD (Nachrichtendeskriptor)

### **Abrechnungsnachrichtendaten**

- Eine Statistknachricht MQCFH (PCF-Header)
- Statistknachrichtendaten, die immer zurückgegeben werden
- Statistknachrichtendaten, die zurückgegeben werden, falls verfügbar

Die Statistknachricht MQCFH (PCF-Header) enthält Informationen zu dem Intervall, für das die Statistknachrichtendaten aufgezeichnet wurden.

Statistische Nachrichtendaten enthalten PCF-Parameter, die die Statistkinformationen speichern. Der Inhalt der Statistknachrichten hängt wie folgt von der Nachrichtenkatgorie ab:

### **MQI-Statistknachricht**

Die MQI-Statistik-Nachrichtendaten bestehen aus einer Reihe von PCF-Parametern, aber keine PCF-Gruppen.

### **Warteschlangenstatistknachricht**

Die Nachrichtendaten der Warteschlangenstatistik bestehen aus einer Reihe von PCF-Parametern und im Bereich von 1 bis 100 *QStatisticsData* -PCF-Gruppen.

Es gibt eine *QStatisticsData* -PCF-Gruppe für jede Warteschlange, die im Intervall aktiv war. Wenn mehr als 100 Warteschlangen in dem Intervall aktiv waren, werden mehrere Statistknachrichten generiert. Jede Nachricht hat die *SeqNumber* im MQCFH (PCF-Header) entsprechend aktualisiert, und die letzte Nachricht in der Sequenz hat den Parameter *Control* in der MQCFH-Datei als MQCFC\_LAST angegeben.

### **Kanalstatistknachricht**

Die Nachrichtendaten der Kanalstatistik bestehen aus einer Reihe von PCF-Parametern und im Bereich von 1 bis 100 *ChlStatisticsData* -PCF-Gruppen.

Es gibt eine *ChlStatisticsData* -PCF-Gruppe für jeden Kanal, der im Intervall aktiv war. Wenn mehr als 100 Kanäle in dem Intervall aktiv waren, werden mehrere Statistknachrichten generiert. Jede Nachricht hat die *SeqNumber* im MQCFH (PCF-Header) entsprechend aktualisiert, und die letzte Nachricht in der Sequenz hat den Parameter *Control* in der MQCFH-Datei als MQCFC\_LAST angegeben.

## **Erfassung statistischer Daten**

Verwenden Sie die Attribute der Warteschlange, des Warteschlangenmanagers und des Kanals, um die Erfassung statistischer Daten zu steuern.

### *Steuerung der MQI-Statistikerfassung*

Verwenden Sie das WS-Manager-Attribut STATMQI, um die Erfassung von MQI-Statistkinformationen zu steuern.

Wenn Sie den Wert dieses Attributs ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter **STATMQI** an. Statistknachrichten werden nur für Warteschlangen generiert, die nach der Aktivierung der Statistikerfassung geöffnet werden. Der Parameter **STATMQI** kann die folgenden Werte haben:

#### **ON**

MQI-Statistkinformationen werden für jede Verbindung zum Warteschlangenmanager erfasst.

#### **OFF**

Die MQI-Statistkinformationen werden nicht erfasst. Dies ist der Standardwert.

Wenn Sie beispielsweise die Erfassung von MQI-Statistikdaten aktivieren möchten, verwenden Sie den folgenden MQSC-Befehl:

*Erfassung von Warteschlangenstatistikdaten steuern*

Verwenden Sie das Warteschlangenattribut STATQ und das Warteschlangenmanagerattribut STATQ, um die Erfassung von Warteschlangenstatistikdaten zu steuern.

Sie können die Erfassung von Warteschlangenstatistikdaten für einzelne Warteschlangen oder für mehrere Warteschlangen aktivieren oder inaktivieren. Wenn Sie einzelne Warteschlangen steuern möchten, legen Sie das Warteschlangenattribut STATQ fest. Sie aktivieren oder inaktivieren die Erfassung von Warteschlangenstatistikdaten auf WS-Managerebene unter Verwendung des Warteschlangenmanagerattributs STATQ. Für alle Warteschlangen, für die das Warteschlangenattribut STATQ mit dem Wert QMGR angegeben wurde, wird die Erfassung von Warteschlangenstatistikdaten auf WS-Managerebene gesteuert.

Warteschlangenstatistikdaten werden nur für Operationen mit IBM MQ MQI-Objektkennungen erhöht, die nach dem Aktivieren der Statistikerfassung geöffnet wurden.

Warteschlangenstatistiknachrichten werden nur für Warteschlangen generiert, für die die Statistikdaten im vorherigen Zeitraum erfasst wurden.

Dieselbe Warteschlange kann über mehrere Operationen zum Einlegen von Objekten und zum Abrufen von Operationen über mehrere Objektkennungen verfügen. Einige Objektkennungen wurden möglicherweise geöffnet, bevor die Statistikerfassung aktiviert wurde, andere jedoch später geöffnet wurden. Aus diesem Grund ist es möglich, dass die Warteschlangenstatistik die Aktivität einiger Operationen zum Einlegen und Abrufen von Operationen erfasst und nicht alle.

Um sicherzustellen, dass die Warteschlangenstatistik die Aktivität aller Anwendungen erfasst, müssen Sie neue Objektkennungen in der Warteschlange oder in den Warteschlangen schließen und erneut öffnen, die Sie überwachen. Der beste Weg, um dies zu erreichen, ist es, alle Anwendungen nach der Aktivierung der Statistikerfassung zu beenden und erneut zu starten.

Wenn Sie den Wert des Warteschlangenattributs STATQ ändern möchten, verwenden Sie den MQSC-Befehl ALTER QLOCAL und geben Sie den Parameter STATQ an. Das Warteschlangenattribut STATQ kann die folgenden Werte haben:

**ON**

Warteschlangenstatistikdaten werden für jede Verbindung zu dem Warteschlangenmanager erfasst, der die Warteschlange öffnet.

**OFF**

Warteschlangenstatistikdaten für diese Warteschlange werden nicht erfasst.

**QMGR**

Die Erfassung der Warteschlangenstatistikdaten für diese Warteschlange wird entsprechend dem Wert des Warteschlangenmanagerattributs STATQ gesteuert. Dies ist der Standardwert.

Wenn Sie den Wert des Warteschlangenmanagerattributs STATQ ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter STATQ an. Das WS-Manager-Attribut STATQ kann die folgenden Werte haben:

**ON**

Warteschlangenstatistikdaten werden für Warteschlangen erfasst, für die das Warteschlangenattribut STATQ als QMGR festgelegt ist.

**OFF**

Warteschlangenstatistikdaten werden nicht für Warteschlangen erfasst, für die das Warteschlangenattribut STATQ als QMGR festgelegt ist. Dies ist der Standardwert.

**KEINE**

Die Erfassung von Warteschlangenstatistikdaten ist für alle Warteschlangen inaktiviert, unabhängig vom Warteschlangenattribut STATQ.

Wenn das Warteschlangenmanagerattribut STATQ auf NONE gesetzt ist, ist die Erfassung von Warteschlangenstatistikdaten für alle Warteschlangen inaktiviert, unabhängig vom Warteschlangenattribut STATQ.

Wenn Sie beispielsweise die Erfassung statistischer Daten für die Warteschlange Q1 aktivieren möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QLOCAL(Q1) STATQ(ON)
```

Verwenden Sie den folgenden MQSC-Befehl, um die Erfassung statistischer Daten für alle Warteschlangen zu aktivieren, die das Warteschlangenattribut STATQ als QMGR angeben:

```
ALTER QMGR STATQ(ON)
```

#### *Erfassung von Kanalstatistikdaten steuern*

Verwenden Sie das Kanalattribut STATCHL, um die Erfassung von Kanalstatistikdaten zu steuern. Sie können auch WS-Manager-Attribute festlegen, um die Informationssammlung zu steuern.

Sie können die Erfassung von Kanalstatistikdaten für einzelne Kanäle oder für mehrere Kanäle aktivieren oder inaktivieren. Um einzelne Kanäle zu steuern, müssen Sie das Kanalattribut STATCHL festlegen, um die Erfassung der Kanalstatistikdaten zu aktivieren oder zu inaktivieren. Um viele Kanäle gemeinsam zu steuern, aktivieren oder inaktivieren Sie die Erfassung von Kanalstatistikdaten auf WS-Managerebene mit dem WS-Managerattribut STATCHL. Für alle Kanäle, die über das Kanalattribut STATCHL verfügen, das mit dem Wert QMGR angegeben wurde, wird die Erfassung der Kanalstatistikdaten auf WS-Managerebene gesteuert.

Automatisch definierte Clustersenderkanäle sind keine IBM MQ-Objekte und weisen daher keine Attribute auf dieselbe Weise wie Kanalobjekte auf. Wenn Sie automatisch definierte Clustersenderkanäle steuern möchten, verwenden Sie das WS-Managerattribut STATACLS. Dieses Attribut bestimmt, ob automatisch definierte Clustersenderkanäle innerhalb eines Warteschlangenmanagers für die Erfassung von Kanalstatistikdaten aktiviert oder inaktiviert sind.

Sie können die Erfassung von Kanalstatistikdaten auf eine der drei folgenden Überwachungsstufen festlegen: niedrig, mittel oder hoch. Sie können die Überwachungsstufe auf Objektebene oder auf WS-Managerebene festlegen. Die Auswahl der zu verwendenden Ebene hängt von Ihrem System ab. Die Erfassung statistischer Daten erfordert möglicherweise einige Anweisungen, die relativ teuer sind, um die Auswirkungen der Erfassung von Kanalstatistikdaten zu reduzieren, die Optionen für mittlere und niedrige Überwachungsdaten eine Stichprobe der Daten in regelmäßigen Intervallen zu messen und nicht die ganze Zeit zu erfassen. In [Tabelle 22 auf Seite 149](#) sind die Stufen zusammengefasst, die mit der Erfassung von Kanalstatistikdaten verfügbar sind:

<b>Stufe</b>	<b>Beschreibung</b>	<b>Verwendung</b>
Niedrig	Messen Sie in regelmäßigen Abständen eine kleine Stichprobe der Daten.	Für Objekte, die ein hohes Volumen an Nachrichten verarbeiten.
Mittel	Messen Sie in regelmäßigen Abständen ein Beispiel für die Daten.	Für die meisten Objekte.
Hoch	Messen Sie alle Daten in regelmäßigen Intervallen.	Für Objekte, die nur wenige Nachrichten pro Sekunde verarbeiten, auf denen die meisten aktuellen Informationen wichtig sind.

Wenn Sie den Wert des Kanalattributs STATCHL ändern möchten, verwenden Sie den MQSC-Befehl ALTER CHANNEL und geben Sie den Parameter STATCHL an.

Wenn Sie den Wert des Warteschlangenmanagerattributs STATCHL ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter STATCHL an.

Wenn Sie den Wert des Warteschlangenmanagerattributs STATACLS ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter STATACLS an.

Das Attribut channel, STATCHL, kann die folgenden Werte haben:

**LOW**

Die Kanalstatistikdaten werden mit einem niedrigen Detaillierungsgrad erfasst.

**MITTEL**

Kanalstatistikdaten werden mit einer mittleren Detaillierungsebene erfasst.

**HIGH**

Kanalstatistikdaten werden mit einem hohen Detaillierungsgrad erfasst.

**OFF**

Kanalstatistikdaten werden für diesen Kanal nicht erfasst.

**QMGR**

Das Kanalattribut wird als QMGR festgelegt. Die Erfassung von Statistikinformationen für diesen Kanal wird durch den Wert des Warteschlangenmanagerattributs STATCHL gesteuert.

Dies ist der Standardwert.

 Auf z/OS -Systemen wird durch die Aktivierung dieses Parameters einfach die Erfassung statistischer Daten aktiviert, unabhängig vom ausgewählten Wert. Die Angabe von LOW, MEDIUM oder HIGH hat keine Auswirkung auf die Ergebnisse. Dieser Parameter muss aktiviert sein, damit Datensätze zur Kanalabrechnung erfasst werden können.

Das WS-Managerattribut STATCHL kann die folgenden Werte haben:

**LOW**

Kanalstatistikdaten werden mit einem niedrigen Detaillierungsgrad für alle Kanäle erfasst, für die das Kanalattribut STATCHL als QMGR festgelegt ist.

**MITTEL**

Kanalstatistikdaten werden mit einer mittleren Detaillierungsebene für alle Kanäle erfasst, für die das Kanalattribut STATCHL als QMGR festgelegt ist.

**HIGH**

Die Kanalstatistikdaten werden mit einem hohen Detaillierungsgrad für alle Kanäle erfasst, für die das Kanalattribut STATCHL als QMGR festgelegt ist.

**OFF**

Die Kanalstatistikdaten werden nicht für alle Kanäle erfasst, für die das Kanalattribut STATCHL als QMGR festgelegt ist.

Dies ist der Standardwert.

**KEINE**

Die Erfassung der Kanalstatistikdaten ist für alle Kanäle inaktiviert, unabhängig vom Kanalattribut STATCHL.

 Auf z/OS -Systemen wird durch die Aktivierung dieses Parameters einfach die Erfassung statistischer Daten aktiviert, unabhängig vom ausgewählten Wert. Die Angabe von LOW, MEDIUM oder HIGH hat keine Auswirkung auf die Ergebnisse. Dieser Parameter muss aktiviert sein, damit Datensätze zur Kanalabrechnung erfasst werden können.

Das WS-Managerattribut STATACLS kann die folgenden Werte haben:

**LOW**

Statistikinformationen werden mit einem niedrigen Detaillierungsgrad für automatisch definierte Clustersenderkanäle erfasst.

## MITTEL

Statistikinformationen werden mit einer mittleren Detaillierungsebene für automatisch definierte Clustersenderkanäle erfasst.

## HIGH

Statistikinformationen werden mit einem hohen Detaillierungsgrad für automatisch definierte Clustersenderkanäle erfasst.

## OFF

Statistikinformationen werden für automatisch definierte Clustersenderkanäle nicht erfasst.

## QMGR

Die Erfassung statistischer Daten für automatisch definierte Clustersenderkanäle wird durch den Wert des WS-Managerattributs STATCHL gesteuert.

Dies ist der Standardwert.

 Auf z/OS -Systemen wird durch die Aktivierung dieses Parameters einfach die Erfassung statistischer Daten aktiviert, unabhängig vom ausgewählten Wert. Die Angabe von LOW, MEDIUM oder HIGH hat keine Auswirkung auf die Ergebnisse. Dieser Parameter muss aktiviert sein, damit Datensätze zur Kanalabrechnung erfasst werden können.

Wenn Sie beispielsweise die Erfassung statistischer Daten mit einer mittleren Detaillierungsebene für den Senderkanal QM1 . T0 . QM2 aktivieren möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER CHANNEL(QM1.T0.QM2) CHLTYPE(SDR) STATCHL(MEDIUM)
```

Verwenden Sie den folgenden MQSC-Befehl, um die Erfassung statistischer Daten auf mittlerer Detaillierungsebene für alle Kanäle zu aktivieren, die das Kanalattribut STATCHL als QMGR angeben:

```
ALTER QMGR STATCHL(MEDIUM)
```

Verwenden Sie den folgenden MQSC-Befehl, um die Erfassung statistischer Daten für alle automatisch definierten Clustersenderkanäle mit mittlerer Detaillierungsebene zu aktivieren:

```
ALTER QMGR STATACLS(MEDIUM)
```

### *Generierung von Statistknachrichten*

Statistknachrichten werden in konfigurierten Intervallen generiert, und wenn ein Warteschlangenmanager kontrolliert heruntergefahren wird.

Das konfigurierte Intervall wird durch das Attribut STATINT des Warteschlangenmanagers gesteuert, das das Intervall (in Sekunden) zwischen der Generierung von Statistknachrichten angibt. Das Standardstatistikintervall beträgt 1800 Sekunden (30 Minuten). Wenn Sie das Statistikintervall ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter **STATINT** an. Wenn Sie zum Beispiel das Statistikintervall auf 900 Sekunden (15 Minuten) ändern möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QMGR STATINT(900)
```

Verwenden Sie den MQSC-Befehl RESET QMGR TYPE(STATISTICS), um die derzeit erfassten Statistikdaten in die Statistikwarteschlange zu schreiben, bevor das Intervall für die Statistikerfassung abgelaufen ist. Wenn Sie diesen Befehl absetzen, werden die erfassten Statistikdaten in die Statistikwarteschlange geschrieben und ein neues Datenerfassungsintervall für die Statistik gestartet.

## Abrechnungs-und Statistikdaten anzeigen

Um die in Abrechnungs-und Statistknachrichten aufgezeichneten Informationen zu verwenden, führen Sie eine Anwendung wie das Beispielprogramm **amqsmon** aus, um die aufgezeichneten Informationen in ein geeignetes Format zu transformieren.

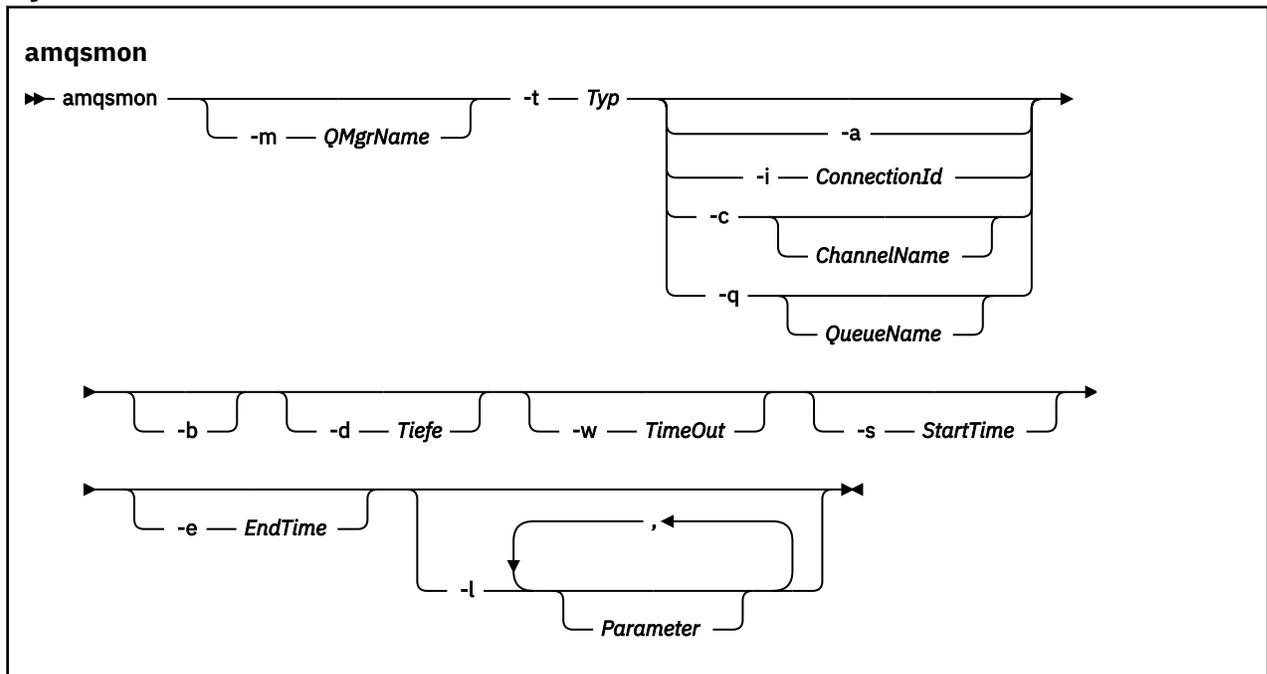
Abrechnungs- und Statistiknachrichten werden in die Systemabrechnungs- und Statistikwarteschlangen geschrieben. **amqsmmon** ist ein mit IBM MQ bereitgestelltes Beispielprogramm, in dem Nachrichten aus Abrechnungs- und Statistikwarteschlangen verarbeitet und die Informationen in einem lesbaren Format in der Anzeige angezeigt werden.

Da **amqsmmon** ein Musterprogramm ist, können Sie den bereitgestellten Quellcode als Schablone verwenden, um Ihre eigene Anwendung zu schreiben, um Abrechnungs- oder Statistiknachrichten zu verarbeiten, oder den Quellcode von **amqsmmon** ändern, um Ihre eigenen speziellen Anforderungen zu erfüllen.

### **amqsmmon (Formatierte Überwachungsdaten anzeigen)**

Verwenden Sie das Beispielprogramm **amqsmmon**, um die in Abrechnungs- und Statistiknachrichten enthaltenen Informationen in einem lesbaren Format anzuzeigen. Das Programm **amqsmmon** liest Abrechnungsnachrichten aus der Abrechnungswarteschlange SYSTEM.ADMIN.ACCOUNTING.QUEUE. und liest Statistiknachrichten aus der Statistikwarteschlange, SYSTEM.ADMIN.STATISTICS.QUEUE.

### **Syntax**



### **Erforderliche Parameter**

#### **-t Type**

Der Typ der Nachrichten, die verarbeitet werden sollen. Geben Sie *Type* als einen der folgenden Werte an:

#### **Buchhaltung**

Abrechnungssätze werden verarbeitet. Nachrichten werden aus der Systemwarteschlange SYSTEM.ADMIN.ACCOUNTING.QUEUE gelesen.

#### **statistics**

Statistikdatensätze werden verarbeitet. Nachrichten werden aus der Systemwarteschlange SYSTEM.ADMIN.STATISTICS.QUEUE gelesen.

### **Optionale Parameter**

#### **-m QMGrName**

Der Name des Warteschlangenmanagers, aus dem Abrechnungs- oder Statistiknachrichten verarbeitet werden sollen.

Wenn Sie diesen Parameter nicht angeben, wird der Standardwarteschlangenmanager verwendet.

**-a**

Prozessnachrichten, die nur MQI-Datensätze enthalten.

Nur MQI-Datensätze anzeigen. Nachrichten, die keine MQI-Datensätze enthalten, bleiben immer in der Warteschlange, aus der sie gelesen wurden.

**-q QueueName**

*QueueName* ist ein optionaler Parameter.

Wenn *QueueName* nicht angegeben wird:

Zeigt nur die Warteschlangenabrechnungs- und Warteschlangenstatistikdatensätze an

Wenn *QueueName* angegeben wird:

Zeigt die Warteschlangenabrechnungs- und Warteschlangenstatistikdatensätze für die Warteschlange an, die nur von *QueueName* angegeben wurde

Wenn *-b* nicht angegeben wird, werden die Abrechnungs- und Statistiknachrichten, aus denen die Datensätze stammen, gelöscht. Da Abrechnungs- und Statistiknachrichten auch Datensätze aus anderen Warteschlangen enthalten können, wenn *-b* nicht angegeben ist, können nicht gesehener Datensätze gelöscht werden.

**-c ChannelName**

*ChannelName* ist ein optionaler Parameter.

Wenn *ChannelName* nicht angegeben wird:

Zeigt nur Kanalstatistikdatensätze an.

Wenn *ChannelName* angegeben wird:

Zeigt Kanalstatistikdatensätze für den Kanal an, der nur von *ChannelName* angegeben wird.

Wenn *-b* nicht angegeben wird, werden die Statistiknachrichten, aus denen die Datensätze stammen, gelöscht. Da Statistiknachrichten auch Datensätze aus anderen Kanälen enthalten können, wenn *-b* nicht angegeben wird, können nicht gesehener Datensätze gelöscht werden.

Dieser Parameter ist nur verfügbar, wenn die Statistiknachrichten angezeigt werden (*-t statistics*).

**-i ConnectionId**

Zeigt Datensätze an, die sich auf die Verbindungskennung beziehen, die nur von *ConnectionId* angegeben wurde.

Dieser Parameter ist nur verfügbar, wenn nur Abrechnungsnachrichten angezeigt werden (*-t accounting*).

Wenn *-b* nicht angegeben wird, werden die Statistiknachrichten, aus denen die Datensätze stammen, gelöscht. Da Statistiknachrichten auch Datensätze aus anderen Kanälen enthalten können, wenn *-b* nicht angegeben wird, können nicht gesehener Datensätze gelöscht werden.

**-b**

Nachrichten durchsuchen.

Nachrichten werden nicht destruktiv abgerufen.

**-d Depth**

Die maximale Anzahl der Nachrichten, die verarbeitet werden können.

Wenn Sie diesen Parameter nicht angeben, kann eine unbegrenzte Anzahl von Nachrichten verarbeitet werden.

**-w TimeOut**

Die maximale Zeit in Sekunden, die auf die Verfügbarkeit einer Nachricht gewartet wird.

Wenn Sie diesen Parameter nicht angeben, wird `amqsmon` beendet, wenn keine weiteren Nachrichten mehr verarbeitet werden.

#### **-s StartTime**

Prozessnachrichten werden nur nach dem angegebenen `StartTime` gesendet.

`StartTime` wird im Format `yyyy-mm-dd hh.mm.ss` angegeben. Wenn ein Datum ohne Uhrzeit angegeben wird, wird die Zeit standardmäßig `00.00.00` an dem angegebenen Datum angegeben. Die Zeiten sind in GMT.

Informationen zur Angabe dieses Parameters finden Sie in [Anmerkung 1](#).

#### **-e EndTime**

Verarbeiten von Nachrichten, die vor dem angegebenen `EndTime` gestellt werden.

Der `EndTime` wird im Format `yyyy-mm-dd hh.mm.ss` angegeben. Wenn ein Datum ohne Uhrzeit angegeben wird, wird die Zeit standardmäßig `23.59.59` an dem angegebenen Datum angegeben. Die Zeiten sind in GMT.

Informationen zur Angabe dieses Parameters finden Sie in [Anmerkung 1](#).

#### **-l Parameter**

Nur die ausgewählten Felder aus den verarbeiteten Datensätzen anzeigen. `Parameter` ist eine durch Kommas getrennte Liste mit ganzzahligen Werten, wobei jede Ganzzahlwertzuordnung der numerischen Konstanten eines Felds zugeordnet wird. Weitere Informationen finden Sie im Abschnitt [amqsmon-Beispiel 5](#).

Wenn Sie diesen Parameter nicht angeben, werden alle verfügbaren Felder angezeigt.

#### **Anmerkung:**

1. Wenn Sie `-s StartTime` oder `-e EndTime` nicht angeben, werden die Nachrichten, die verarbeitet werden können, nicht durch die Eingeschränkte Zeit eingeschränkt.

### ***amqsmon (Formatierte Überwachungsdaten anzeigen), Beispiele***

Verwenden Sie diese Seite, um Beispiele für die Ausführung des Beispielprogramms 'amqsmon' (Formatierte Überwachungsdaten anzeigen) anzuzeigen.

1. Siehe „[Nachrichtendaten der Warteschlangenstatistik](#)“ auf Seite 195 für eine Erklärung der Attribute.

Mit dem folgenden Befehl werden alle MQI-Statistikenachrichten vom WS-Manager `saturn.queue.manager` angezeigt:

```
amqsmon -m saturn.queue.manager -t statistics -a
```

Die Ausgabe dieses Befehls folgt:

```
RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 73, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqCount: [4, 2102, 0, 0, 0, 46, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [26, 1]
PutFailCount: 0
Put1Count: [40, 0]
```

```

Put1FailCount: 0
PutBytes: [57064, 12320]
GetCount: [18, 1]
GetBytes: [52, 12320]
GetFailCount: 2254
BrowseCount: [18, 60]
BrowseBytes: [23784, 30760]
BrowseFailCount: 9
CommitCount: 0
CommitFailCount: 0
BackCount: 0
ExpiredMsgCount: 0
PurgeCount: 0

```

2. Mit dem folgenden Befehl werden alle Warteschlangenstatistiknachrichten für die Warteschlange LOCALQ auf dem Warteschlangenmanager saturn.queue.manager angezeigt:

```
amqsmon -m saturn.queue.manager -t statistics -q LOCALQ
```

Die Ausgabe dieses Befehls folgt:

```

RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics:
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
  QueueDefinitionType: Local
  QMinDepth: 0
  QMaxDepth: 18
  AverageQueueTime: [29827281, 0]
  PutCount: [26, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [88, 0]
  GetCount: [18, 0]
  GetBytes: [52, 0]
  GetFailCount: 0
  BrowseCount: [0, 0]
  BrowseBytes: [0, 0]
  BrowseFailCount: 1
  NonQueuedMsgCount: 0
  ExpiredMsgCount: 0
  PurgedMsgCount: 0

```

3. Mit dem folgenden Befehl werden alle Statistiknachrichten angezeigt, die seit 15:30 Uhr am 30. April 2005 vom Warteschlangenmanager saturn.queue.manager erfasst wurden.

```
amqsmon -m saturn.queue.manager -t statistics -s "2005-04-30 15.30.00"
```

Die Ausgabe dieses Befehls folgt:

```

RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
...

```

```

RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics: 0
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
  ...
QueueStatistics: 1
  QueueName: 'SAMPLEQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
  ...

```

4. Siehe „[Nachrichtendaten der Warteschlangenabrechnung](#)“ auf Seite 173 für eine Erklärung der Attribute.

Mit dem folgenden Befehl werden alle Abrechnungsnachrichten angezeigt, die am 30. April 2005 vom WS-Manager saturn.queue.manager erfasst wurden:

```
amqsmon -m saturn.queue.manager -t accounting -s "2005-04-30" -e "2005-04-30"
```

Die Ausgabe dieses Befehls folgt:

```

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.29'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.09.30'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'
SeqNumber: 0
ApplicationName: 'amqsput'
ApplicationPid: 8572
ApplicationTid: 1
UserId: 'admin'
ConnDate: '2005-03-16'
ConnTime: '15.09.29'
DiscDate: '2005-03-16'
DiscTime: '15.09.30'
DiscType: Normal
OpenCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [1, 0]
PutFailCount: 0
PutBytes: [4, 0]
GetCount: [0, 0]
GetFailCount: 0
GetBytes: [0, 0]
BrowseCount: [0, 0]
BrowseFailCount: 0
BrowseBytes: [0, 0]
CommitCount: 0
CommitFailCount: 0
BackCount: 0
InqCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-03-16'
IntervalStartTime: '15.16.22'
IntervalEndDate: '2005-03-16'
IntervalEndTime: '15.16.22'
CommandLevel: 600

```

```
ConnectionId: x'414d514354524556312020202020208d0b3742010c0020'  
SeqNumber: 0  
ApplicationName: 'runmqsc'  
ApplicationPid: 8615  
ApplicationTid: 1  
...
```

5. Der folgende Befehl durchsucht die Abrechnungswarteschlange und zeigt den Anwendungsnamen und die Verbindungskennung jeder Anwendung an, für die MQI-Abrechnungsdaten verfügbar sind:

```
amqsmon -m saturn.queue.manager -t accounting -b -l 7006,3024
```

Die Ausgabe dieses Befehls folgt:

```
MonitoringType: QueueAccounting  
ConnectionId: x'414d5143514d39303520202020202020f1855e01e80322'  
ApplicationName: 'WebSphere MQ\bin\amqsput.exe'  
QueueAccounting: 0  
  
MonitoringType: QueueAccounting  
ConnectionId: x'414d5143514d39303520202020202020f1855e01ea0322'  
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'  
QueueAccounting: 0  
QueueAccounting: 1  
QueueAccounting: 2  
QueueAccounting: 3  
QueueAccounting: 4  
QueueAccounting: 5  
QueueAccounting: 6  
QueueAccounting: 7  
QueueAccounting: 8  
QueueAccounting: 9  
  
MonitoringType: QueueAccounting  
ConnectionId: x'414d5143514d39303520202020202020f1855e01e90322'  
ApplicationName: 's\IBM\MQ_4\bin64\amqsput.exe'  
QueueAccounting: 0  
  
MonitoringType: QueueAccounting  
ConnectionId: x'414d5143514d39303520202020202020f1855e01ef0322'  
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'  
QueueAccounting: 0  
QueueAccounting: 1  
QueueAccounting: 2  
QueueAccounting: 3  
QueueAccounting: 4  
QueueAccounting: 5  
QueueAccounting: 6  
QueueAccounting: 7  
QueueAccounting: 8  
QueueAccounting: 9  
  
MonitoringType: QueueAccounting  
ConnectionId: x'414d5143514d39303520202020202020f1855e01e60322'  
ApplicationName: 's\IBM\MQ_4\bin64\runmqsc.exe'  
QueueAccounting: 0  
  
5 Records Processed
```

Weitere Informationen zur Lokalisierung der in diesem Abschnitt verwendeten Variablen finden Sie unter „[Zuordnung für ApplicationName \(3024\) und ConnectionId \(7006\) suchen](#)“ auf Seite 157 .

Details dazu, wie Sie Variablen lokalisieren, wenn Sie *Statistik* für den Parameter **Type** im Befehl **Amqsmon** ausgewählt haben, finden Sie unter „[Zuordnung für AvgTimeOnQ \(703\) und QmaxDepth \(739\) suchen](#)“ auf Seite 158 .

*Zuordnung für **ApplicationName** (3024) und **ConnectionId** (7006) suchen*

In diesem Abschnitt wird erläutert, wie die im Beispiel verwendeten Variablen für die Überwachung von Abrechnungsdaten gefunden werden.

Zusammenfassung:

- 3024 bedeutet MQCACF\_APPL\_NAME, d. h. **ApplicationName**

- 7006 bedeutet MQBACF\_CONNECTION\_ID, d. h. **ConnectionId**

Zur Ermittlung der Zuordnung müssen Sie zwei Prozeduren ausführen:

1. Eine Erläuterung der Attribute „ApplicationName“ auf Seite 174 und „ConnectionId“ auf Seite 174 bei Auswahl von *accounting* für den Parameter **Type** im Befehl **amqsmom** finden Sie unter „Nachrichtendaten der Warteschlangenabrechnung“ auf Seite 173 .

Suchen Sie in jedem Fall nach dem Attribut **Identifizier**.

Der **Identifizier** für **ApplicationName** ist MQCACF\_APPL\_NAME und für **ConnectionId** ist MQBACF\_CONNECTION\_ID

2. Suchen Sie nach den IDs, die Sie in Schritt „1“ auf Seite 158 gefunden haben.

Wechseln Sie zum Abschnitt Konstanten und blättern Sie abwärts, bis Sie die Liste MQCACF\_\* (Befehlsformat Zeichenparametertypen) finden. Suchen Sie MQCACF\_APPL\_NAME und der Wert 3024 wird angezeigt.

Suchen Sie in ähnlicher Weise nach der Liste MQBACF\_\* (Befehlsformat Byteparametertypen). Suchen Sie MQBACF\_CONNECTION\_ID und der Wert 7006 wird angezeigt.

*Zuordnung für AvgTimeOnQ (703) und QmaxDepth (739) suchen*

In diesem Abschnitt wird erläutert, wie die im Beispiel verwendeten Variablen für Überwachungsstatistiken gefunden werden.

Zusammenfassung:

- 703 bedeutet MQIAMO64\_AVG\_Q\_TIME, d. h. **AvgTimeOnQ**
- 739 bedeutet MQIAMO\_Q\_MAX\_DEPTH, d. h. **QMaxDepth**

Zur Ermittlung der Zuordnung müssen Sie zwei Prozeduren ausführen:

1. Eine Erläuterung der Attribute „AvgTimeOnQ“ auf Seite 199 und „QMaxDepth“ auf Seite 198 bei Auswahl von *statistics* für den Parameter **Type** im Befehl **amqsmom** finden Sie unter „Nachrichtendaten der Warteschlangenstatistik“ auf Seite 195 .

Suchen Sie in jedem Fall nach dem Attribut **Identifizier**.

**Identifizier** für **AvgTimeOnQ** ist MQIAMO64\_AVG\_Q\_TIME und für **QMaxDepth** ist MQIAMO\_Q\_MAX\_DEPTH.

2. Suchen Sie nach den IDs, die Sie in Schritt „1“ auf Seite 158 gefunden haben.

Wechseln Sie zum Abschnitt Konstanten und blättern Sie abwärts, bis Sie die Liste MQIAMO\_\* (Befehlsformat Ganzzahlüberwachung Parametertypen) finden. Suchen Sie MQIAMO\_Q\_MAX\_DEPTH und Sie sehen den Wert 739.

Suchen Sie in ähnlicher Weise nach der Liste MQIAMO64\_\* (Befehlsformat 64-Bit-Parametertypen für Ganzzahlüberwachung) . Suchen Sie MQIAMO64\_AVG\_Q\_TIME und der Wert 703 wird angezeigt.

## Nachrichtenreferenz für Abrechnung und Statistik

Verwenden Sie diese Seite, um eine Übersicht über das Format der Abrechnungs- und Statistikenachrichten und die in diesen Nachrichten zurückgegebenen Informationen zu erhalten.

Abrechnungs- und Statistikenachrichten sind IBM MQ-Standardnachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Die Nachrichtendaten enthalten Informationen zu den MQI-Operationen, die von IBM MQ-Anwendungen ausgeführt werden, oder Informationen zu den Aktivitäten in einem IBM MQ-System.

### Nachrichtendeskriptor

- Eine MQMD-Struktur

### Nachrichtendaten

- Ein PCF-Header (MQCFH)

- Accounting-oder Statistiknachrichtendaten, die immer zurückgegeben werden
- Abrechnungs-oder Statistiknachrichtendaten, die zurückgegeben werden, falls verfügbar

### Nachrichtenformat für Abrechnung und Statistik

Verwenden Sie diese Seite als Beispiel für die Struktur einer MQI-Abrechnungsnachricht.

Tabelle 23. MQI-Abrechnungsnachrichtenstruktur		
MQMD-Struktur	MQCFH-Struktur des Abrechnungsnachrichtenheaders	MQI-Abrechnungsnachrichtendaten <sup>1</sup>
Struktur-ID Strukturversion Berichtsoptionen Nachrichtentyp Verfallszeit Rückmeldungscode Encoding ID des codierten Zeichensatzes (CCSID). Nachrichtenformat Nachrichtenpriorität Permanenz Nachrichten-ID Korrelations-ID Rücksetzungszähler Empfangswarteschlange für Antworten Warteschlangenmanager für Antwortwarteschlange Benutzer-ID Abrechnung Anwendungsidentitätsdaten Anwendungstyp Anwendungsname Aufnahmedatum Aufnahmezeit Ursprungsdaten der Anwendung Gruppen-ID Nachrichtenfolgennummer Offset Nachrichtenmarkierungen Ursprüngliche Länge	Strukturtyp Strukturlänge Strukturversion Befehls-ID Nachrichtenfolgennummer Steueroptionen Beendigungscode Ursachencode Parameteranzahl	Warteschlangenmanager Startdatum des Intervalls Intervallstartzeit Enddatum des Intervalls Endzeit des Intervalls Befehlsebene Verbindungskennung Folgennummer Anwendungsname Anwendungsprozess-ID Anwendungsthread-ID Benutzer-ID Verbindungsdatum Verbindungszeit Verbindungsname Kanalname Datum der Verbindung trennen Zeitpunkt der Verbindungsverbindung Typ der Verbindung trennen Offene Zählung Offene Fehleranzahl Schließzahl Schließungsanzahl schließen Put-Anzahl Put-Fehleranzahl Put1-Zähler Anzahl der Put1-Fehler Lege Byte Abrutzähler Fehleranzahl abrufen Bytes abrufen Durchsuchungsanzahl Durchsuchungsanzahl durchsuchen Bytes durchsuchen Festschreibungszähler Commit-Fehleranzahl Rücksetzungszähler Inquire-Zähler Inquire-Anzahl der Fehler Anzahl festlegen Anzahl der Fehler festlegen

Tabelle 23. MQI-Abrechnungsnachrichtenstruktur (Forts.)

MQMD-Struktur	MQCFH-Struktur des Abrechnungsnachrichtenheaders	MQI-Abrechnungsnachrichtendaten <sup>1</sup>
<p><b>Anmerkung:</b></p> <p>1. Die angezeigten Parameter sind die Parameter, die für eine MQI-Abrechnungsnachricht zurückgegeben werden. Die tatsächlichen Abrechnungs- oder Statistikenachrichtendaten hängen von der Nachrichtenkatgorie ab.</p>		

### **Abrechnungs- und Statistikenachricht MQMD (Nachrichtendeskriptor)**

Verwenden Sie diese Seite, um die Unterschiede zwischen dem Nachrichtendeskriptor von Abrechnungs- und Statistikenachrichten und dem Nachrichtendeskriptor von Ereignisnachrichten zu verstehen.

Die Parameter und Werte im Nachrichtendeskriptor der Abrechnungs- und Statistikenachricht sind dieselben wie im Nachrichtendeskriptor von Ereignisnachrichten, mit der folgenden Ausnahme:

#### **Format**

Beschreibung:	Formatname der Nachrichtendaten.
Datentyp:	MQCHAR8.
Wert:	<b>MQFMT_ADMIN</b> Admin-Nachricht.

Einige der Parameter, die im Nachrichtendeskriptor der Abrechnungs- und Statistikenachricht enthalten sind, enthalten feste Daten, die von dem Warteschlangenmanager bereitgestellt werden, der die Nachricht generiert hat.

Der MQMD gibt außerdem den Namen des Warteschlangenmanagers (abgeschnitten auf 28 Zeichen) an, der die Nachricht eingibt, sowie das Datum und die Uhrzeit, an dem bzw. zu der die Nachricht in die Abrechnungs- oder Statistikwarteschlange gestellt wurde.

### **Nachrichtendaten in Abrechnungs- und Statistikenachrichten**

Die Nachrichtendaten in Abrechnungs- und Statistikenachrichten basieren auf dem programmierbaren Befehlsformat (PCF), das in PCF-Befehlsabfragen und -antworten verwendet wird. Die Nachrichtendaten in Abrechnungs- und Statistikenachrichten bestehen aus einem PCF-Header (MQCFH) und einem Accounting- oder Statistikbericht.

### **Abrechnungs- und Statistikenachricht MQCFH (PCF-Header)**

Der Nachrichtenheader von Abrechnungs- und Statistikenachrichten ist eine MQCFH-Struktur. Die Parameter und Werte in der Nachrichtenheader der Abrechnungs- und Statistikenachricht sind dieselben wie im Nachrichtenheader von Ereignisnachrichten mit den folgenden Ausnahmen:

#### **Command**

Beschreibung:	Befehls-ID. Gibt die Nachrichtenkatgorie für die Abrechnung oder die Statistikenachricht an.
Datentyp:	MQLONG.

Werte:	<b>MQCMD_ACCOUNTING_MQI</b> MQI-Abrechnungsnachricht.
	<b>MQCMD_ACCOUNTING_Q</b> Warteschlangenabrechnungsnachricht.
	<b>MQCMD_STATISTICS_MQI</b> MQI-Statistikenachricht.
	<b>MQCMD_STATISTICS_Q</b> Warteschlangenstatistikenachricht.
	<b>MQCMD_STATISTICS_CHANNEL</b> Kanalstatistikenachricht.

### **Version**

Beschreibung:	Strukturversionsnummer.
Datentyp:	MQLONG.
Wert:	<b>MQCFH_VERSION_3</b> Version-3 für Abrechnungs- und Statistikenachrichten.

### **Abrechnungs- und Statistikenachrichtendaten**

Der Inhalt der Nachrichtendaten für Abrechnungs- und Statistikenachrichtendaten ist abhängig von der Kategorie der Abrechnungs- oder Statistikenachricht wie folgt:

#### **MQI-Abrechnungsnachricht**

Die MQI-Abrechnungsnachrichtendaten bestehen aus einer Reihe von PCF-Parametern, aber keine PCF-Gruppen.

#### **Warteschlangenabrechnungsnachricht**

Die Daten der Warteschlangenabrechnungsnachricht bestehen aus einer Reihe von PCF-Parametern und im Bereich von 1 bis 100 *QAccountingData* -PCF-Gruppen.

#### **MQI-Statistikenachricht**

Die MQI-Statistik-Nachrichtendaten bestehen aus einer Reihe von PCF-Parametern, aber keine PCF-Gruppen.

#### **Warteschlangenstatistikenachricht**

Die Nachrichtendaten der Warteschlangenstatistik bestehen aus einer Reihe von PCF-Parametern und im Bereich von 1 bis 100 *QStatisticsData* -PCF-Gruppen.

#### **Kanalstatistikenachricht**

Die Nachrichtendaten der Kanalstatistik bestehen aus einer Reihe von PCF-Parametern und im Bereich von 1 bis 100 *ChlStatisticsData* -PCF-Gruppen.

### **MQI-Abrechnungsnachrichtendaten**

Verwenden Sie diese Seite, um die Struktur einer MQI-Abrechnungsnachricht anzuzeigen.

Nachrichtenname:	MQI-Abrechnungsnachricht.
Plattformen:	Alle außer IBM MQ for z/OS.
Systemwarteschlange:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

### **QueueManager**

Beschreibung:	Der Name des Warteschlangenmanagers.
ID:	MQCA_Q_MGR_NAME
Datentyp:	MQCFST

Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH  
Zurückgegeben: Immer

### ***IntervalStartDate***

Beschreibung: Das Datum des Startzeitbeginns des Überwachungszeitraums  
ID: MQCAMO\_START\_DATE  
Datentyp: MQCFST  
Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Immer

### ***IntervalStartTime***

Beschreibung: Der Zeitpunkt des Startzeitpunktes des Überwachungszeitraums.  
ID: MQCAMO\_START\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Immer

### ***IntervalEndDate***

Beschreibung: Das Datum des Endes des Überwachungszeitraums  
ID: MQCAMO\_END\_DATE  
Datentyp: MQCFST  
Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Immer

### ***IntervalEndTime***

Beschreibung: Die Zeit bis zum Ende des Überwachungszeitraums  
ID: MQCAMO\_END\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Immer

### ***CommandLevel***

Beschreibung: Die Befehlsebene des Warteschlangenmanagers  
ID: MQIA\_COMMAND\_LEVEL  
Datentyp: MQCFIN  
Zurückgegeben: Immer

### ***ConnectionId***

Beschreibung: Die Verbindungs-ID für die IBM MQ-Verbindung  
ID: MQBACF\_CONNECTION\_ID  
Datentyp: MQCFBS

Maximale Länge: MQ\_CONNECTION\_ID\_LENGTH  
Zurückgegeben: Immer

### ***SeqNumber***

Beschreibung: Die Folgenummer. Dieser Wert wird für jeden nachfolgenden Datensatz für lange laufende Verbindungen erhöht.  
ID: MQIACF\_SEQUENCE\_NUMBER  
Datentyp: MQCFIN  
Zurückgegeben: Immer

### ***ApplicationName***

Beschreibung: Der Name der Anwendung. Der Inhalt dieses Felds entspricht dem Inhalt des Feldes *PutApplName* im Nachrichtendeskriptor.  
ID: MQCACF\_APPL\_NAME  
Datentyp: MQCFST  
Maximale Länge: MQ\_APPL\_NAME\_LENGTH  
Zurückgegeben: Immer

### ***ApplicationPid***

Beschreibung: Die Prozess-ID des Betriebssystems der Anwendung.  
ID: MQIACF\_PROCESS\_ID  
Datentyp: MQCFIN  
Zurückgegeben: Immer

### ***ApplicationTid***

Beschreibung: Die IBM MQ-Thread-ID der Verbindung in der Anwendung  
ID: MQIACF\_THREAD\_ID  
Datentyp: MQCFIN  
Zurückgegeben: Immer

### ***UserId***

Beschreibung: Der Benutzer-ID-Kontext der Anwendung.  
ID: MQCACF\_USER\_IDENTIFIER  
Datentyp: MQCFST  
Maximale Länge: MQ\_USER\_ID\_LENGTH  
Zurückgegeben: Immer

### ***ConnDate***

Beschreibung: Datum der MQCONN-Operation  
ID: MQCAMO\_CONN\_DATE  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH

Zurückgegeben: Wenn verfügbar

### **ConnTime**

Beschreibung: Zeit der MQCONN-Operation  
ID: MQCAMO\_CONN\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Wenn verfügbar

### **ConnName**

Beschreibung: Verbindungsname für Clientverbindung  
ID: MQCACH\_CONNECTION\_NAME  
Datentyp: MQCFST  
Maximale Länge: MQ\_CONN\_NAME\_LENGTH  
Zurückgegeben: Wenn verfügbar

### **ChannelName**

Beschreibung: Kanalname für Clientverbindung  
ID: MQCACH\_CHANNEL\_NAME  
Datentyp: MQCFST  
Maximale Länge: MQ\_CHANNEL\_NAME\_LENGTH  
Zurückgegeben: Wenn verfügbar

### **V 9.1.0 RemoteProduct**

Beschreibung: Ferne Produkt-ID für Clientverbindung, wie im Feld **RPRODUCT** von DISPLAY CHSTATUS angezeigt  
ID: MQCACH\_REMOTE\_PRODUCT  
Datentyp: MQCFST  
Maximale Länge: MQ\_REMOTE\_PRODUCT\_LENGTH  
Zurückgegeben: Wenn verfügbar

### **V 9.1.0 RemoteVersion**

Beschreibung: Ferne Produktversion für Clientverbindung, wie im Feld **RVERSION** von DISPLAY CHSTATUS angezeigt  
ID: MQCACH\_REMOTE\_VERSION  
Datentyp: MQCFST  
Maximale Länge: MQ\_REMOTE\_VERSION\_LENGTH  
Zurückgegeben: Wenn verfügbar

### **DiscDate**

Beschreibung: Datum der MQDISC-Operation

ID: MQCAMO\_DISC\_DATE  
Datentyp: MQCFST  
Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Wenn verfügbar

### ***DiscTime***

Beschreibung: Zeit der MQDISC-Operation  
ID: MQCAMO\_DISC\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Wenn verfügbar

### ***DiscType***

Beschreibung: Typ der Verbindung  
ID: MQIAMO\_DISC\_TYPE  
Datentyp: MQCFIN  
Werte: Folgende Werte sind möglich:  
**MQDISCONNECT\_NORMAL**  
Angefordert von Anwendung  
**MQDISCONNECT\_IMPLICIT**  
Abnormale Beendigung der Anwendung  
**MQDISCONNECT\_Q\_MGR**  
Verbindung des WS-Managers unterbrochen  
Zurückgegeben: Wenn verfügbar

### ***OpenCount***

Beschreibung: Die Anzahl der erfolgreich geöffneten Objekte, durch direkte Ausgabe eines Aufrufs an MQOPEN oder durch Verwendung des Verbs MQPUT1. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_OPENS  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar

### ***OpenFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche zum Öffnen eines Objekts. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_OPENS\_FAILED  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar

### ***CloseCount***

Beschreibung: Die Anzahl der geschlossenen Objekte. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_CLOSES  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar

### ***CloseFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, ein Objekt zu schließen. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_CLOSES\_FAILED  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar

### ***PutCount***

Beschreibung: Die Anzahl persistenter und nicht persistenter Nachrichten, die erfolgreich in eine Warteschlange gestellt wurden, mit Ausnahme der Nachrichten, die mit dem MQPUT1-Aufruf gesendet wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).

ID: MQIAMO\_PUTS  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar

### ***PutFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche zum Angeben einer Nachricht.

ID: MQIAMO\_PUTS\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar

### ***Put1Count***

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Aufrufen erfolgreich in die Warteschlange gestellt wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).

ID: MQIAMO\_PUT1S  
Datentyp: MQCFIL  
In PCF-Gruppe enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***Put1FailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche zum Angeben einer Nachricht unter Verwendung von MQPUT1-Aufrufen

ID: MQIAMO\_PUT1S\_FAILED  
Datentyp: MQCFIN

In PCF-Gruppe enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### **PutBytes**

Beschreibung: Die Anzahl der Byte, die mit put-aufrufe für persistente und nicht persistente Nachrichten geschrieben werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#) ).  
ID: MQIAMO64\_PUT\_BYTES  
Datentyp: MQCFIL64  
Zurückgegeben: Wenn verfügbar

### **GetCount**

Beschreibung: Die Anzahl der erfolgreichen destruktiven MQGET-Aufrufe für persistente und nicht persistente Nachrichten. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#) ).  
ID: MQIAMO\_GETS  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar

### **GetFailCount**

Beschreibung: Die Anzahl der fehlgeschlagenen destruktiven MQGET-Aufrufe.  
ID: MQIAMO\_GETS\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar

### **GetBytes**

Beschreibung: Die Gesamtzahl der Byte, die für persistente und nicht persistente Nachrichten abgerufen wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#) ).  
ID: MQIAMO64\_GET\_BYTES  
Datentyp: MQCFIL64  
Zurückgegeben: Wenn verfügbar

### **BrowseCount**

Beschreibung: Die Anzahl der erfolgreichen nicht-destruktiven MQGET-Aufrufe für persistente und nicht persistente Nachrichten. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#) ).  
ID: MQIAMO\_BROWSES  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar

### **BrowseFailCount**

Beschreibung: Die Anzahl der nicht erfolgreichen MQGET-Aufrufe ohne Löschen.

ID: MQIAMO\_BROWSES\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar

### ***BrowseBytes***

Beschreibung: Die Gesamtzahl der Byte, die für persistente und nicht persistente Nachrichten durchsucht wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64\_BROWSE\_BYTES  
Datentyp: MQCFIL64  
Zurückgegeben: Wenn verfügbar

### ***CommitCount***

Beschreibung: Die Anzahl der erfolgreichen Transaktionen. Diese Zahl schließt die Transaktionen ein, die implizit von der verbundenen Anwendung festgeschrieben wurden. Commit-Anforderungen, bei denen keine ausstehenden Arbeiten vorhanden sind, werden in dieser Anzahl berücksichtigt.

ID: MQIAMO\_COMMITS  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar

### ***CommitFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Transaktion abzuschließen.

ID: MQIAMO\_COMMITS\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar

### ***BackCount***

Beschreibung: Die Anzahl der verarbeiteten Zurücksetzungen, einschließlich impliziter Zurücksetzungen aufgrund einer abnormalen Abschaltung.

ID: MQIAMO\_BACKOUTS  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar

### ***InqCount***

Beschreibung: Die Anzahl der erfolgreichen Objekte, die in den Quot-Wert versetzt wurden. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indiziert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_INQS  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar

### ***InqFailCount***

Beschreibung: Die Anzahl der fehlgeschlagenen Versuche, die Objekt inquire zu versuchen. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_INQS\_FAILED

Datentyp: MQCFIL

Zurückgegeben: Wenn verfügbar

### ***SetCount***

Beschreibung: Die Anzahl der erfolgreichen MQSET-Aufrufe. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_SETS

Datentyp: MQCFIL

Zurückgegeben: Wenn verfügbar

### ***SetFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQSET-Aufrufe. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_SETS\_FAILED

Datentyp: MQCFIL

Zurückgegeben: Wenn verfügbar

### ***SubCountDur***

Beschreibung: Die Anzahl der erfolgreichen Subskriptionsanforderungen, die permanente Subskriptionen erstellt, geändert oder wiederaufgenommen haben. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.

0 = Die Anzahl der erstellten Subskriptionen

1 = Die Anzahl der geänderten Subskriptionen

2 = Die Anzahl der wiederaufgenommenen Subskriptionen

ID: MQIAMO\_SUBS\_DUR

Datentyp: MQCFIL

Zurückgegeben: Wenn verfügbar.

### ***SubCountNDur***

Beschreibung: Die Anzahl der erfolgreichen Subskriptionsanforderungen, die nicht permanente Subskriptionen erstellt, geändert oder wieder aufgenommen haben. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.

0 = Die Anzahl der erstellten Subskriptionen

1 = Die Anzahl der geänderten Subskriptionen

2 = Die Anzahl der wiederaufgenommenen Subskriptionen

ID: MQIAMO\_SUBS\_NDUR

Datentyp: MQCFIL

Zurückgegeben: Wenn verfügbar.

### ***SubFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Subskriptionsanforderungen.  
ID: MQIAMO\_SUBS\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

### ***UnsubCountDur***

Beschreibung: Die Anzahl der erfolgreichen Abmeldeanforderungen für permanente Subskriptionen. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.  
0-Die Subskription wurde geschlossen, aber nicht entfernt.  
1-Die Subskription wurde geschlossen und entfernt.  
ID: MQIAMO\_UNSUBS\_DUR  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar.

### ***UnsubCountNDur***

Beschreibung: Die Anzahl der erfolgreichen Abmeldeanforderungen für permanente Subskriptionen. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.  
0-Die Subskription wurde geschlossen, aber nicht entfernt.  
1-Die Subskription wurde geschlossen und entfernt.  
ID: MQIAMO\_UNSUBS\_NDUR  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar.

### ***UnsubFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Abmeldeanforderungen.  
ID: MQIAMO\_UNSUBS\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

### ***SubRqCount***

Beschreibung: Die Anzahl der erfolgreichen MQSUBRQ-Anforderungen.  
ID: MQIAMO\_SUBRQS  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

### ***SubRqFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQSUB-Anforderungen.  
ID: MQIAMO\_SUBRQS\_FAILED  
Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar.

### ***CBCount***

Beschreibung: Die Anzahl der erfolgreichen MQCB-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indiziert werden.

0-Ein Callback wurde erstellt oder geändert.

1-Ein Callback wurde entfernt

2-Ein Callback wurde wieder aufgenommen

3-Ein Callback wurde ausgesetzt

ID: MQIAMO\_CBS

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar.

### ***CBFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQCB-Anforderungen.

ID: MQIAMO\_CBS\_FAILED

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar.

### ***CtlCount***

Beschreibung: Die Anzahl der erfolgreichen MQCTL-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indiziert werden.

0-Die Verbindung wurde gestartet.

1-Die Verbindung wurde gestoppt.

2-Die Verbindung wurde wieder aufgenommen

3-Die Verbindung wurde unterbrochen.

ID: MQIAMO\_CTLS

Datentyp: MQCFIL

Zurückgegeben: Wenn verfügbar.

### ***CtlFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQCTL-Anforderungen.

ID: MQIAMO\_CTLS\_FAILED

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar.

### ***StatCount***

Beschreibung: Die Anzahl der erfolgreichen MQSTAT-Anforderungen.

ID: MQIAMO\_STATS.

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar.

### **StatFailCount**

Beschreibung: Die Anzahl der nicht erfolgreichen MQSTAT-Anforderungen.  
ID: MQIAMO\_STATS\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

### **PutTopicCount**

Beschreibung: Die Anzahl persistenter und nicht persistenter Nachrichten, die erfolgreich in ein Thema gestellt wurden, mit Ausnahme von Nachrichten, die mit dem MQPUT1-Aufruf ausgegeben werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).  
Hinweis: Nachrichten, die einen Warteschlangenalias verwenden, der in ein Thema aufgelöst wird, sind in diesem Wert enthalten.  
ID: MQIAMO\_TOPIC\_PUTS  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar.

### **PutTopicFailCount**

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Nachricht in ein Thema zu stellen.  
ID: MQIAMO\_TOPIC\_PUTS\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

### **Put1TopicCount**

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Aufrufen erfolgreich in ein Thema gestellt wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).  
Hinweis: Nachrichten, die einen Warteschlangenalias verwenden, der in ein Thema aufgelöst wird, sind in diesem Wert enthalten.  
ID: MQIAMO\_TOPIC\_PUT1S  
Datentyp: MQCFIL  
Zurückgegeben: Wenn verfügbar.

### **Put1TopicFailCount**

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Nachricht mit Hilfe von MQPUT1-Aufrufen in ein Thema zu stellen.  
ID: MQIAMO\_TOPIC\_PUT1S\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

### ***PutTopicBytes***

Beschreibung:	Die Anzahl der Byte, die mit Hilfe von put-Aufrufen für persistente und nicht persistente Nachrichten geschrieben werden, die in eine Veröffentlichungsoperation aufgelöst werden. Dies ist die Anzahl der Byte, die von der Anwendung gestellt werden, und nicht die resultierende Anzahl Byte, die an Subskribenten zugestellt werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO64_TOPIC_PUT_BYTES
Datentyp:	MQCFIL64
Zurückgegeben:	Wenn verfügbar.

### ***Nachrichtendaten der Warteschlangenabrechnung***

Auf dieser Seite können Sie die Struktur einer Warteschlangenabrechnungsnachricht anzeigen.

Nachrichtenname:	Warteschlangenabrechnungsnachricht.
Plattformen:	Alle außer IBM MQ for z/OS.
Systemwarteschlange:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

### ***QueueManager***

Beschreibung:	Der Name des Warteschlangenmanagers.
ID:	MQCA_Q_MGR_NAME
Datentyp:	MQCFST
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH
Zurückgegeben:	Immer

### ***IntervalStartDate***

Beschreibung:	Das Datum des Startzeitbeginns des Überwachungszeitraums
ID:	MQCAMO_START_DATE
Datentyp:	MQCFST
Maximale Länge:	MQ_DATE_LENGTH
Zurückgegeben:	Immer

### ***IntervalStartTime***

Beschreibung:	Der Zeitpunkt des Startzeitpunktes des Überwachungszeitraums.
ID:	MQCAMO_START_TIME
Datentyp:	MQCFST
Maximale Länge:	MQ_TIME_LENGTH
Zurückgegeben:	Immer

### ***IntervalEndDate***

Beschreibung:	Das Datum des Endes des Überwachungszeitraums
ID:	MQCAMO_END_DATE
Datentyp:	MQCFST

Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Immer

### ***IntervalEndTime***

Beschreibung: Die Zeit bis zum Ende des Überwachungszeitraums  
ID: MQCAMO\_END\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Immer

### ***CommandLevel***

Beschreibung: Die Befehlsebene des Warteschlangenmanagers  
ID: MQIA\_COMMAND\_LEVEL  
Datentyp: MQCFIN  
Zurückgegeben: Immer

### ***ConnectionId***

Beschreibung: Die Verbindungs-ID für die IBM MQ-Verbindung  
ID: MQBACF\_CONNECTION\_ID  
Datentyp: MQCFBS  
Maximale Länge: MQ\_CONNECTION\_ID\_LENGTH  
Zurückgegeben: Immer

### ***SeqNumber***

Beschreibung: Die Folgenummer. Dieser Wert wird für jeden nachfolgenden Datensatz für lange laufende Verbindungen erhöht.  
ID: MQIACF\_SEQUENCE\_NUMBER  
Datentyp: MQCFIN  
Zurückgegeben: Immer

### ***ApplicationName***

Beschreibung: Der Name der Anwendung. Der Inhalt dieses Felds entspricht dem Inhalt des Felds PutApplName im Nachrichtendeskriptor.  
ID: MQCACF\_APPL\_NAME  
Datentyp: MQCFST  
Maximale Länge: MQ\_APPL\_NAME\_LENGTH  
Zurückgegeben: Immer

### ***ApplicationPid***

Beschreibung: Die Prozess-ID des Betriebssystems der Anwendung.  
ID: MQIACF\_PROCESS\_ID  
Datentyp: MQCFIN

Zurückgegeben: Immer

### ***ApplicationTid***

Beschreibung: Die IBM MQ-Thread-ID der Verbindung in der Anwendung

ID: MQIACF\_THREAD\_ID

Datentyp: MQCFIN

Zurückgegeben: Immer

### ***UserId***

Beschreibung: Der Benutzer-ID-Kontext der Anwendung.

ID: MQCACF\_USER\_IDENTIFIER

Datentyp: MQCFST

Maximale Länge: MQ\_USER\_ID\_LENGTH

Zurückgegeben: Immer

### ***ObjectCount***

Beschreibung: Die Anzahl der Warteschlangen, auf die in dem Intervall zugegriffen wurde, für das Abrechnungsdaten aufgezeichnet wurden. Dieser Wert wird auf die Anzahl der in der Nachricht enthaltenen *QAccountingData* -PCF-Gruppen festgelegt.

ID: MQIAMO\_OBJECT\_COUNT

Datentyp: MQCFIN

Zurückgegeben: Immer

### ***QAccountingData***

Beschreibung: Gruppierte Parameter, die Abrechnungsdetails für eine Warteschlange angeben

ID: MQGACF\_Q\_ACCOUNTING\_DATA

Datentyp: MQCFGR

Parameter in Gruppe:	<i>QName</i> <i>CreateDate</i> <i>CreateTime</i> <i>QType</i> <i>QDefinitionType</i> <i>OpenCount</i> <i>OpenDate</i> <i>OpenTime</i> <i>CloseDate</i> <i>CloseTime</i> <i>PutCount</i> <i>PutFailCount</i> <i>Put1Count</i> <i>Put1FailCount</i> <i>PutBytes</i> <i>PutMinBytes</i> <i>PutMaxBytes</i> <i>GetCount</i> <i>GetFailCount</i> <i>GetBytes</i> <i>GetMinBytes</i> <i>GetMaxBytes</i> <i>BrowseCount</i> <i>BrowseFailCount</i> <i>BrowseBytes</i> <i>BrowseMinBytes</i> <i>BrowseMaxBytes</i> <i>TimeOnQMin</i> <i>TimeOnQAvg</i> <i>TimeOnQMax</i>
----------------------	---

Zurückgegeben: Immer

### ***QName***

Beschreibung:	Der Name der Warteschlange.
ID:	MQCA_Q_NAME
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Wenn verfügbar

### ***CreateDate***

Beschreibung:	Das Datum, an dem die Warteschlange erstellt wurde.
ID:	MQCA_CREATION_DATE
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>

Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Wenn verfügbar

### **CreateTime**

Beschreibung: Die Zeit, zu der die Warteschlange erstellt wurde.  
ID: MQCA\_CREATION\_TIME  
Datentyp: MQCFST  
In PCF-Gruppe enthalten: *QAccountingData*  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Wenn verfügbar

### **QType**

Beschreibung: Der Typ der Warteschlange.  
ID: MQIA\_Q\_TYPE  
Datentyp: MQCFIN  
In PCF-Gruppe enthalten: *QAccountingData*  
Wert: MQQT\_LOCAL  
Zurückgegeben: Wenn verfügbar

### **QDefinitionType**

Beschreibung: Der Typ der Warteschlangendefinition.  
ID: MQIA\_DEFINITION\_TYPE  
Datentyp: MQCFIN  
In PCF-Gruppe enthalten: *QAccountingData*  
Werte: Mögliche Werte:  
**MQQDT\_PREDEFINED**  
**MQQDT\_PERMANENT\_DYNAMIC**  
**MQQDT\_TEMPORARY\_DYNAMIC**  
Zurückgegeben: Wenn verfügbar

### **OpenCount**

Beschreibung: Gibt an, wie oft diese Warteschlange in dem betreffenden Intervall von der Anwendung geöffnet wurde, durch direkte Ausgabe eines Aufrufs an MQOPEN oder durch Verwendung des Verbs MQPUT1.  
ID: MQIAMO\_OPENS  
Datentyp: MQCFIL  
In PCF-Gruppe enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### **OpenDate**

Beschreibung:	Das Datum, an dem die Warteschlange zum ersten Öffnen in diesem Aufzeichnungsintervall geöffnet wurde. Wenn die Warteschlange zu Beginn dieses Intervalls bereits geöffnet war, gibt dieser Wert das Datum wieder, an dem die Warteschlange ursprünglich geöffnet wurde.
ID:	MQCAMO_OPEN_DATE
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **OpenTime**

Beschreibung:	Die Zeit, zu der die Warteschlange zum ersten Mal in diesem Aufzeichnungsintervall geöffnet wurde. Wenn die Warteschlange bereits zu Beginn dieses Intervalls geöffnet war, spiegelt dieser Wert die Zeit wider, zu der die Warteschlange ursprünglich geöffnet wurde.
ID:	MQCAMO_OPEN_TIME
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **CloseDate**

Beschreibung:	Das Datum der endgültigen Schließung der Warteschlange in diesem Aufzeichnungsintervall. Wenn die Warteschlange noch geöffnet ist, wird der Wert nicht zurückgegeben.
ID:	MQCAMO_CLOSE_DATE
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **CloseTime**

Beschreibung:	Die Zeit der endgültigen Schließung der Warteschlange in diesem Aufzeichnungsintervall. Wenn die Warteschlange noch geöffnet ist, wird der Wert nicht zurückgegeben.
ID:	MQCAMO_CLOSE_TIME
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **PutCount**

Beschreibung:	Die Anzahl der persistenten und nicht persistenten Nachrichten, die erfolgreich in die Warteschlange gestellt wurden, mit Ausnahme von MQPUT1-Aufrufen. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO_PUTS
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **PutFailCount**

Beschreibung:	Die Anzahl der nicht erfolgreichen Versuche zum Angeben einer Nachricht mit Ausnahme von MQPUT1-Aufrufen.
ID:	MQIAMO_PUTS_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **Put1Count**

Beschreibung:	Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Aufrufen erfolgreich in die Warteschlange gestellt wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO_PUT1S
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **Put1FailCount**

Beschreibung:	Die Anzahl der nicht erfolgreichen Versuche zum Angeben einer Nachricht unter Verwendung von MQPUT1-Aufrufen
ID:	MQIAMO_PUT1S_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **PutBytes**

Beschreibung:	Die Gesamtzahl der Byte, die für persistente und nicht persistente Nachrichten gestellt werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO64_PUT_BYTES

Datentyp: MQCFIL64  
In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***PutMinBytes***

Beschreibung: Die kleinste persistente und nicht persistente Nachrichtengröße, die in die Warteschlange gestellt wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#) ).

ID: MQIAMO\_PUT\_MIN\_BYTES  
Datentyp: MQCFIL  
In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***PutMaxBytes***

Beschreibung: Die größte persistente und nicht persistente Nachrichtengröße, die in die Warteschlange gestellt wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#) ).

ID: MQIAMO\_PUT\_MAX\_BYTES  
Datentyp: MQCFIL  
In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***GeneratedMsgCount***

Beschreibung: Die Anzahl der generierten Nachrichten. Generierte Nachrichten:

- Warteschlangenlänge-Hi-Ereignisse
- Niedrige Ereignisse in der Warteschlange

ID: MQIAMO\_GENERATED\_MSGS  
Datentyp: MQCFIN  
In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***GetCount***

Beschreibung: Die Anzahl der erfolgreichen destruktiven MQGET-Aufrufe für persistente und nicht persistente Nachrichten. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#) ).

ID: MQIAMO\_GETS  
Datentyp: MQCFIL  
In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### **GetFailCount**

Beschreibung:	Die Anzahl der fehlgeschlagenen destruktiven MQGET-Aufrufe.
ID:	MQIAMO_GETS_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **GetBytes**

Beschreibung:	Die Anzahl der Byte, die in destruktiven MQGET-Aufrufen für persistente und nicht persistente Nachrichten gelesen werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO64_GET_BYTES
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **GetMinBytes**

Beschreibung:	Die Größe der kleinsten persistenten und nicht persistenten Nachricht, die aus der Warteschlange abgerufen wurde. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO_GET_MIN_BYTES
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **GetMaxBytes**

Beschreibung:	Die Größe der größten persistenten und nicht persistenten Nachricht, die aus der Warteschlange abgerufen wurde. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO_GET_MAX_BYTES
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### **BrowseCount**

Beschreibung:	Die Anzahl der erfolgreichen nicht-destruktiven MQGET-Aufrufe für persistente und nicht persistente Nachrichten. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO_BROWSES
Datentyp:	MQCFIL

In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***BrowseFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQGET-Aufrufe ohne Löschen.  
ID: MQIAMO\_BROWSES\_FAILED  
Datentyp: MQCFIN  
In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***BrowseBytes***

Beschreibung: Die Anzahl der Byte, die in nicht-destruktiven MQGET-Aufrufen gelesen wurden, die persistente Nachrichten zurückgegeben haben  
ID: MQIAMO64\_BROWSE\_BYTES  
Datentyp: MQCFIL64  
In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***BrowseMinBytes***

Beschreibung: Die Größe der kleinsten persistenten und nicht persistenten Nachricht, die aus der Warteschlange durchsucht wurde. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#) ).  
ID: MQIAMO\_BROWSE\_MIN\_BYTES  
Datentyp: MQCFIL  
In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***BrowseMaxBytes***

Beschreibung: Die Größe der größten persistenten und nicht persistenten Nachricht, die aus der Warteschlange durchsucht wurde. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#) ).  
ID: MQIAMO\_BROWSE\_MAX\_BYTES  
Datentyp: MQCFIL  
In PCF-Gruppe  
enthalten: *QAccountingData*  
Zurückgegeben: Wenn verfügbar

### ***TimeOnQMin***

Beschreibung:	Die kürzeste Zeit, in der eine persistente und nicht persistente Nachricht in der Warteschlange verblieb, bevor sie destruktiv abgerufen wurde, in Mikrosekunden. Für Nachrichten, die unter Synchronisationspunkt abgerufen werden, enthält dieser Wert keine Zeit, bevor die get-Operation festgeschrieben wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO64_Q_TIME_MIN
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### ***TimeOnQAvg***

Beschreibung:	Die durchschnittliche Zeit, in der eine persistente und nicht persistente Nachricht in der Warteschlange verblieb, bevor sie destruktiv abgerufen wurde, in Mikrosekunden. Für Nachrichten, die unter Synchronisationspunkt abgerufen werden, enthält dieser Wert keine Zeit, bevor die get-Operation festgeschrieben wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO64_Q_TIME_AVG
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### ***TimeOnQMax***

Beschreibung:	Die längste Zeit, in der eine persistente und nicht persistente Nachricht in der Warteschlange verblieb, bevor sie destruktiv abgerufen wurde (in Mikrosekunden). Für Nachrichten, die unter Synchronisationspunkt abgerufen werden, enthält dieser Wert keine Zeit, bevor die get-Operation festgeschrieben wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO64_Q_TIME_MAX
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

### ***CBCCount***

Beschreibung:	Die Anzahl der erfolgreichen MQCB-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indiziert werden. 0-Ein Callback wurde erstellt oder geändert. 1-Ein Callback wurde entfernt 2-Ein Callback wurde wieder aufgenommen 3-Ein Callback wurde ausgesetzt
---------------	--

ID: MQIAMO\_CBS  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

### ***CBFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQCB-Anforderungen.  
ID: MQIAMO\_CBS\_FAILED  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

### ***MQI-Statistikdaten, Nachrichtendaten***

Verwenden Sie diese Seite, um die Struktur einer MQI-Statistiknachricht anzuzeigen.

Nachrichtenname:	MQI-Statistiknachricht.
Plattformen:	Alle außer IBM MQ for z/OS.
Systemwarteschlange:	SYSTEM.ADMIN.STATISTICS.QUEUE.

### ***QueueManager***

Beschreibung: Name des Warteschlangenmanagers.  
ID: MQCA\_Q\_MGR\_NAME.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH.  
Zurückgegeben: Immer.

### ***IntervalStartDate***

Beschreibung: Das Datum zu Beginn des Überwachungszeitraums.  
ID: MQCAMO\_START\_DATE.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Immer.

### ***IntervalStartTime***

Beschreibung: Die Uhrzeit am Anfang des Überwachungszeitraums.  
ID: MQCAMO\_START\_TIME.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Immer.

### ***IntervalEndDate***

Beschreibung: Das Datum am Ende des Überwachungszeitraums.  
ID: MQCAMO\_END\_DATE.  
Datentyp: MQCFST.

Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Immer.

### ***IntervalEndTime***

Beschreibung: Die Zeit am Ende des Überwachungszeitraums.  
ID: MQCAMO\_END\_TIME.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Immer.

### ***CommandLevel***

Beschreibung: Die Befehlsebene des Warteschlangenmanagers.  
ID: MQIA\_COMMAND\_LEVEL.  
Datentyp: MQCFIN.  
Zurückgegeben: Immer.

### ***ConnCount***

Beschreibung: Die Anzahl der erfolgreichen Verbindungen zum WS-Manager.  
ID: MQIAMO\_CONNS.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### ***ConnFailCount***

Beschreibung: Die Anzahl der fehlgeschlagenen Verbindungsversuche.  
ID: MQIAMO\_CONNS\_FAILED.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### ***ConnsMax***

Beschreibung: Die maximale Anzahl gleichzeitiger Verbindungen im Aufzeichnungsintervall.  
ID: MQIAMO\_CONNS\_MAX.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### ***DiscCount***

Beschreibung: Die Anzahl der Trennungsverbindungen vom Warteschlangenmanager. Dies ist eine ganze Zahl, die durch die folgenden Konstanten indexiert wird:

- MQDISCONNECT\_NORMAL
- MQDISCONNECT\_IMPLICIT
- MQDISCONNECT\_Q\_MGR

ID: MQIAMO\_DISCS.  
Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***OpenCount***

Beschreibung: Die Anzahl der erfolgreich geöffneten Objekte, durch direkte Ausgabe eines Aufrufs an MQOPEN oder durch Verwendung des Verbs MQPUT1. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_OPENS.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***OpenFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche zum Öffnen von Objekten. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_OPENS\_FAILED.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***CloseCount***

Beschreibung: Die Anzahl der erfolgreich abgeschlossenen Objekte. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_CLOSES.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***CloseFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche zum Schließen von Objekten. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_CLOSES\_FAILED.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***InqCount***

Beschreibung: Die Anzahl der Objekte, die erfolgreich in den Quiet-Wert versetzt wurden. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_INQS.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***InqFailCount***

Beschreibung: Die Anzahl der fehlgeschlagenen Versuche, die Objekt inquire zu versuchen. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO\_INQS\_FAILED.  
Datentyp: MQCFIL.  
Zurückgegeben: Wenn verfügbar.

### **SetCount**

Beschreibung: Die Anzahl der erfolgreich aktualisierten Objekte (SET). Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#) ).  
ID: MQIAMO\_SETS.  
Datentyp: MQCFIL.  
Zurückgegeben: Wenn verfügbar.

### **SetFailCount**

Beschreibung: Die Anzahl der nicht erfolgreichen SET-Versuche. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#) ).  
ID: MQIAMO\_SETS\_FAILED.  
Datentyp: MQCFIL.  
Zurückgegeben: Wenn verfügbar.

### **PutCount**

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die erfolgreich in eine Warteschlange gestellt wurden, mit Ausnahme von MQPUT1-Anforderungen. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#) ).  
ID: MQIAMO\_PUTS.  
Datentyp: MQCFIL.  
Zurückgegeben: Wenn verfügbar.

### **PutFailCount**

Beschreibung: Die Anzahl der fehlgeschlagenen Versuche zum Angeben von Nachrichten.  
ID: MQIAMO\_PUTS\_FAILED.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### **Put1Count**

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Anforderungen erfolgreich in eine Warteschlange gestellt wurden. Dieser Parameter ist eine ganze Zahl, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#) ).  
ID: MQIAMO\_PUT1S.  
Datentyp: MQCFIL.  
Zurückgegeben: Wenn verfügbar.

### **Put1FailCount**

Beschreibung:	Die Anzahl der nicht erfolgreichen Versuche, eine persistente und nicht persistente Nachricht mit Hilfe von MQPUT1-Anforderungen in eine Warteschlange zu stellen. Dieser Parameter ist eine ganze Zahl, die nach dem Persistenzwert indexiert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO_PUT1S_FAILED.
Datentyp:	MQCFIL.
Zurückgegeben:	Wenn verfügbar.

### **PutBytes**

Beschreibung:	Die Anzahl Byte für persistente und nicht persistente Nachrichten, die bei der Verwendung von put-Requests geschrieben wurden. Dieser Parameter ist eine ganze Zahl, die nach dem Persistenzwert indexiert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO64_PUT_BYTES.
Datentyp:	MQCFIL64.
Zurückgegeben:	Wenn verfügbar.

### **GetCount**

Beschreibung:	Die Anzahl der erfolgreichen zerstörerischen Anforderungen zum Abrufen persistenter und nicht persistenter Nachrichten. Dieser Parameter ist eine ganze Zahl, die nach dem Persistenzwert indexiert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO_GETS.
Datentyp:	MQCFIL.
Zurückgegeben:	Wenn verfügbar.

### **GetFailCount**

Beschreibung:	Die Anzahl der nicht erfolgreichen Anforderungen zum Löschen von Löschen.
ID:	MQIAMO_GETS_FAILED.
Datentyp:	MQCFIN.
Zurückgegeben:	Wenn verfügbar.

### **GetBytes**

Beschreibung:	Die Anzahl der gelesenen Byte in destruktiven Abrufe ruft Anforderungen für persistente und nicht persistente Nachrichten ab. Dieser Parameter ist eine ganze Zahl, die nach dem Persistenzwert indexiert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO64_GET_BYTES.
Datentyp:	MQCFIL64.
Zurückgegeben:	Wenn verfügbar.

### **BrowseCount**

Beschreibung:	Die Anzahl der erfolgreichen nicht-destruktiven Anforderungen zum Abrufen von persistenten und nicht persistenten Nachrichten. Dieser Parameter ist eine ganze Zahl, die nach dem Persistenzwert indexiert ist (siehe <a href="#">Referenznote 2</a> ).
ID:	MQIAMO_BROWSES.

Datentyp: MQCFIL.  
Zurückgegeben: Wenn verfügbar.

### ***BrowseFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen get-Anforderungen ohne Löschen.  
ID: MQIAMO\_BROWSES\_FAILED.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### ***BrowseBytes***

Beschreibung: Die Anzahl der Byte, die in nicht-destruktiven Leseanforderungen für persistente und nicht persistente Nachrichten gelesen werden. Dieser Parameter ist eine ganze Zahl, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#) ).  
ID: MQIAMO64\_BROWSE\_BYTES.  
Datentyp: MQCFIL64.  
Zurückgegeben: Wenn verfügbar.

### ***CommitCount***

Beschreibung: Die Anzahl der erfolgreich abgeschlossenen Transaktionen. Diese Zahl schließt Transaktionen ein, die implizit von der Anwendungsabschaltung festgeschrieben wurden, und Festschreiben von Anforderungen, bei denen es keine ausstehende Arbeit gibt.  
ID: MQIAMO\_COMMITS.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### ***CommitFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Transaktion abzuschließen.  
ID: MQIAMO\_COMMITS\_FAILED.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### ***BackCount***

Beschreibung: Die Anzahl der verarbeiteten Zurücksetzungen, einschließlich impliziter Zurücksetzungen bei abnormaler Verbindung.  
ID: MQIAMO\_BACKOUTS.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### ***ExpiredMsgCount***

Beschreibung: Die Anzahl persistenter und nicht persistenter Nachrichten, die gelöscht wurden, weil sie abgelaufen waren, bevor sie abgerufen werden konnten.  
ID: MQIAMO\_MSGS\_EXPIRED.  
Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

### **PurgeCount**

Beschreibung: Die Häufigkeit, mit der die Warteschlange gelöscht wurde.

ID: MQIAMO\_MSGS\_PURGED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

### **SubCountDur**

Beschreibung: Die Anzahl der erfolgreichen Subskriptionsanforderungen, die permanente Subskriptionen erstellt, geändert oder wiederaufgenommen haben. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.

0 = Die Anzahl der erstellten Subskriptionen

1 = Die Anzahl der geänderten Subskriptionen

2 = Die Anzahl der wiederaufgenommenen Subskriptionen

ID: MQIAMO\_SUBS\_DUR.

Datentyp: MQCFIL

Zurückgegeben: Wenn verfügbar.

### **SubCountNDur**

Beschreibung: Die Anzahl der erfolgreichen Subskriptionsanforderungen, die nicht permanente Subskriptionen erstellt, geändert oder wieder aufgenommen haben. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.

0 = Die Anzahl der erstellten Subskriptionen

1 = Die Anzahl der geänderten Subskriptionen

2 = Die Anzahl der wiederaufgenommenen Subskriptionen

ID: MQIAMO\_SUBS\_NDUR.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### **SubFailCount**

Beschreibung: Die Anzahl der nicht erfolgreichen Subskriptionsanforderungen.

ID: MQIAMO\_SUBS\_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

### **UnsubCountDur**

Beschreibung: Die Anzahl der erfolgreichen Abmeldeanforderungen für permanente Subskriptionen. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.

0-Die Subskription wurde geschlossen, aber nicht entfernt.

1-Die Subskription wurde geschlossen und entfernt.

ID: MQIAMO\_UNSUBS\_DUR.  
Datentyp: MQCFIL.  
Zurückgegeben: Wenn verfügbar.

#### ***UnsubCountNDur***

Beschreibung: Die Anzahl der erfolgreichen Abmeldeanforderungen für nicht permanente Subskriptionen. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.  
0-Die Subskription wurde geschlossen, aber nicht entfernt.  
1-Die Subskription wurde geschlossen und entfernt.

ID: MQIAMO\_UNSUBS\_NDUR.  
Datentyp: MQCFIL.  
Zurückgegeben: Wenn verfügbar.

#### ***UnsubFailCount***

Beschreibung: Die Anzahl der fehlgeschlagenen Abmeldeanforderungen.  
ID: MQIAMO\_UNSUBS\_FAILED.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

#### ***SubRqCount***

Beschreibung: Die Anzahl der erfolgreichen MQSUBRQ-Anforderungen.  
ID: MQIAMO\_SUBRQS  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

#### ***SubRqFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQSUBRQ-Anforderungen.  
ID: MQIAMO\_SUBRQS\_FAILED.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

#### ***CBCCount***

Beschreibung: Die Anzahl der erfolgreichen MQCB-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.  
0-Ein Callback wurde erstellt oder geändert.  
1-Ein Callback wurde entfernt  
2-Ein Callback wurde wieder aufgenommen  
3-Ein Callback wurde ausgesetzt  
ID: MQIAMO\_CBS.  
Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***CBFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQCB-Anforderungen.

ID: MQIAMO\_CBS\_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

### ***CtlCount***

Beschreibung: Die Anzahl der erfolgreichen MQCTL-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden:

0-Die Verbindung wurde gestartet.

1-Die Verbindung wurde gestoppt.

2-Die Verbindung wurde wieder aufgenommen

3-Die Verbindung wurde unterbrochen.

ID: MQIAMO\_CTLS.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***CtlFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQCTL-Anforderungen.

ID: MQIAMO\_CTLS\_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

### ***StatCount***

Beschreibung: Die Anzahl der erfolgreichen MQSTAT-Anforderungen.

ID: MQIAMO\_STATS.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

### ***StatFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen MQSTAT-Anforderungen.

ID: MQIAMO\_STATS\_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

### ***SubCountDurHighWater***

**Beschreibung:** Die hohe Wassermarkierung für die Anzahl der permanenten Subskriptionen während des Zeitintervalls. Dies ist ein Array mit Werten, die durch SUBTYPE indexiert werden.

- 0-Die Hochwasser-Markierung für alle permanenten Subskriptionen im System
- 1-Die Hochwassermarkierung für permanente Anwendungssubskriptionen (MQSUBTYPE\_API)
- 2-Die Hochwasser-Markierung für die permanente Administratorsubskription (MQSUBTYPE\_ADMIN)
- 3-Die Obere Wassermarke für dauerhafte Proxy-Subskriptionen (MQSUBTYPE\_PROXY)

**ID:** MQIAMO\_SUB\_DUR\_HIGHWATER

**Datentyp:** MQCFIL.

**Zurückgegeben:** Wenn verfügbar.

### ***SubCountDurLowWater***

**Beschreibung:** Die Niedrigwasserzeichen auf der Anzahl der permanenten Subskriptionen während des Zeitintervalls. Dies ist ein Array mit Werten, die von SUBTYPE indexiert werden.

- 0-Die Niedrigwassermarkierung für alle permanenten Subskriptionen im System
- 1-Die Unterwasser-Markierung für permanente Anwendungssubskriptionen (MQSUBTYPE\_API)
- 2-Die Niedrigwassermarkierung für dauerhafte Verwaltungssubskriptionen (MQSUBTYPE\_ADMIN)
- 3-Die Niedrigwassermarkierung für permanente Proxy-Subskriptionen (MQSUBTYPE\_PROXY)

**ID:** MQIAMO\_SUB\_DUR\_LOWWATER

**Datentyp:** MQCFIL.

**Zurückgegeben:** Wenn verfügbar.

### ***SubCountNDurHighWater***

**Beschreibung:** Die Hochwasser-Markierung für die Anzahl der nicht permanenten Subskriptionen während des Zeitintervalls. Dies ist ein Array mit Werten, die durch SUBTYPE indexiert werden.

- 0-Die Hochwasser-Markierung für alle nicht permanenten Subskriptionen im System
- 1-Die Hochwassermarkierung für nicht permanente Anwendungssubskriptionen (MQSUBTYPE\_API)
- 2-Die Hochwasser-Markierung für die nicht permanente Administratorsubskription (MQSUBTYPE\_ADMIN)
- 3-Die Obere Wassermarke für nicht permanente Proxy-Subskriptionen (MQSUBTYPE\_PROXY)

**ID:** MQIAMO\_SUB\_NDUR\_HIGHWATER

**Datentyp:** MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***SubCountNDurLowWater***

**Beschreibung:** Die Niedrigwassermarkierung für die Anzahl der nicht permanenten Subskriptionen während des Zeitintervalls. Dies ist ein Array mit Werten, die von SUBTYPE indexiert werden.

0-Die Niedrigwassermarkierung für alle nicht permanenten Subskriptionen im System

1-Die Niedrigwassermarkierung für nicht permanente Anwendungssubskriptionen (MQSUBTYPE\_API)

2-Die Unterwasser-Markierung für nicht permanente Verwaltungssubskriptionen (MQSUBTYPE\_ADMIN)

3-Die Unterwasser-Markierung für nicht permanente Proxy-Subskriptionen (MQSUBTYPE\_PROXY)

**ID:** MQIAMO\_SUB\_NDUR\_LOWWATER

**Datentyp:** MQCFIL.

**Zurückgegeben:** Wenn verfügbar.

### ***PutTopicCount***

**Beschreibung:** Die Anzahl persistenter und nicht persistenter Nachrichten, die erfolgreich in ein Thema gestellt wurden, mit Ausnahme von Nachrichten, die mit dem MQPUT1-Aufruf ausgegeben werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).

Hinweis: Nachrichten, die einen Warteschlangenalias verwenden, der in ein Thema aufgelöst wird, sind in diesem Wert enthalten.

**ID:** MQIAMO\_TOPIC\_PUTS.

**Datentyp:** MQCFIL.

**Zurückgegeben:** Wenn verfügbar.

### ***PutTopicFailCount***

**Beschreibung:** Die Anzahl der nicht erfolgreichen Versuche, eine Nachricht in ein Thema zu stellen.

**ID:** MQIAMO\_TOPIC\_PUTS\_FAILED.

**Datentyp:** MQCFIN.

**Zurückgegeben:** Wenn verfügbar.

### ***Put1TopicCount***

**Beschreibung:** Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Aufrufen erfolgreich in ein Thema gestellt wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).

Hinweis: Nachrichten, die einen Warteschlangenalias verwenden, der in ein Thema aufgelöst wird, sind in diesem Wert enthalten.

**ID:** MQIAMO\_TOPIC\_PUT1S.

**Datentyp:** MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***Put1TopicFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Nachricht mit Hilfe von MQPUT1-Aufrufen in ein Thema zu stellen.

ID: MQIAMO\_TOPIC\_PUT1S\_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

### ***PutTopicBytes***

Beschreibung: Die Anzahl der Byte, die mit Hilfe von put-Aufrufen für persistente und nicht persistente Nachrichten geschrieben werden, die in eine Veröffentlichungsoperation aufgelöst werden. Dies ist die Anzahl der Byte, die von der Anwendung gestellt werden, und nicht die resultierende Anzahl an Subskribenten, die an die Subskribenten zugestellt werden, siehe PublishMsgBytes für diesen Wert. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64\_TOPIC\_PUT\_BYTES.

Datentyp: MQCFIL64.

Zurückgegeben: Wenn verfügbar.

### ***PublishMsgCount***

Beschreibung: Die Anzahl der Nachrichten, die innerhalb des Zeitintervalls an Subskriptionen gesendet wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64\_PUBLISH\_MSG\_COUNT

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

### ***PublishMsgBytes***

Beschreibung: Die Anzahl der Byte, die in dem Zeitintervall an Subskriptionen gesendet wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64\_PUBLISH\_MSG\_BYTES

Datentyp: MQCFIL64.

Zurückgegeben: Wenn verfügbar.

### ***Nachrichtendaten der Warteschlangenstatistik***

Verwenden Sie diese Seite, um die Struktur einer Warteschlangenstatistiknachricht anzuzeigen.

Nachrichtenname: Warteschlangenstatistiknachricht.

---

Plattformen: Alle außer IBM MQ for z/OS.

---

Systemwarteschlange: SYSTEM.ADMIN.STATISTICS.QUEUE.

### ***QueueManager***

Beschreibung: Name des Warteschlangenmanagers

ID: MQCA\_Q\_MGR\_NAME  
Datentyp: MQCFST  
Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH  
Zurückgegeben: Immer

### ***IntervalStartDate***

Beschreibung: Das Datum zu Beginn des Überwachungszeitraums  
ID: MQCAMO\_START\_DATE  
Datentyp: MQCFST  
Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Immer

### ***IntervalStartTime***

Beschreibung: Die Uhrzeit am Beginn des Überwachungszeitraums.  
ID: MQCAMO\_START\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Immer

### ***IntervalEndDate***

Beschreibung: Das Datum am Ende des Überwachungszeitraums  
ID: MQCAMO\_END\_DATE  
Datentyp: MQCFST  
Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Immer

### ***IntervalEndTime***

Beschreibung: Die Zeit am Ende des Überwachungszeitraums  
ID: MQCAMO\_END\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Immer

### ***CommandLevel***

Beschreibung: Die Befehlsebene des Warteschlangenmanagers  
ID: MQIA\_COMMAND\_LEVEL  
Datentyp: MQCFIN  
Zurückgegeben: Immer

### **ObjectCount**

Beschreibung:	Die Anzahl der Warteschlangenobjekte, auf die in dem Intervall zugegriffen wurde, für das Statistikdaten aufgezeichnet wurden. Dieser Wert wird auf die Anzahl der in der Nachricht enthaltenen QStatisticsData-PCF-Gruppen gesetzt.
ID:	MQIAMO_OBJECT_COUNT
Datentyp:	MQCFIN
Zurückgegeben:	Immer

### **QStatisticsData**

Beschreibung:	Gruppierte Parameter, die Statistikdetails für eine Warteschlange angeben
ID:	MQGACF_Q_STATISTICS_DATA
Datentyp:	MQCFGR
Parameter in Gruppe:	<i>QName</i> <i>CreateDate</i> <i>CreateTime</i> <i>QType</i> <i>QDefinitionType</i> <i>QMinDepth</i> <i>QMaxDepth</i> <i>AvgTimeOnQ</i> <i>PutCount</i> <i>PutFailCount</i> <i>Put1Count</i> <i>Put1FailCount</i> <i>PutBytes</i> <i>GetCount</i> <i>GetFailCount</i> <i>GetBytes</i> <i>BrowseCount</i> <i>BrowseFailCount</i> <i>BrowseBytes</i> <i>NonQueuedMsgCount</i> <i>ExpiredMsgCount</i> <i>PurgeCount</i>
Zurückgegeben:	Immer

### **QName**

Beschreibung:	Der Name der Warteschlange.
ID:	MQCA_Q_NAME
Datentyp:	MQCFST
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Immer

### **CreateDate**

Beschreibung:	Das Datum, an dem die Warteschlange erstellt wurde.
---------------	---

ID: MQCA\_CREATION\_DATE  
Datentyp: MQCFST  
Maximale Länge: MQ\_DATE\_LENGTH  
Zurückgegeben: Immer

### **CreateTime**

Beschreibung: Die Zeit, zu der die Warteschlange erstellt wurde.  
ID: MQCA\_CREATION\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Immer

### **QType**

Beschreibung: Der Typ der Warteschlange.  
ID: MQIA\_Q\_TYPE  
Datentyp: MQCFIN  
Wert: MQOT\_LOCAL  
Zurückgegeben: Immer

### **QDefinitionType**

Beschreibung: Der Typ der Warteschlangendefinition.  
ID: MQIA\_DEFINITION\_TYPE  
Datentyp: MQCFIN  
Werte: Mögliche Werte:

- MQQDT\_PREDEFINED
- MQQDT\_PERMANENT\_DYNAMIC
- MQQDT\_TEMPORARY\_DYNAMIC

Zurückgegeben: Wenn verfügbar

### **QMinDepth**

Beschreibung: Die minimale Warteschlangenlänge während des Überwachungszeitraums  
ID: MQIAMO\_Q\_MIN\_DEPTH  
Datentyp: MQCFIN  
In PCF-Gruppe enthalten: *QStatisticsData*  
Zurückgegeben: Wenn verfügbar

### **QMaxDepth**

Beschreibung: Die maximale Warteschlangenlänge während des Überwachungszeitraums  
ID: MQIAMO\_Q\_MAX\_DEPTH  
Datentyp: MQCFIN

In PCF-Gruppe  
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

### ***AvgTimeOnQ***

Beschreibung: Die durchschnittliche Latenzzeit (in Mikrosekunden) von Nachrichten, die während des Überwachungszeitraums aus der Warteschlange destruktiv abgerufen wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64\_AVG\_Q\_TIME

Datentyp: MQCFIL64

In PCF-Gruppe  
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

### ***PutCount***

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die erfolgreich in die Warteschlange gestellt wurden, mit Ausnahme von MQPUT1-Anforderungen. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe [Referenznote 2](#).

ID: MQIAMO\_PUTS

Datentyp: MQCFIL

In PCF-Gruppe  
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

### ***PutFailCount***

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Nachricht in die Warteschlange zu stellen.

ID: MQIAMO\_PUTS\_FAILED

Datentyp: MQCFIN

In PCF-Gruppe  
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

### ***Put1Count***

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Aufrufen erfolgreich in die Warteschlange gestellt wurden. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe [Referenznote 2](#).

ID: MQIAMO\_PUT1S

Datentyp: MQCFIL

In PCF-Gruppe  
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

### **Put1FailCount**

Beschreibung:	Die Anzahl der nicht erfolgreichen Versuche zum Angeben einer Nachricht unter Verwendung von MQPUT1-Aufrufen
ID:	MQIAMO_PUT1S_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

### **PutBytes**

Beschreibung:	Die Anzahl der Byte, die in der Warteschlange in die Warteschlange geschrieben wurden.
ID:	MQIAMO64_PUT_BYTES
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

### **GetCount**

Beschreibung:	Die Anzahl der erfolgreichen zerstörerischen Anforderungen zum Abrufen persistenter und nicht persistenter Nachrichten. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe <a href="#">Referenznote 2</a> .
ID:	MQIAMO_GETS
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

### **GetFailCount**

Beschreibung:	Die Anzahl der nicht erfolgreichen Anforderungen zum Löschen von Zerstör
ID:	MQIAMO_GETS_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

### **GetBytes**

Beschreibung:	Die Anzahl der Byte, die in destruktiven Schreibenanforderungen für persistente und nicht persistente Nachrichten gelesen werden. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe <a href="#">Referenznote 2</a> .
ID:	MQIAMO64_GET_BYTES
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>

Zurückgegeben: Wenn verfügbar

### **BrowseCount**

Beschreibung: Die Anzahl der erfolgreichen nicht-destruktiven Anforderungen zum Abrufen von persistenten und nicht persistenten Nachrichten. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe [Referenznote 2](#).

ID: MQIAMO\_BROWSES

Datentyp: MQCFIL

In PCF-Gruppe  
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

### **BrowseFailCount**

Beschreibung: Die Anzahl der nicht erfolgreichen get-Anforderungen ohne Löschen.

ID: MQIAMO\_BROWSES\_FAILED

Datentyp: MQCFIN

In PCF-Gruppe  
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

### **BrowseBytes**

Beschreibung: Die Anzahl der Byte, die in nicht-destruktiven Leseanforderungen für persistente und nicht persistente Nachrichten gelesen werden. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe [Referenznote 2](#).

ID: MQIAMO64\_BROWSE\_BYTES

Datentyp: MQCFIL64

In PCF-Gruppe  
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

### **NonQueuedMsgCount**

Beschreibung: Die Anzahl der Nachrichten, die die Warteschlange umgangen und direkt in eine wartende Anwendung übertragen wurden.

Das Umgehen einer Warteschlange kann unter bestimmten Umständen nur auftreten. Diese Zahl gibt an, wie oft IBM MQ in der Lage war, die Warteschlange zu umgehen, und nicht die Anzahl der Wartezeiten einer Anwendung.

ID: MQIAMO\_MSGS\_NOT\_QUEUED

Datentyp: MQCFIN

In PCF-Gruppe  
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

### **ExpiredMsgCount**

Beschreibung: Die Anzahl persistenter und nicht persistenter Nachrichten, die gelöscht wurden, weil sie abgelaufen waren, bevor sie abgerufen werden konnten.

ID: MQIAMO\_MSGS\_EXPIRED  
Datentyp: MQCFIN  
In PCF-Gruppe  
enthalten: *QStatisticsData*  
Zurückgegeben: Wenn verfügbar

### **PurgeCount**

Beschreibung: Die Anzahl der bereinigten Nachrichten.  
ID: MQIAMO\_MSGS\_PURGED  
Datentyp: MQCFIN  
In PCF-Gruppe  
enthalten: *QStatisticsData*  
Zurückgegeben: Wenn verfügbar

### **Nachrichtendaten der Kanalstatistik**

Verwenden Sie diese Seite, um die Struktur einer Kanalstatistiknachricht anzuzeigen.

Nachrichtenname:	Kanalstatistiknachricht.
Plattformen:	Alle außer IBM MQ for z/OS.
Systemwarteschlange:	SYSTEM.ADMIN.STATISTICS.QUEUE.

### **QueueManager**

Beschreibung: Der Name des Warteschlangenmanagers.  
ID: MQCA\_Q\_MGR\_NAME.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH.  
Zurückgegeben: Immer.

### **IntervalStartDate**

Beschreibung: Das Datum zu Beginn des Überwachungszeitraums.  
ID: MQCAMO\_START\_DATE.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_DATE\_LENGTH.  
Zurückgegeben: Immer.

### **IntervalStartTime**

Beschreibung: Die Uhrzeit am Anfang des Überwachungszeitraums.  
ID: MQCAMO\_START\_TIME.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_TIME\_LENGTH.  
Zurückgegeben: Immer.

### ***IntervalEndDate***

Beschreibung: Das Datum am Ende des Überwachungszeitraums  
ID: MQCAMO\_END\_DATE.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_DATE\_LENGTH.  
Zurückgegeben: Immer.

### ***IntervalEndTime***

Beschreibung: Die Zeit am Ende des Überwachungszeitraums  
ID: MQCAMO\_END\_TIME.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_TIME\_LENGTH  
Zurückgegeben: Immer.

### ***CommandLevel***

Beschreibung: Die Befehlsebene des Warteschlangenmanagers.  
ID: MQIA\_COMMAND\_LEVEL.  
Datentyp: MQCFIN.  
Zurückgegeben: Immer.

### ***ObjectCount***

Beschreibung: Die Anzahl der Kanalobjekte, auf die in dem Intervall zugegriffen wurde, für das Statistikdaten aufgezeichnet wurden. Dieser Wert wird auf die Anzahl der PCF-Gruppen 'ChlStatisticsData' gesetzt, die in der Nachricht enthalten sind.  
ID: MQIAMO\_OBJECT\_COUNT  
Datentyp: MQCFIN.  
Zurückgegeben: Immer.

### ***ChlStatisticsData***

Beschreibung: Gruppierete Parameter, die Statistikdetails für einen Kanal angeben.  
ID: MQGACF\_CHL\_STATISTICS\_DATA.  
Datentyp: MQCFGR.

Parameter in Gruppe:	<i>ChannelName</i> <i>ChannelType</i> <i>RemoteQmgr</i> <i>ConnectionName</i> <i>MsgCount</i> <i>TotalBytes</i> <i>NetTimeMin</i> <i>NetTimeAvg</i> <i>NetTimeMax</i> <i>ExitTimeMin</i> <i>ExitTimeAvg</i> <i>ExitTimeMax</i> <i>FullBatchCount</i> <i>IncplBatchCount</i> <i>AverageBatchSize</i> <i>PutRetryCount</i>
----------------------	---

Zurückgegeben: Immer.

### **ChannelName**

Beschreibung:	Der Name des Kanals.
ID:	MQCACH_CHANNEL_NAME.
Datentyp:	MQCFST.
Maximale Länge:	MQ_CHANNEL_NAME_LENGTH.
Zurückgegeben:	Immer.

### **ChannelType**

Beschreibung:	Der Kanaltyp.
ID:	MQIACH_CHANNEL_TYPE.
Datentyp:	MQCFIN.
Werte:	Mögliche Werte: <b>MQCHT_SENDER</b> Senderkanal. <b>MQCHT_SERVER</b> Serverkanal. <b>MQCHT_RECEIVER</b> Empfängerkanal. <b>MQCHT_REQUESTER</b> Requesterkanal. <b>MQCHT_CLUSRCVR</b> Clusterempfängerkanal. <b>MQCHT_CLUSSDR</b> Clustersenderkanal.

Zurückgegeben: Immer.

### **RemoteQmgr**

Beschreibung:	Der Name des fernen Warteschlangenmanagers.
---------------	---

ID: MQCA\_REMOTE\_Q\_MGR\_NAME.  
Datentyp: MQCFST.  
Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH  
Zurückgegeben: Wenn verfügbar.

### **ConnectionName**

Beschreibung: Verbindungsname des fernen Warteschlangenmanagers.  
ID: MQCACH\_CONNECTION\_NAME.  
Datentyp: MQCFST  
Maximale Länge: MQ\_CONN\_NAME\_LENGTH  
Zurückgegeben: Wenn verfügbar.

### **MsgCount**

Beschreibung: Die Anzahl der gesendeten oder empfangenen persistenten und nicht persistenten Nachrichten.  
ID: MQIAMO\_MSGS.  
Datentyp: MQCFIN  
Zurückgegeben: Wenn verfügbar.

### **TotalBytes**

Beschreibung: Die Anzahl der Byte, die für persistente und nicht persistente Nachrichten gesendet oder empfangen wurden.  
ID: MQIAMO64\_BYTES.  
Datentyp: MQCFIN64.  
Zurückgegeben: Wenn verfügbar.

### **NetTimeMin**

Beschreibung: Die kürzeste aufgezeichnete Kanalrundfahrt, die im Aufzeichnungsintervall gemessen wurde, in Mikrosekunden.  
ID: MQIAMO\_NET\_TIME\_MIN.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### **NetTimeAvg**

Beschreibung: Die im Aufzeichnungsintervall gemessene durchschnittliche Rundungsrundfahrt in Mikrosekunden (in Mikrosekunden).  
ID: MQIAMO\_NET\_TIME\_AVG.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

### **NetTimeMax**

Beschreibung: Die längste aufgezeichnete Kanalrundfahrt, die im Aufzeichnungsintervall gemessen wurde, in Mikrosekunden.

ID: MQIAMO\_NET\_TIME\_MAX.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

#### ***ExitTimeMin***

Beschreibung: Die kürzeste aufgezeichnete Zeit in Mikrosekunden, die für die Ausführung eines Benutzerexits im Aufzeichnungsintervall aufgewendet wurde.  
ID: MQIAMO\_EXIT\_TIME\_MIN.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

#### ***ExitTimeAvg***

Beschreibung: Die durchschnittliche aufgezeichnete Zeit in Mikrosekunden, die für die Ausführung eines Benutzerexits im Aufzeichnungsintervall aufgewendet wurde. Gemessen in Mikrosekunden.  
ID: MQIAMO\_EXIT\_TIME\_AVG.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

#### ***ExitTimeMax***

Beschreibung: Die längste aufgezeichnete Zeit (in Mikrosekunden), die für die Ausführung eines Benutzerexits im Aufzeichnungsintervall verwendet wurde. Gemessen in Mikrosekunden.  
ID: MQIAMO\_EXIT\_TIME\_MAX.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

#### ***FullBatchCount***

Beschreibung: Die Anzahl der vom Kanal verarbeiteten Stapel, die gesendet wurden, weil der Wert der Kanalattribute BATCHSZ oder BATCLIM erreicht wurde.  
ID: MQIAMO\_FULL\_BATCHES.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

#### ***IncomplBatchCount***

Beschreibung: Die Anzahl der vom Kanal verarbeiteten Batches, die gesendet wurden, ohne dass der Wert des Kanalattributs BATCHSZ erreicht wurde.  
ID: MQIAMO\_INCOMPLETE\_BATCHES.  
Datentyp: MQCFIN.  
Zurückgegeben: Wenn verfügbar.

#### ***AverageBatchSize***

Beschreibung: Die durchschnittliche Stapelgröße von Stapelverarbeitungen, die vom Kanal verarbeitet werden.

ID: MQIAMO\_AVG\_BATCH\_SIZE.  
 Datentyp: MQCFIN.  
 Zurückgegeben: Wenn verfügbar.

**PutRetryCount**

Beschreibung: Die Anzahl der Male in dem Zeitintervall, in dem eine Nachricht nicht eingereist ist, und eine Wiederholungsschleife eingegeben wurde.  
 ID: MQIAMO\_PUT\_RETRIES.  
 Datentyp: MQCFIN.  
 Zurückgegeben: Wenn verfügbar.

**Referenzinformationen**

Verwenden Sie diese Seite, um die Anmerkungen anzuzeigen, auf die sich die Beschreibungen der Struktur von Abrechnungs- und Statistiknachrichten beziehen.

Die folgenden Nachrichtendatenbeschreibungen beziehen sich auf diese Hinweise:

- „MQI-Abrechnungsnachrichtendaten“ auf Seite 161
- „Nachrichtendaten der Warteschlangenabrechnung“ auf Seite 173
- „MQI-Statistikdaten, Nachrichtendaten“ auf Seite 184
- „Nachrichtendaten der Warteschlangenstatistik“ auf Seite 195
- „Nachrichtendaten der Kanalstatistik“ auf Seite 202

1. Dieser Parameter bezieht sich auf IBM MQ-Objekte. Dieser Parameter ist ein Array von Werten (MQCFIL oder MQCFIL64), die durch die folgenden Konstanten indiziert werden:

*Tabelle 24. Feldgruppe nach Objekttyp indiziert*

Objekttyp	Wertkontext
MQOT_Q (1)	Enthält den Wert, der sich auf Warteschlangenobjekte bezieht.
MQOT_NAMELIST (2)	Enthält den Wert, der sich auf namelistische Objekte bezieht.
MQOT_PROCESS (3)	Enthält den Wert, der sich auf Prozessobjekte bezieht.
MQOT_Q_MGR (5)	Enthält den Wert, der sich auf WS-Manager-Objekte bezieht.
MQOT_CHANNEL (6)	Enthält den Wert, der sich auf Kanalobjekte bezieht.
MQOT_AUTH_INFO (7)	Enthält den Wert, der sich auf Authentifizierungsobjekte bezieht.
MQOT_TOPIC (8)	Enthält den Wert, der sich auf Themenobjekte bezieht.

**Anmerkung:** Es werden ein Array mit 13 MQCFIL- oder MQCFIL64-Werten zurückgegeben, aber nur die aufgeführten Werte sind aussagekräftig.

2. Dieser Parameter bezieht sich auf IBM MQ-Nachrichten. Dieser Parameter ist ein Array von Werten (MQCFIL oder MQCFIL64), die durch die folgenden Konstanten indiziert werden:

Tabelle 25. Array indexiert durch Persistenzwert	
Konstant	Wert
1	Enthält den Wert für nicht persistente Nachrichten.
2	Enthält den Wert für persistente Nachrichten.

**Anmerkung:** Der Index für jedes dieser Arrays beginnt bei Null, so dass sich ein Index von 1 auf die zweite Zeile des Arrays bezieht. Elemente dieser Arrays, die nicht in diesen Tabellen aufgeführt sind, enthalten keine Abrechnungs- oder Statistikdaten.

## Anwendungsaktivitätstrace

Der Aktivitätstrace der Anwendung erzeugt detaillierte Informationen zum Verhalten von Anwendungen, die mit einem Warteschlangenmanager verbunden sind. Dabei wird das Verhalten einer Anwendung verfolgt und es wird eine detaillierte Ansicht der Parameter bereitgestellt, die von einer Anwendung während der Interaktion mit IBM MQ verwendet werden. Es zeigt auch die Reihenfolge der MQI-Aufrufe, die von einer Anwendung ausgegeben werden.

Verwenden Sie den Anwendungsaktivitätstrace, wenn Sie mehr Informationen benötigen, als von Ereignisüberwachung, Nachrichtenüberwachung, Accounting- und Statistiknachrichten und Real-time Monitoring bereitgestellt werden.

**Anmerkung:** Der Aktivitätstrace wird in der IBM MQ -Verbindung für jede Anwendung generiert. Wenn also Abrechnungsnachrichten aktiviert sind, zählen die Aktivitätstraceoperationen zu den MQI-Abrechnungsdaten für jede Anwendung.

IBM MQ unterstützt zwei Methoden zur Erfassung von Tracedaten zur Anwendungsaktivität.

- Zentrale Erfassung von Trace-Informationen der Anwendungsaktivität, wobei der Trace für Anwendungsaktivitäten erfasst und gelesen wird, indem die PCF-Nachrichten des Aktivitätentrace in die Systemwarteschlange SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE geschrieben werden.
- Subskription von Aktivitätstracedaten, die in bestimmte IBM MQ-Systemthemen geschrieben werden.

Beachten Sie, dass der Aktivitätstrace von IBM MQ for z/OS nicht unterstützt wird.

 Sie können jetzt Anwendungsnamen in den meisten Programmiersprachen angeben, die von IBM MQ unterstützt werden, und finden unter [Anwendungsname in unterstützten Programmiersprachen angeben](#) weitere Informationen hierzu.

## Zentrale Erfassung von Trace-Informationen für Anwendungsaktivitäten konfigurieren

Eine Tracenachricht der Anwendungsaktivität ist eine PCF-Nachricht. Sie können den Aktivitätstrace mit Hilfe einer Konfigurationsdatei konfigurieren. Wenn Sie die zentrale Erfassung von Trace-Informationen für die Anwendungsaktivität konfigurieren möchten, setzen Sie das WS-Manager-Attribut ACTVTRC. Sie können diese Einstellung auf Verbindungsebene mit Hilfe von MQCONNX-Optionen oder unter Verwendung der Aktivitätstracekonfigurationsdatei auf Anwendungszeilengruppe überschreiben.

### Informationen zu diesem Vorgang

Aktivitätstracenachrichten bestehen aus einer MQMD-Struktur: einer PCF-Headerstruktur (MQCFH), gefolgt von einer Reihe von PCF-Parametern. Eine Sequenz der PCF-Gruppen der Anwendung "ApplicationTraceData" folgt den PCF-Parametern. Diese PCF-Gruppen erfassen Informationen zu den MQI-Operationen, die eine Anwendung ausführt, während eine Verbindung zu einem Warteschlangenmanager besteht. Zum Konfigurieren des Aktivitätstrace verwenden Sie eine Konfigurationsdatei namens mqat.ini.

Um zu steuern, ob die Trace-Informationen der Anwendungsaktivität erfasst werden, konfigurieren Sie eine oder mehrere der folgenden Einstellungen:

1. Das WS-Manager-Attribut ACTVTRC.
2. Die ACTVCONO-Einstellungen (in der MQCNO-Struktur, die in MQCONNX übergeben wurde).
3. Die übereinstimmende Zeilengruppe für die Anwendung in der Aktivitätstracekonfigurationsdatei `mqat.ini`.

Die vorherige Sequenz ist signifikant. Das Attribut ACTVTRC wird durch die Einstellungen für ACTVCONO außer Kraft gesetzt, die durch die Einstellungen in der Datei `mqat.ini` außer Kraft gesetzt werden.

Traceeinträge werden nach Abschluss jeder Operation geschrieben, sofern nicht anders angegeben. Diese Einträge werden zuerst in die Systemwarteschlange `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE` geschrieben und dann in die Anwendungsaktivitätstracenachrichten geschrieben, wenn die Anwendung die Verbindung zum Warteschlangenmanager trennt. Bei Anwendungen mit langer Laufzeit werden temporäre Nachrichten geschrieben, wenn eines der folgenden Ereignisse eintritt:

- Die Lebensdauer der Verbindung erreicht einen definierten Zeitlimitwert.
- Die Anzahl der Operationen erreicht eine angegebene Zahl.
- Die Menge der im Speicher erfassten Daten erreicht die maximale Nachrichtenlänge, die für die Warteschlange zulässig ist.

Sie legen den Zeitlimitwert mit dem Parameter **ActivityInterval** fest. Die Anzahl der Operationen wird mit dem Parameter **ActivityCount** festgelegt. Beide Parameter werden in der Aktivitätstracekonfigurationsdatei `mqat.ini` angegeben.

Das Aktivieren des Trace der Anwendungsaktivität kann sich auf die Leistung auswirken. Der Systemaufwand kann reduziert werden, indem der **ActivityCount** und die **ActivityInterval**-Einstellungen optimiert werden. Weitere Informationen finden Sie im Abschnitt [„Leistungswirkung des Trace der Anwendungsaktivität optimieren“](#) auf Seite 217.

Die einfachste Methode zur Anzeige der Inhalte von Tracenachrichten zur Anwendungsaktivität ist, das [„Beispielprogramm 'amqsact'“](#) auf Seite 218 zu verwenden.

## Vorgehensweise

1. [„Festlegen von ACTVTRC zum Steuern der Erfassung von Aktivitätstraceinformationen“](#) auf Seite 209.
2. [„Festlegen von MQCONNX-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen“](#) auf Seite 210.
3. [„Aktivitätstraceverhalten mit mqat.ini konfigurieren“](#) auf Seite 211.
4. [„Leistungswirkung des Trace der Anwendungsaktivität optimieren“](#) auf Seite 217.

## ***Festlegen von ACTVTRC zum Steuern der Erfassung von Aktivitätstraceinformationen***

Verwenden Sie das WS-Managerattribut ACTVTRC, um die Erfassung von Trace-Informationen für MQI-Anwendungsaktivitäten zu steuern.

## Informationen zu diesem Vorgang

Tracenachrichten für Anwendungsaktivitäten werden nur für Verbindungen generiert, die nach der Aktivierung der Anwendungsaktivitätstrace gestartet werden. Der Parameter **ACTVTRC** kann die folgenden Werte haben:

### **ON**

Die API-Aktivitätstracensammlung ist

### **OFF**

Die API-Aktivitätstracesammlung ist

**Anmerkung:** Die Einstellung **ACTVTRC** kann durch den Parameter **ACTVCONO** des Warteschlangenmanagers überschrieben werden. Wenn Sie den Parameter **ACTVCONO** auf `ENABLED` setzen, kann die Einstellung **ACTVTRC** für eine bestimmte Verbindung mit dem Feld **Options** in der MQCNO -Struktur außer Kraft gesetzt werden. Weitere Informationen finden Sie im Abschnitt [„Festlegen von MQCONNX-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen“](#) auf Seite 210.

## Beispiel

Wenn Sie den Wert des Parameters **ACTVTRC** ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR. Verwenden Sie beispielsweise den folgenden MQSC-Befehl, um die Erfassung von Trace-Informationen für MQI-Anwendungsaktivitäten zu aktivieren:

```
ALTER QMGR ACTVTRC(ON)
```

## Nächste Schritte

Die einfachste Methode zur Anzeige der Inhalte von Tracenachrichten zur Anwendungsaktivität ist, das „Beispielprogramm 'amqsact'“ auf Seite 218 zu verwenden.

Das Aktivieren des Trace der Anwendungsaktivität kann sich auf die Leistung auswirken. Der Systemaufwand kann reduziert werden, indem der **ActivityCount** und die **ActivityInterval**-Einstellungen optimiert werden. Weitere Informationen finden Sie im Abschnitt „Leistungswirkung des Trace der Anwendungsaktivität optimieren“ auf Seite 217.

## ***Festlegen von MQCONNX-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen***

Wenn das WS-Managerattribut **ACTVCONO** auf ENABLED gesetzt ist, können Sie den Parameter **ConnectOpts** im Aufruf MQCONNX verwenden, um die Anwendungsaktivitätsberichte auf einer Basis zu aktivieren oder zu inaktivieren. Diese Optionen überschreiben das Aktivitätstraceverhalten, das durch das Warteschlangenmanagerattribut **ACTVTRC** definiert ist, und können durch Einstellungen in der Aktivitätstracekonfigurationsdatei mqat.ini außer Kraft gesetzt werden.

## Vorgehensweise

1. Setzen Sie das WS-Managerattribut **ACTVCONO** auf ENABLED .

**Anmerkung:** Wenn eine Anwendung versucht, das Abrechnungsverhalten einer Anwendung mit dem Parameter **ConnectOpts** zu ändern, und das QMGR-Attribut **ACTVCONO** auf DISABLED gesetzt ist, wird kein Fehler an die Anwendung zurückgegeben, und die Aktivitätstraceerfassung wird durch die Warteschlangenmanagerattribute oder die Aktivitätstracekonfigurationsdatei mqat.ini definiert.

2. Setzen Sie den Parameter **ConnectOpts** auf dem MQCONNX-Aufruf auf MQCNO\_ACTIVITY\_TRACE\_ENABLED .

Der Parameter **ConnectOpts** im MQCONNX-Aufruf kann die folgenden Werte haben:

### **MQCNO\_ACTIVITY\_TRACE\_DISABLED**

Der Aktivitätstrace ist für die Verbindung inaktiviert.

### **MQCNO\_ACTIVITY\_TRACE\_ENABLED**

Der Aktivitätstrace ist für die Verbindung aktiviert.

**Anmerkung:** Wenn eine Anwendung sowohl MQCNO\_ACTIVITY\_TRACE\_ENABLED als auch MQCNO\_ACTIVITY\_TRACE\_DISABLED für MQCONNX auswählt, schlägt der Aufruf mit einem Ursacheencode von MQRC\_OPTIONS\_ERROR fehl.

3. Stellen Sie sicher, dass diese Aktivitätstraceeinstellungen nicht durch Einstellungen in der Aktivitätstracekonfigurationsdatei mqat.ini außer Kraft gesetzt werden.

Weitere Informationen finden Sie im Abschnitt „Aktivitätstraceverhalten mit mqat.ini konfigurieren“ auf Seite 211.

## Nächste Schritte

Die einfachste Methode zur Anzeige der Inhalte von Tracenachrichten zur Anwendungsaktivität ist, das „Beispielprogramm 'amqsact'“ auf Seite 218 zu verwenden.

Das Aktivieren des Trace der Anwendungsaktivität kann sich auf die Leistung auswirken. Der Systemaufwand kann reduziert werden, indem der **ActivityCount** und die **ActivityInterval**-Einstellungen

optimiert werden. Weitere Informationen finden Sie im Abschnitt „Leistungswirkung des Trace der Anwendungsaktivität optimieren“ auf Seite 217.

### **Aktivitätstraceverhalten mit mqat.ini konfigurieren**

Das Verhalten des Aktivitätstrace wird mit Hilfe einer Konfigurationsdatei namens `mqat.ini` konfiguriert. Diese Datei wird verwendet, um die Stufe und Häufigkeit der Tracedaten für die Berichtsaktivität zu definieren. Die Datei bietet außerdem die Möglichkeit, Regeln zu definieren, mit deren Hilfe der Aktivitätstrace auf der Basis des Namens einer Anwendung aktiviert und inaktiviert werden kann.

### **Informationen zu diesem Vorgang**

**Linux** **UNIX** Auf UNIX and Linux-Systemen befindet sich die Datei `mqat.ini` im Verzeichnis mit den Warteschlangenmanagerdaten, das die gleiche Position wie die Datei `qm.ini` hat.

**Windows** Auf Windows-Systemen befindet sich die Datei `mqat.ini` im Verzeichnis `C:\Program Files\IBM\WebSphere MQ\mqgrs\queue_manager_name` mit den Warteschlangenmanagerdaten. Benutzer, die Anwendungen ausführen, für die ein Trace durchgeführt wird, benötigen die Berechtigung zum Lesen dieser Datei

**Anmerkung:** Für Warteschlangenmanager, die ausgehend von IBM WebSphere MQ 7.1 oder früher migriert wurden, fehlt die Datei `mqat.ini`. In solchen Fällen muss die Datei `mqat.ini` manuell erstellt werden und es müssen Berechtigungen des Typs 660 für die Datei festgelegt werden.

Beim Ändern der Datei `mqat.ini` werden neu erstellte IBM MQ-Verbindungen entsprechend der geänderten Version verarbeitet. Vorhandene Verbindungen verwenden weiterhin die vorherige Version, es sei denn, die Parameter des Warteschlangenmanagers werden geändert, z. B. nach einem Befehl [ALTER QMGR](#).

Diese Datei befolgt dasselbe Format für Zeilengruppenschlüssel und Parameter/Wert-Paare wie die Dateien `mq5.ini` und `qm.ini`.

Die Datei besteht aus einer einzelnen Zeilengruppe, **AllActivityTrace**, um die Stufe und Häufigkeit der Trace-Daten der Berichtsaktivitäten standardmäßig für alle Aktivitätstrace zu konfigurieren.

Die Datei kann auch mehrere Zeilengruppen **ApplicationTrace** enthalten. Jede dieser Eigenschaften definiert eine Regel für das Traceverhalten für eine oder mehrere Verbindungen, die auf der Übereinstimmung des Anwendungsnamens der Verbindungen zur Regel basiert.

#### **Zeilengruppe 'AllActivityTrace'**

Eine einzelne Zeilengruppe **AllActivityTrace** definiert Einstellungen für den Aktivitätstrace, der für alle IBM MQ-Verbindungen angewendet wird, bis er überschrieben wird.

Einzelne Werte in der Zeilengruppe **AllActivityTrace** können durch spezifischere Informationen in einer Zeilengruppe **ApplicationTrace stanza** außer Kraft gesetzt werden.

Wenn mehr als eine **AllActivityTrace**-Zeilengruppe angegeben wird, werden die Werte in der letzten Zeilengruppe verwendet. Für die in der ausgewählten **AllActivityTrace** angegebenen Parameter werden Standardwerte verwendet. Parameter und Werte aus früheren Zeilengruppen **AllActivityTrace** werden ignoriert.

Die folgenden Parameter können unter der Zeilengruppe **AllActivityTrace** angegeben werden:

Tabelle 26. Parameter/Wert-Paare, die in der Aktivitätstracekonfigurations-Konfigurationsdatei verwendet werden können

Ihren Namen	Werte (Standardwert in Fettdruck)	Beschreibung
ActivityInterval	0-99999999 ( <b>1</b> )	Ungefächertes Zeitintervall in Sekunden zwischen Tracenachrichten. Alle Aktivitäten, die von einer Verbindung in diesem Intervall ausgeführt werden, werden in einer einzigen Nachricht geschrieben. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung disconnects (oder wenn die Anzahl der Aktivitäten erreicht ist).
ActivityCount	0-99999999 ( <b>100</b> )	Die Anzahl der MQI-oder XA-Operationen zwischen Tracenachrichten. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung die Verbindung trennt (oder wenn das Aktivitätsintervall abgelaufen ist).
TraceLevel	LOW/ <b>MEDIUM</b> /HIGH	Die Menge der Parameterdetails, für die für jede Operation ein Trace durchgeführt wurde. In der Beschreibung der einzelnen Operationen wird angegeben, welche Parameter für die einzelnen Trace-Ebenen angegeben werden.
TraceMessageData	<b>0</b> -104 857 600 (maximal 100 MB)	Die Menge der Nachrichtendaten, die in Byte für MQGET-, MQPUT-, MQPUT1- und Callback-Operationen aufgezeichnet wurden.
StopOnGetTraceMsg	<b>EIN</b> /AUS	Die Verwendung von Aktivitätstrace, Trace-Anwendungen, die auch Aktivitäts-Trace-Nachrichten verarbeiten, ist aufgrund einer möglichen Schleife nicht ratsam.
SubscriptionDelivery	<b>BATCHED</b> /SOFORT	Legt fest, ob die Parameter "ActivityInterval" und "ActivityCount" verwendet werden sollen, wenn eine oder mehrere Aktivitätstrace-Subskriptionen vorhanden sind. Wenn Sie diesen Parameter auf IMMEDIATE setzen, werden die Werte für "ActivityInterval" und "ActivityCount" mit den tatsächlichen Werten von 1 überschrieben, wenn die Trace-Daten eine übereinstimmende Subskription aufweisen. Jeder Aktivitätstracesatz wird nicht mit anderen Datensätzen aus derselben Verbindung ausgeliefert und stattdessen sofort an die Subskription gesendet, ohne dass es zu einer Verzögerung kommt.

### Zeilengruppe 'ApplicationTrace'

Die Zeilengruppe 'ApplicationTrace' enthält eine Regel, mit der definiert wird, für welche IBM MQ-Verbindungen auf Basis des Anwendungsnamens ein Trace durchgeführt wird. Optional das Standardverhalten, das unter den Einstellungen "Allsettings" definiert ist, die die globale Tracestufe und die Häufigkeitseinstellungen außer Kraft setzen

Diese Zeilengruppe kann die Parameter ApplName, ApplFunction und ApplClass enthalten, die entsprechend den übereinstimmenden Regeln verwendet werden, die in Verbindungsübereinstimmungsregeln definiert sind, um zu bestimmen, ob die Zeilengruppe für eine bestimmte Verbindung gilt.

Die Zeilengruppe muss den Parameter 'Trace' enthalten, um festzustellen, ob diese Regel den Aktivitätstrace für übereinstimmende Verbindungen aktiviert oder inaktiviert.

Eine off-Regel kann verwendet werden, um den Trace für spezifischere Anwendungsnamen explizit zu inaktivieren und die ACTVTRC-Einstellung des Warteschlangenmanagers oder der Aktivitätstrace-Verbindungsoptionen außer Kraft zu setzen.

Die folgenden Parameter können in der Zeilengruppe **ApplicationTrace** angegeben werden:

<i>Tabelle 27. Parameter/Wert-Paare, die in der Anwendungstracekonfigurations-Konfigurationsdatei verwendet werden können</i>		
<b>Ihren Namen</b>	<b>Werte (Standardwert in Fett-druck)</b>	<b>Beschreibung</b>
Trace	ON/OFF (Erforderlicher Parameter-kein Standardwert)	Aktivitätstrace-Switch. Dieser Schalter kann in der anwendungsspezifischen Zeilengruppe verwendet werden, um festzustellen, ob der Aktivitätstrace für den Geltungsbereich der aktuellen Anwendungszeilengruppe aktiv ist. Beachten Sie, dass dieser Wert die Einstellungen ACTVTRC und ACTVCONO für den WS-Manager außer Funktion setzt.
ApplName	Zeichenfolge (Erforderlicher Parameter-kein Standardwert)	<p>Dieser Wert wird verwendet, um festzustellen, auf welche Anwendungen die Zeilengruppe 'ApplicationTrace' angewendet wird. Sie wird mit dem Wert ApplName aus der API-Exit-Kontextstruktur (die äquivalent zu MQMD.PutApplName) ist, abgeglichen. Der Inhalt des ApplName-Werts variiert abhängig von der Anwendungsumgebung.</p> <p>Auf allen Plattformen außer z/OS wird nur der Teil von MQAXC.ApplName, der den Dateinamen enthält, mit dem Wert in der Zeilengruppe abgeglichen. Zeichen links vom Pfadtrennzeichen rechts werden ignoriert, wenn der Vergleich durchgeführt wird.</p> <p><b>z/OS</b> Bei z/OS-Anwendungen wird der gesamte Wert von MQAXC.ApplName mit dem Wert in der Zeilengruppe abgeglichen.</p> <p>Ein einzelnes Platzhalterzeichen (*) kann am Ende des ApplName-Werts verwendet werden, um eine beliebige Anzahl von Zeichen nach diesem Punkt abzugleichen. Wenn der Wert für "ApplName" auf ein einzelnes Platzhalterzeichen (*) gesetzt wird, entspricht der Wert für "ApplName" allen Anwendungen.</p>

Tabelle 27. Parameter/Wert-Paare, die in der Anwendungstracekonfigurations-Konfigurationsdatei verwendet werden können (Forts.)

Ihren Namen	Werte (Standardwert in Fett- druck)	Beschreibung
 Appl- Function	Zeichenfolge (Standardwert *)	Dieser Wert wird verwendet, um zu qualifizieren, auf welche Anwendungsprogramme die Zeilengruppe <b>ApplicationTrace</b> und der Wert <b>AppName</b> angewendet werden.  Die Zeilengruppe ist optional, ist aber nur für IBM i-Warteschlangenmanager gültig. Ein einzelnes Platzhalterzeichen (*) kann am Ende des <b>AppName</b> -Werts verwendet werden, um eine beliebige Anzahl von Zeichen abzugleichen.  Eine Zeilengruppe <b>ApplicationTrace</b> , die <b>AppName = *</b> und <b>AppFunction = AMQSPUTO</b> angibt, gilt beispielsweise für alle Invocations des AMQSPUTO-Programms von einem beliebigen Job aus.
AppClass	USER / MCA / <b>ALL</b>	Die Klasse der Anwendung. In der folgenden Tabelle finden Sie eine Erläuterung, wie die <b>AppType</b> -Werte den IBM MQ-Verbindungen entsprechen.

In der folgenden Tabelle ist dargestellt, wie die *AppClass* -Werte den Feldern **APICallerType** und **APIEnvironment** in der Kontext-Struktur der Verbindungs-API entsprechen.

Tabelle 28. Appclass-Werte und ihre Übereinstimmung mit den Feldern APICallerType und APIEnvironment

APPLCLASS	API-Aufruertyp:	API-Umgebung:	Beschreibung
BENUTZER	MQXACT_EXTERNAL	MQXE_OTHER	Es wird nur ein Trace für Benutzeranwendungen
Nachrichtenkanal-agent	(Beliebiger Wert)	MQXE_MCA MQXE_MCA_CLNTCONN MQXE_MCA_SVRCONN	Clients und Kanäle (amqrmppa)
ALLE	(Beliebiger Wert)	(Beliebiger Wert)	Trace für alle Verbindungen



**Achtung:** Sie müssen eine **APPLCLASS** von **MCA** für Clientbenutzeranwendungen verwenden, da eine Klasse von **USER** nicht mit diesen übereinstimmt.

Sie können beispielsweise den folgenden Code verwenden, um einen Trace für die Beispielanwendung **amqspu**tc zu erstellen:

```

ApplicationTrace:
AppClass=MCA                                # Application type
                                             # Values: (USER | MCA | INTERNAL | ALL)
                                             # Default: USER
AppName=amqspu      # Application name (may be wildcard)
                                             # (matched to app name without path)
                                             # Default: *
Trace=ON                                           # Activity trace switch for application
                                             # Values: ( ON | OFF )
                                             # Default: OFF
ActivityInterval=30                               # Time interval between trace messages
                                             # Values: 0-99999999 (0=off)
                                             # Default: 0
ActivityCount=1                                   # Number of operations between trace msgs
    
```

```

TraceLevel=MEDIUM
TraceMessageData=1000
# Values: 0-99999999 (0=off)
# Default: 0
# Amount of data traced for each operation
# Values: LOW | MEDIUM | HIGH
# Default: MEDIUM
# Amount of message data traced
# Values: 0-100000000
# Default: 0

```

Der Standardwert `mqat.ini`, der bei der Erstellung eines Warteschlangenmanagers generiert wird, enthält eine einzige Regel zum expliziten Inaktivieren des Aktivitätstrace für das bereitgestellte Aktivitätstrace-Beispiel `amqsact`.

## Regeln für die Übereinstimmungen von Verbindungen

Der Warteschlangenmanager wendet die folgenden Regeln an, um festzustellen, welche Zeilengruppen für eine Verbindung verwendet werden sollen.

1. Ein in der Zeilengruppe **AllActivityTrace** angegebener Wert wird für die Verbindung verwendet, es sei denn, der Wert tritt auch in einer Zeilengruppe **ApplicationTrace** auf und die Zeilengruppe erfüllt die Übereinstimmungskriterien für die Verbindung, die in den Punkten [2](#), [3](#) und [4](#) beschrieben wird.
2. Der Wert von **App1Class** wird mit dem Typ der IBM MQ-Verbindung abgeglichen. Wenn der **App1Class** nicht mit dem Verbindungstyp übereinstimmt, wird die Zeilengruppe für diese Verbindung ignoriert.
3. Der **App1Name** -Wert in der Zeilengruppe wird mit dem Dateinamensabschnitt des Feldes **App1Name** von der API-Exit-Kontextstruktur (MQAXC) für die Verbindung abgeglichen.

Der Dateinamensabschnitt wird von den Zeichen rechts vom letzten Pfadtrennzeichen (/oder \) abgeleitet. Wenn die Zeilengruppe **App1Name** ein Platzhalterzeichen (\*) enthält, werden nur die Zeichen links vom Platzhalterzeichen mit der äquivalenten Anzahl von Zeichen aus der **App1Name** der Verbindung verglichen.

Wenn z. B. der Zeilengruppenwert "FRE\*" angegeben wird, werden nur die ersten drei Zeichen im Vergleich verwendet, so dass "path/FREEDOM" und "path\FREDDY" übereinstimmen, aber "path/FRIEND" nicht. Wenn der **App1Name** -Wert der Zeilengruppe nicht mit der Verbindung **App1Name** übereinstimmt, wird die Zeilengruppe für diese Verbindung ignoriert.

4. Wenn mehr als eine Zeilengruppe mit dem **App1Name** und dem **App1Class** der Verbindung übereinstimmt, wird die Zeilengruppe mit dem spezifischsten **App1Name** verwendet.

Der spezifischste **App1Name** ist als der definiert, der die meisten Zeichen verwendet, um den **App1Name** der Verbindung abzugleichen.

Beispiel: Wenn die Datei `ini` eine Zeilengruppe mit **App1Name** = "FRE\*" und eine andere Zeilengruppe mit **App1Name** = "FREE\*" enthält, wird die Zeilengruppe mit **App1Name** = "FREE\*" als beste Übereinstimmung für eine Verbindung mit **App1Name** = "path/FREEDOM" ausgewählt, da sie vier Zeichen enthält (in der Erwägung, dass **App1Name** = "FRE\*" stimmt mit nur drei überein).

5. Wenn nach dem Anwenden der Regeln in den Punkten [2](#), [3](#) und [4](#) mehr als eine Zeilengruppe mit den Verbindungen **App1Name** und **App1Class** der Verbindung übereinstimmt, werden die Werte aus dem letzten Abgleich verwendet und alle anderen Zeilengruppen werden ignoriert.

## Standardeinstellungen für jede Regel überschreiben

Optional können die globalen Tracestufe und die Häufigkeitseinstellungen unter der Zeilengruppe **AllActivityTrace** für die Verbindungen überschrieben werden, die mit der Zeilengruppe **ApplicationTrace** übereinstimmen.

Die folgenden Parameter können unter der Zeilengruppe **ApplicationTrace** festgelegt werden. Wenn sie nicht festgelegt sind, wird der Wert von den Zeilengruppeneinstellungen **AllActivityTrace** übernommen:

- **ActivityInterval**
- **ActivityCount**
- **TraceLevel**
- **TraceMessageData**

- **StopOnTraceMsg**

### Syntax von mqat.ini

Für das Format der Datei mqat.ini gelten die folgenden Syntaxregeln:

- Text, der mit einem Hash- oder Semikolon beginnt, wird als Kommentar betrachtet, der sich bis zum Ende der Zeile erstreckt.
- Die erste signifikante Zeile (ohne Kommentar) muss ein Zeilengruppenschlüssel sein.
- Ein Zeilengruppenschlüssel setzt sich aus dem Namen der Zeilengruppe gefolgt von einem Doppelpunkt zusammen.
- Ein Parameterwert-Paar besteht aus dem Namen eines Parameters, gefolgt von einem Gleichheitszeichen und dann dem Wert.
- Es kann nur ein Paar aus einem Parameterwert in einer Zeile stehen. (Ein Parameter-Wert darf nicht in eine andere Zeile umgebrochen werden.)
- Führende und abschließende Leerzeichen werden ignoriert. Es gibt keine Begrenzung für den Leerraum zwischen Zeilengruppenamen, Parameternamen und -werten oder Parameter/Wert-Paaren. Zeilenumbrüche sind bedeutsam und werden nicht ignoriert.
- Die maximale Länge für eine Zeile beträgt 2048 Zeichen.
- Bei den Zeilengruppenschlüsseln, Parameternamen und konstanten Parameterwerten wird die Groß-/Kleinschreibung nicht beachtet, die Variablenparameterwerte (*ApplName* und *DebugPath*) werden jedoch von der Groß-/Kleinschreibung abhängig gemacht.

### Beispiel für Anwendungsaktivität-Trace-Datei

Das folgende Beispiel zeigt, wie die Konfigurationsdaten in der INI-Datei des Aktivitätstrace angegeben sind.

```
AllActivityTrace:
ActivityInterval=1
ActivityCount=100
TraceLevel=MEDIUM
TraceMessageData=0
StopOnGetTraceMsg=ON

ApplicationTrace:
ApplName=amqs*
Trace=ON
TraceLevel=HIGH
TraceMessageData=1000

ApplicationTrace:
ApplName=amqsact*
Trace=OFF
```

Die oben genannte Zeilengruppe **AllActivityTrace** definiert, wie der Aktivitätstrace standardmäßig ausgeführt wird, wenn er entweder über ApplicationTrace-Regeln oder durch das Attribut ACTVTRC des Warteschlangenmanagers oder über das Programm durch eine Anwendung aktiviert wird.

Die erste Zeilengruppe **ApplicationTrace** definiert eine Regel, die zu einer beliebigen MQI-Aktivität durch eine Anwendung führt, deren Name mit "amqs" beginnt, für die ein Trace durchgeführt wird. Der für diese Anwendungen generierte Trace hat einen hohen Detailausschnitt und enthält bis zu 1000 Byte an Nachrichtendaten. Das Aktivitätsintervall und die Anzahl der Parameter werden übernommen.

Die zweite Zeilengruppe **ApplicationTrace** definiert eine Regel, die den Trace für Anwendungen inaktiviert, die den Namen "amqsact" (die Aktivitätstrace-Stichprobe) starten. Diese Regel setzt die frühere 'on'-Regel für die Anwendung 'amqsact' außer Kraft, woraus sich kein Trace für diese Anwendung ergibt.

Ein Beispiel wird auch unter dem Namen mqat.ini im Verzeichnis mit den Beispielen für die Programmiersprache C bereitgestellt (dies ist dasselbe Verzeichnis, in dem sich auch die Datei 'amqsact.c' befindet). Diese Datei kann für Warteschlangenmanager, die aus einem früheren Release von IBM MQ migriert wurden, in das Verzeichnis mit den Warteschlangenmanagerdaten kopiert werden.

## Nächste Schritte

Das Aktivieren des Trace der Anwendungsaktivität kann sich auf die Leistung auswirken. Der Systemaufwand kann reduziert werden, indem der **ActivityCount** und die **ActivityInterval**-Einstellungen optimiert werden. Weitere Informationen finden Sie im Abschnitt „[Leistungswirkung des Trace der Anwendungsaktivität optimieren](#)“ auf Seite 217.

### **Leistungswirkung des Trace der Anwendungsaktivität optimieren**

Die Aktivierung der Anwendungsaktivitätstrace kann zu einer Leistungseinbuße führen. Dies kann dadurch reduziert werden, dass Traces nur für benötigte Anwendungen erstellt werden, dass die Anzahl der Anwendungen, die Daten aus der Warteschlange entnehmen, erhöht wird und dass eine Optimierung von **ActivityInterval**, **ActivityCount** und **TraceLevel** in der Datei `mqat.ini` vorgenommen wird.

## Informationen zu diesem Vorgang

Wenn Sie den Aktivitätstrace der Anwendungsaktivität selektiv für eine Anwendung oder für alle WS-Manager-Anwendungen aktivieren, können zusätzliche Messaging-Aktivitäten und im Warteschlangenmanager zusätzlicher Speicherbereich erforderlich sein. In Umgebungen, in denen die Messaging-Leistung kritisch ist, z. B. in Anwendungen mit hoher Auslastung oder wenn ein Service-Level-Agreement (SLA) eine minimale Antwortzeit vom Messaging-Provider erfordert, ist es unter Umständen nicht sinnvoll, einen Trace der Anwendungsaktivität zu erfassen, oder es kann erforderlich sein, die Details oder die Häufigkeit von Nachrichten, die von Trace-Aktivitäten erzeugt werden, anzupassen. Die vordefinierten Werte für **ActivityInterval**, **ActivityCount** und **TraceLevel** in der Datei `mqat.ini` ergeben eine ausgeglichene Standardeinstellung für Details und Leistung. Sie können diese Werte jedoch optimieren, um die präzisen funktionalen und leistungsspezifischen Anforderungen Ihres Systems zu erfüllen.

## Prozedur

- Trace nur für die Anwendungen, die Sie benötigen.

Erstellen Sie hierzu für 'ApplicationTrace' eine anwendungsspezifische Zeilengruppe in `mqat.ini` oder ändern Sie die Anwendung so, dass `MQCNO_ACTIVITY_TRACE_ENABLED` im Optionsfeld in der Struktur **MQCNO** für einen Aufruf von `MQCONN` angegeben ist. Weitere Informationen hierzu finden Sie unter „[Aktivitätstraceverhalten mit mqat.ini konfigurieren](#)“ auf Seite 211 und „[Festlegen von MQCONN-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen](#)“ auf Seite 210.

- Bevor Sie den Trace starten, überprüfen Sie, ob mindestens eine Anwendung aktiv ist und bereit ist, die Aktivitäts-Trace-Nachrichtendaten aus der Datei `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE` abzurufen.
- Halten Sie die Warteschlangenlänge so niedrig wie möglich, indem Sie die Anzahl der Anwendungen erhöhen, die die Warteschlange in die Warteschlange stellen.
- Legen Sie den Wert für **TraceLevel** in der Datei `mqat.ini` so fest, dass die erforderliche Mindestmenge an Daten erfasst wird.

`TraceLevel=LOW` hat die geringste Auswirkung auf die Messaging-Leistung. Weitere Informationen finden Sie im Abschnitt „[Aktivitätstraceverhalten mit mqat.ini konfigurieren](#)“ auf Seite 211.

- Optimieren Sie die Werte für **ActivityCount** und **ActivityInterval** in `mqat.ini`, um die Häufigkeit der Generierung von Aktivitätstracenachrichten anzupassen.

Wenn Sie mehrere Anwendungen verfolgen, werden die Aktivitätstracenachrichten möglicherweise schneller erstellt, als sie aus der Datei `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE` entfernt werden können. Wenn Sie jedoch die Anzahl der generierten Aktivitätstracenachrichten verringern, erhöhen Sie auch den Speicherplatz, der vom Warteschlangenmanager benötigt wird, und die Größe der Nachrichten, wenn sie in die Warteschlange geschrieben werden.

## Nächste Schritte

Die einfachste Methode zur Anzeige der Inhalte von Tracenachrichten zur Anwendungsaktivität ist, das „[Beispielprogramm 'amqsact'](#)“ auf Seite 218 zu verwenden.

## Beispielprogramm 'amqsact'

**amqsact** formatiert Tracenachrichten zur Anwendungsaktivität für Sie und wird mit IBM MQ bereitgestellt.

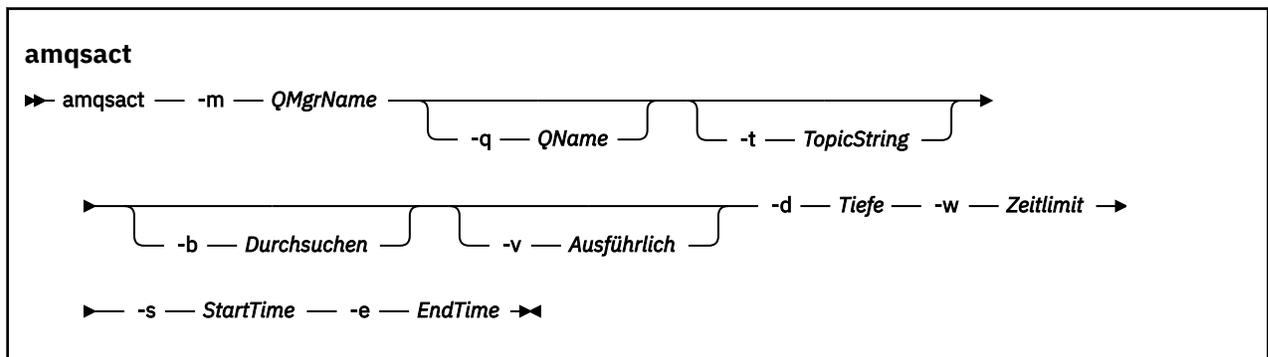
Das kompilierte Programm befindet sich im Verzeichnis 'samples':

- **Linux** **UNIX** Unter Linux und UNIX MQ\_INSTALLATION\_PATH/samp/bin
- **Windows** Unter Windows MQ\_INSTALLATION\_PATH\tools\c\Samples\Bin

## Anzeigemodus

Standardmäßig verarbeitet **amqsact** im Anzeigemodus Nachrichten unter SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. Sie können dieses Verhalten außer Kraft setzen, indem Sie einen Warteschlangennamen oder eine Themenzeichenfolge angeben.

Sie können auch den angezeigten Ablaufverfolgungszeitraum steuern und angeben, ob die Aktivitätstrace-Nachrichten nach der Anzeige entfernt oder beibehalten werden sollen.



## Erforderliche Parameter für den Anzeigemodus

### -m QMgrName

Name des Warteschlangenmanagers.

### -d Depth

Die Anzahl der Datensätze, die angezeigt werden sollen.

### -w Timeout

Zeit zu warten, in Sekunden. Wenn im angegebenen Zeitraum keine Tracenachrichten angezeigt werden, wird **amqsact** beendet.

### -s StartTime

Startzeit des Datensatzes für den Prozess.

### -e EndTime

Endzeit des Datensatzes für den Prozess.

## Optionale Parameter für den Anzeigemodus

### -q QName

Geben Sie eine bestimmte Warteschlange an, um den Standardwarteschlangennamen zu überschreiben.

### -t TopicString

Abonnieren eines Ereignisthemas

### -b

Nur Datensätze durchsuchen

### -v

Ausführliche Ausgabe

## Beispielausgabe für den Anzeigemodus

Verwenden Sie **amqsact** im Warteschlangenmanager *TESTQM* mit einer ausführlichen Ausgabe in einem MQCONN-API-Aufruf:

```
amqsact -m TESTQM -v
```

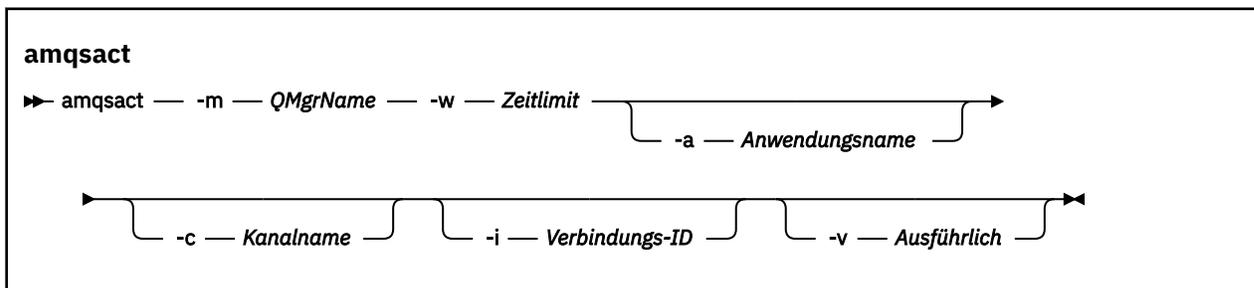
Der vorhergehende Befehl gibt die folgende Beispielausgabe aus:

```
MonitoringType: MQI Activity Trace
Correl_id:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM  '
00000010: B5F6 4251 2000 E601
QueueManager: 'TESTQM'
Host Name: 'ADMINIB-1VTJ6N1'
IntervalStartDate: '2014-03-15'
IntervalStartTime: '12:08:10'
IntervalEndDate: '2014-03-15'
IntervalEndTime: '12:08:10'
CommandLevel: 750
SeqNumber: 0
ApplicationName: 'IBM MQ_1\bin\amqsput.exe'
Application Type: MQAT_WINDOWS_7
ApplicationPid: 14076
UserId: 'Emma_Bushby'
API Caller Type: MQXACT_EXTERNAL
API Environment: MQXE_OTHER
Application Function: ''
Appl Function Type: MQFUN_TYPE_UNKNOWN
Trace Detail Level: 2
Trace Data Length: 0
Pointer size: 4
Platform: MQPL_WINDOWS_7
MQI Operation: 0
Operation Id: MQXF_CONN
ApplicationTid: 1
OperationDate: '2014-03-15'
OperationTime: '12:08:10'
ConnectionId:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM  '
00000010: FFFFFFFB5FFFFFFFFF6 4251 2000 FFFFFFFE601  '
QueueManager: 'TESTQM'
Completion Code: MQCC_OK
Reason Code: 0
```

## Dynamischer Modus

Sie können den dynamischen Modus aktivieren, indem Sie einen Anwendungsnamen, einen Kanalnamen oder eine Verbindungs-ID als Argument für **amqsact** angeben. Beachten Sie, dass Sie Platzhalterzeichen im Namen verwenden können.

Im dynamischen Modus werden die Aktivitätstracedaten am Anfang der Stichprobe durch die Verwendung eines unfähigen Subskriptionsabonnements für ein Systemthema aktiviert. Die Erfassung von Aktivitätstracedaten wird gestoppt, wenn **amqsact** gestoppt wird. Sie müssen ein Zeitlimit für **amqsact** im dynamischen Modus angeben. Sie können mehrere Kopien von **amqsact** gleichzeitig ausführen, wobei jede Instanz eine Kopie aller Aktivitätstracedaten erhält.



## Erforderliche Parameter für den dynamischen Modus

### -m *QMgrName*

Name des Warteschlangenmanagers.

### -w *Timeout*

Zeit zu warten, in Sekunden. Wenn im angegebenen Zeitraum keine Tracenachrichten angezeigt werden, wird **amqsact** beendet.

## Optionale Parameter für den dynamischen Modus

### -a *Application name*

Geben Sie einen Anwendungsnamen für die Erfassung von Nachrichten für

### -c *Channel name*

Geben Sie einen Kanal an, in dem Nachrichten erfasst werden

### -i *Connection ID*

Geben Sie eine Verbindung an, für die Nachrichten erfasst werden sollen.

### -v

Ausführliche Ausgabe

## Beispielausgabe für den dynamischen Modus

Mit dem folgenden Befehl werden Aktivitätstracenachrichten für Verbindungen generiert, die von Anwendungen, die mit dem Text " amqs " beginnen, hergestellt werden. Nach 30 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

Mit dem folgenden Befehl werden Aktivitätstracenachrichten für Aktivitäten auf dem Kanal QMGR1.TO.QMGR2 generiert und angezeigt. Nach 10 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

Mit dem folgenden Befehl werden ausführliche Aktivitätstracenachrichten für eine beliebige Aktivität in der vorhandenen IBM MQ-Verbindung mit den CONN-Wert "6B576B5420000701" und dem EXTCNN-Wert "414D5143514D475231202020202020" generiert und angezeigt. Nach einer Minute Inaktivität wird das **amqsact** -Programm beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D4752312020202020206B576B5420000701 -v
```

## Trace-Informationen für Anwendungsaktivitäten abonnieren

Als Alternative zur Erfassung von Informationen über die Konfiguration auf Warteschlangenmanagerebene können Sie ab IBM MQ 9.0 Traceinformationen zur Anwendungsaktivität auch dynamisch abonnieren.

## Informationen zu diesem Vorgang

Der Trace zur Anwendungsaktivität verfolgt das Verhalten einer Anwendung und stellt eine detaillierte Ansicht der Parameter bereit, die von einer Anwendung bei der Interaktion mit IBM MQ-Ressourcen verwendet werden. Es zeigt auch die Reihenfolge der MQI-Aufrufe, die von einer Anwendung ausgegeben werden.

Vor IBM MQ 9.0 wird der Trace der Anwendungsaktivität erfasst und gelesen, indem PCF-Nachrichten des Aktivitätstrace in die Systemwarteschlange SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. Weitere Informati-

onen finden Sie unter [„Zentrale Erfassung von Trace-Informationen für Anwendungsaktivitäten konfigurieren“](#) auf Seite 208.

Zusätzlich zum Schreiben von Tracedaten in die Systemwarteschlange bietet IBM MQ ab IBM MQ 9.0 die Möglichkeit, Aktivitätstracedaten dynamisch zu subscribieren, die in spezielle IBM MQ -Systemthemen geschrieben werden, anstatt die Informationen über die Konfiguration auf Warteschlangenmanagerebene zu erfassen.

Beachten Sie, dass IBM MQ 9.0 zu diesem Zweck keine Exits verwendet. Wenn Sie bereits Exits zum Trace der Anwendungsaktivität verwendet haben, müssen Sie zur Verwendung der neuen Methode für die Erfassung von Anwendungsaktivitätstrace wechseln.

Die Erstellung einer Subskription aktiviert den Aktivitätstrace. Sie müssen keine WS-Manager-oder Anwendungsattribute für die zentrale Erfassung von Tracedaten festlegen. Eine explizite Blockierung des Aktivitätstrace durch Inaktivieren des Trace auf WS-Manager-oder Anwendungsebene blockiert jedoch auch den Aktivitätstrace, wenn er an alle übereinstimmenden Subskriptionen zugestellt wird.

## Prozedur

- [„Trace für Subskriptionen für Anwendungsaktivitäten“](#) auf Seite 221
- [„Subskriptionen für Anwendungsaktivitätstrace erstellen“](#) auf Seite 221
- [„amqsact zum Anzeigen von Tracenachrichten verwenden“](#) auf Seite 223
- [„Tracestufen mit mqat.ini konfigurieren“](#) auf Seite 225

### ***Trace für Subskriptionen für Anwendungsaktivitäten***

Sie können ein IBM MQ-Systemthema subscribieren, um Traceinformationen für Anwendungsaktivitäten zu erfassen.

Sie subscribieren eine bestimmte IBM MQ-Systemthemenzeichenfolge, die die Aktivität darstellt, für die ein Trace durchgeführt werden soll. Das Subscribieren generiert automatisch Aktivitäts-Trace-Daten nachrichten und veröffentlicht sie in der Zielwarteschlange der Subskription. Wenn Sie die Subskription löschen, wird die Generierung von Aktivitätstracedaten für diese Subskription gestoppt.

Eine Subskription kann die Aktivität auf einer der folgenden Ressourcen verfolgen:

- Eine angegebene Anwendung
- Ein angegebener IBM MQ-Kanal
- Eine vorhandene IBM MQ-Verbindung

Sie können mehrere Subskriptionen mit unterschiedlichen oder denselben Themenzeichenfolgen erstellen. Wenn Sie mehrere Subskriptionen mit denselben Trace-Themenzeichenfolgen für Systemaktivitäten erstellen, empfängt jede Subskription eine Kopie der Aktivitätstracedaten, und dies kann negative Auswirkungen auf die Leistung haben.

Die Aktivierung einer beliebigen Stufe des Aktivitätstrace kann negative Auswirkungen auf die Leistung haben. Je mehr Subskriptionen oder die mehr Ressourcen subscribiert haben, um so größer ist der potenzielle Leistungsaufwand. Um den Aufwand für die Erfassung der Aktivitätstrace zu minimieren, werden die Daten in Nachrichten geschrieben und asynchron von der Anwendungsaktivität selbst an die Subskriptionen zugestellt. Häufig werden mehrere Operationen in eine einzelne Aktivitäts-Trace-Datennachricht geschrieben. Die asynchrone Operation kann eine Verzögerung zwischen der Anwendungsoperation und dem Empfang der Tracedaten, die die Operation aufzeichnet, einleiten.

### ***Subskriptionen für Anwendungsaktivitätstrace erstellen***

Sie können Subskriptionen für bestimmte Themen erstellen, um Tracedaten für Anwendungsaktivitäten in IBM MQ 9.0 zu erfassen.

Wenn eine Subskription für bestimmte Systemthemenzeichenfolgen erstellt wird, werden die entsprechenden PCF-Datennachrichten der Aktivitätstrace automatisch in dieser Subskription veröffentlicht. Ausführliche Informationen zum Subscribieren von Themen finden Sie unter [Publish/Subscribe-Messaging](#).

Die Themenzeichenfolgen haben das folgende Format:

```
$SYS/MQ/INFO/QMGR/qmgr_name/ActivityTrace/resource_type/resource_identifizier
```

Dabei gilt Folgendes:

- *Warteschlangenmanagername* gibt den Warteschlangenmanager an, mit dem die verfolgte Anwendung verbunden ist. *Warteschlangenmanagername* ist der Name des Warteschlangenmanagers, wobei alle abschließenden Leerzeichen entfernt und alle Schrägstriche (/) durch ein Et-Zeichen (&) ersetzt werden. Zeichen.
- *resource\_type* gibt den Typ der Ressourcendaten an, die für erfasst werden, und ist eine der folgenden Zeichenfolgen:
  - *AppName* zum Angeben einer Anwendung. In der Anforderung werden alle IBM MQ-Verbindungen mit einem Anwendungsnamen subskribiert, der mit einem der durch die *Ressourcen-ID* angegebenen Anwendungsnamen übereinstimmt.
  - *ChannelName* für die Angabe eines IBM MQ-Kanals.
  - *ConnectionId* für die Angabe einer IBM MQ-Verbindung.
- *resource\_identifizier* gibt die tatsächliche Ressource an. Das Format richtet sich nach dem Ressourcentyp:
  - Bei einem Ressourcentyp *AppName* ist *Ressourcen-ID* der abschließende Teil (der Wert, der auf das letzte Zeichen / oder \ folgt) im Anwendungsnamen für den Warteschlangenmanager, wobei alle abschließenden Leerzeichen entfernt wurden. Der Wert stimmt mit dem Wert für *AppName* aus der API-Exit-Kontextstruktur (MQAXC) überein. Der *AppName* einer Verbindung wird als APPLTAG-Wert zurückgegeben, wenn Sie den MQSC-Befehl **DISPLAY CONN** verwenden.
  - Für einen Ressourcentyp von *ChannelName* ist der *resource\_identifizier* der Name des Kanals, für den ein Trace durchgeführt werden soll. Wenn der Kanalname einen SVRCONN-Kanal identifiziert, wird ein Trace für alle Anwendungsaktivitäten für verbundene Clients durchgeführt. Wenn der Kanalname einen Warteschlangenmanager für den WS-Manager-Kanal identifiziert, wird ein Trace der eingehenden und abgehenden Nachrichten durchgeführt. Die *Ressourcen-ID* ist der Kanalname, aus dem alle abschließenden Leerzeichen entfernt und alle Zeichen '/' durch ein '&' ersetzt werden. Zeichen.
  - Für den Ressourcentyp *ConnectionId* ist die *Ressourcen-ID* die eindeutige Verbindungs-ID, die jeder Verbindung zugeordnet ist. Die Verbindungs-ID in der Themenzeichenfolge ist der vollständige 24-Byte-Wert, der als hexadezimale Zeichenfolge geschrieben wird. Dieser Wert ist die Verkettung des EXTCNN gefolgt von den CONN-Werten, die vom MQSC-Befehl **DISPLAY CONN** zurückgegeben werden.

Sie können Platzhalterzeichen in einem *resource\_identifizier* verwenden, um mehrere Ressourcenidentitäten in einer einzigen Subskription abzugleichen. Das Platzhalterzeichen kann entweder im Standardthemenstil ('#' oder '+') oder im Zeichenstil ('\*' oder '?') verwendet werden. Wenn Sie das Platzhalterzeichen für den Topic-Stil verwenden, kann es nicht mit einem Teil eines Ressourcennamens kombiniert werden. Es kann nur verwendet werden, um alle möglichen Anwendungen, Kanäle oder Verbindungen abzugleichen. Die Verwendung von Platzhalterzeichen erhöht die Stufe der generierten Tracedaten, die sich auf die Leistung auswirken können.

Wenn Sie diese Themenzeichenfolgen subskribieren möchten, müssen Sie die Berechtigung " subscribe " haben. Systemthemen übernehmen keine Berechtigungen vom Stamm der Themenstruktur des Warteschlangenmanagers. Ein Benutzer muss Zugriff auf ein verwaltetes Themenobjekt erhalten, das sich an oder über dem \$SYS/MQ-Punkt in der Themenstruktur befinden kann. Sie können subskribieren, wenn Sie Zugriff auf die Warteschlange SYSTEM.ADMIN.TOPIC haben, obwohl dies Zugriff auf alle \$SYS/MQ-Themenzeichenfolgen gewährt, nicht nur für den Aktivitätstrace. Um den Zugriff genauer zu steuern, können neue verwaltete Themenobjekte für tiefere Punkte in der Baumstruktur definiert werden, entweder für alle Aktivitätstrace oder z. B. für einen bestimmten Anwendungsnamen oder einen bestimmten Kanalnamen.

## Beispiele

Das folgende Beispiel zeigt eine Themenzeichenfolge für eine Anwendung namens amqspu<sub>t</sub>c, die auf einem Windows-System ausgeführt wird:

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqsputc.exe
```

Das folgende Beispiel zeigt eine Themenzeichenfolge für einen Kanal:

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/SYSTEM.DEF.SVRCONN
```

Das folgende Beispiel zeigt eine Themenzeichenfolge für eine Verbindung:

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/Connection□  
Id/414D5143514D4752312020202020206B576B5420000701
```

Das folgende Beispiel zeigt eine Themenzeichenfolge, die eine Subskription für die Tracedaten für alle Kanäle auf dem Warteschlangenmanager QMGR1 erstellt:

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/#
```

Das folgende Beispiel zeigt eine Themenzeichenfolge, die eine Subskription erstellt, um Tracedaten für Anwendungen mit Namen zu erstellen, die mit " amqs " beginnen. (Beachten Sie, dass die Subskription mit dem Platzhalterzeichen " \* " erstellt werden muss.)

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqs*
```

## Zugehörige Konzepte

„Systemthemen für Überwachung und Aktivitätstrace“ auf Seite 298

Systemthemen in Themenstrukturen des Warteschlangenmanagers werden für die Ressourcenüberwachung (einige davon ähneln dem Inhalt von Statistiknachrichten) und als Methode zur Verarbeitung von Anwendungsaktivitätstraces verwendet.

## **amqsact zum Anzeigen von Tracenachrichten verwenden**

Sie können das Programm **amqsact** mit IBM MQ 9.0 verwenden, um Tracenachrichten zu generieren und anzuzeigen.

Bei dem Programm **amqsact** handelt es sich um ein IBM MQ-Beispiel. Um dieses Beispiel zu verwenden, müssen Sie die mit dem Client verbundene ausführbare Datei **amqsactc** verwenden. Die ausführbare Datei befindet sich im Verzeichnis 'samples':

- Linux- und UNIX-Plattformen: `MQ_INSTALLATION_PATH/samp/bin64`
- Windows-Plattformen: `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`

Sie können **amqsact** auf zwei Arten verwenden:

### Anzeigemodus

Formatieren und zeigen Sie die Tracedaten für Aktivitätstraces an, die an SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE zugestellt werden.

### Dynamischer Modus

Erstellen Sie eine Subskription für eine Gruppe von Ressourcen und zeigen Sie den generierten Aktivitätstrace an, indem Sie **amqsact** ausführen.

## Anzeigemodus

Standardmäßig verarbeitet **amqsact** im Anzeigemodus Nachrichten unter SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. Sie können dieses Verhalten außer Kraft setzen, indem Sie einen Warteschlangennamen oder eine Themenzeichenfolge angeben. Der Aktivitätstrace muss aktiviert werden, indem eine der

Methoden verwendet wird, die im Abschnitt [Traceinformationen für Anwendungsaktivitäten erfassen](#) beschrieben werden. Sie können den angezeigten Ablaufverfolgungszeitraum steuern und angeben, ob die Aktivitätstrace-Nachrichten nach der Anzeige entfernt oder beibehalten werden sollen. Im Anzeigemodus verwendet **amqsact** die folgenden Argumente:

**-m *queue\_manager\_name***

Erforderlich. Geben Sie den Warteschlangenmanager an, für den Tracenachrichten erfasst werden.

**-q *queue\_name***

Es werden nur Tracenachrichten angezeigt, die sich auf die benannte Warteschlange beziehen.

**-t *topic\_string***

Es werden nur Tracenachrichten angezeigt, die sich auf das benannte Thema beziehen.

**-b**

Geben Sie an, dass Tracenachrichten nach der Anzeige beibehalten werden

**-v**

Tracenachrichten im ausführlichen Modus anzeigen.

**-d *depth***

Die Anzahl der Nachrichten, die angezeigt werden sollen.

**-w *timeout***

Geben Sie ein Zeitlimit an. Wenn in diesem Zeitraum keine Tracenachrichten angezeigt werden, wird **amqsact** beendet.

**-s *start\_time***

Verwenden Sie dieses Argument mit dem Argument -e, um einen Zeitraum anzugeben. Es werden Tracenachrichten aus dem angegebenen Zeitraum angezeigt.

**-e *end\_time***

Verwenden Sie dieses Argument mit dem Argument -s, um einen Zeitraum anzugeben. Es werden Tracenachrichten aus dem angegebenen Zeitraum angezeigt.

Der folgende Befehl zeigt beispielsweise Aktivitätstrace-Nachrichten an, die in SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE gehalten werden, und löscht die Nachrichten nach der Anzeige:

```
amqsact -m QMGR1
```

Mit dem folgenden Befehl werden Aktivitätstracenachrichten in der angegebenen Warteschlange SUB.QUEUE angezeigt, und die Nachrichten werden nach der Anzeige gelöscht. Die Nachrichten werden weiterhin bis zu einem Zeitraum von 30 Sekunden angezeigt, ohne dass neue Nachrichten vergehen. Dieser Befehl kann beispielsweise mit einer Subskription für eine Themenzeichenfolge des Aktivitätstrace-Systems verwendet werden.

```
amqsact -m QMGR1 -q SUB.QUEUE.1 -w 30
```

Der folgende Befehl zeigt alle Aktivitätstracedaten im ausführlichen Format an, die derzeit in der Warteschlange SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE enthalten sind, die in dem angegebenen Zeitraum von 20 Minuten aufgetreten ist. Die Nachrichten bleiben nach der Anzeige in der Warteschlange.

```
amqsact -m QMGR1 -b -v -s 2014-12-31 23.50.00 -e 2015-01-01 00.10.00
```

## Dynamischer Modus

Sie aktivieren den dynamischen Modus, indem Sie einen Anwendungsnamen, einen Kanalnamen oder eine Verbindungs-ID als Argument für **amqsact** angeben. Sie können Platzhalterzeichen im Namen verwenden. Im dynamischen Modus werden die Aktivitätstracedaten am Anfang des Beispiels durch die Verwendung einer nicht permanenten Subskription für ein Systemthema aktiviert. Die Erfassung von Aktivitätstracedaten wird gestoppt, wenn **amqsact** gestoppt wird. Sie müssen ein Zeitlimit für **amqsact** im dynamischen Modus angeben. Sie können mehrere Kopien von **amqsact** gleichzeitig ausführen und

jede Instanz erhält eine Kopie aller Aktivitätstracedaten. Im dynamischen Modus verwendet **amqsact** die folgenden Argumente:

**-m queue\_manager\_name**

Erforderlich. Geben Sie den Warteschlangenmanager an, für den Tracenachrichten erfasst werden.

**-w timeout**

Erforderlich. Geben Sie ein Zeitlimit an. Wenn in diesem Zeitraum keine Tracenachrichten angezeigt werden, wird **amqsact** beendet.

**-a application\_name**

Geben Sie eine Anwendung an, für die Nachrichten erfasst werden sollen.

**-c channel\_name**

Geben Sie einen Kanal an, für den Nachrichten erfasst werden sollen.

**-i connection\_id**

Geben Sie eine Verbindung an, für die Nachrichten erfasst werden sollen.

**-v**

Tracenachrichten im ausführlichen Modus anzeigen.

Der folgende Befehl generiert beispielsweise Aktivitäts-Trace-Nachrichten für alle Verbindungen, die von Anwendungen mit dem Namen " amqsget.exe " erstellt werden. Nach 30 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 30 -a amqsget.exe
```

Der folgende Befehl generiert Aktivitäts-Trace-Nachrichten für alle Verbindungen, die von Anwendungen erstellt werden, die mit dem Text " amqs " beginnen. Nach 30 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

Mit dem folgenden Befehl werden Aktivitätstracenachrichten für Aktivitäten auf dem Kanal QMGR1.TO.QMGR2 generiert und angezeigt. Nach 10 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

Mit dem folgenden Befehl werden Aktivitätstracenachrichten für jede Aktivität auf beliebigen Kanälen generiert und angezeigt. Nach 10 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 10 -c #
```

Mit dem folgenden Befehl werden ausführliche Aktivitätstracenachrichten für eine beliebige Aktivität in der vorhandenen IBM MQ-Verbindung mit den CONN-Wert "6B576B5420000701" und dem EXTCONN-Wert "414D5143514D47523120202020202020" generiert und angezeigt. Nach einer Minute Inaktivität wird das **amqsact** -Programm beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D4752312020202020202020206B576B5420000701 -v
```

### **Tracestufen mit mqat.ini konfigurieren**

Sie konfigurieren Tracestufen für einen Warteschlangenmanager, indem Sie der Konfigurationsdatei mqat.ini Werte für die Zeilengruppe 'AllActivityTrace' festlegen.

Sie können die folgenden Werte für die Zeilengruppe AllActivityTrace festlegen:

**ActivityInterval**

Zeitintervall in Sekunden zwischen Tracenachrichten. Der Aktivitätstrace verwendet keinen Zeitgeberthread, daher wird die Tracenachricht nicht zu dem Zeitpunkt geschrieben, zu dem die Zeit vergeht, sondern geschrieben wird, wenn die erste MQI-Operation nach Ablauf des Zeitintervalls ausgeführt wird. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung disconnects (oder wenn die Anzahl der Aktivitäten erreicht ist). Standardwert: 1.

**ActivityCount**

Die Anzahl der MQI-Operationen zwischen Tracenachrichten. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung die Verbindung trennt (oder wenn das Aktivitätsintervall abgelaufen ist). Der Standardwert ist 100.

**TraceLevel**

Die Menge an Parameterdetails, für die für jede Operation ein Trace durchgeführt wird. In der Beschreibung der einzelnen Operationen wird angegeben, welche Parameter für die einzelnen Trace-Ebenen angegeben werden. Setzen Sie die Option auf LOW, MEDIUM oder HIGH. Standardwert ist MEDIUM.

**TraceMessageData**

Die Menge der Nachrichtendaten, für die ein Trace in Byte für MQGET-, MQPUT-, MQPUT1- und Call-back-Operationen durchgeführt wird. Der Standardwert ist 0.

**StopOnGetTraceMsg**

Kann auf ON oder OFF gesetzt werden. Standardwert ist ON.

**SubscriptionDelivery**

Kann auf BATCHED oder IMMEDIATE gesetzt werden. Legt fest, ob die Parameter "ActivityInterval" und "ActivityCount" verwendet werden sollen, wenn eine oder mehrere Aktivitätstrace-Subskriptionen vorhanden sind. Wenn Sie diesen Parameter auf IMMEDIATE setzen, werden die Werte für "ActivityInterval" und "ActivityCount" mit den tatsächlichen Werten von 1 überschrieben, wenn die Trace-Daten eine übereinstimmende Subskription aufweisen. Jeder Aktivitätstracesatz wird nicht mit anderen Datensätzen aus derselben Verbindung ausgeliefert und stattdessen sofort an die Subskription gesendet, ohne dass es zu einer Verzögerung kommt. Mit der Einstellung IMMEDIATE wird der Leistungsaufwand für die Erfassung von Aktivitätstracedaten erhöht. Die Standardeinstellung ist BATCHED.

**Trace-Nachrichtenreferenz für Anwendungsaktivitäten**

Verwenden Sie diese Seite, um eine Übersicht über das Format der Tracenachrichten der Anwendungsaktivitäten und die in diesen Nachrichten zurückgegebenen Informationen zu erhalten.

Tracenachrichten für die Anwendungsaktivität sind standardmäßige IBM MQ-Nachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Die Nachrichtendaten enthalten Informationen zu den MQI-Operationen, die von IBM MQ-Anwendungen ausgeführt werden, oder Informationen zu den Aktivitäten in einem IBM MQ-System.

**Nachrichtendeskriptor**

- Eine MQMD-Struktur

**Nachrichtendaten**

- Ein PCF-Header (MQCFH)
- Tracenachrichtendaten der Anwendungsaktivität, die immer zurückgegeben werden
- Tracenachrichtendaten der Anwendungsaktivität, die betriebspezifisch sind

***Trace für Anwendungsaktivitätstrace MQMD (Nachrichtendeskriptor)***

Verwenden Sie diese Seite, um die Unterschiede zwischen dem Nachrichtendeskriptor der Anwendungsaktivitätstracenachrichten und dem Nachrichtendeskriptor von Ereignisnachrichten zu verstehen.

Die Parameter und Werte im Nachrichtendeskriptor der Anwendungsaktivitätstracenachricht sind dieselben wie im Nachrichtendeskriptor von Ereignisnachrichten, mit der folgenden Ausnahme:

### **Format**

Beschreibung: Formatname der Nachrichtendaten.  
Wert: **MQFMT\_ADMIN**  
Admin-Nachricht.

### **CorrelId**

Beschreibung: Korrelations-ID  
Wert: Initialisiert mit der Verbindungs-ID (ConnectionId) der Anwendung

### **MQCFH (PCF-Header)**

Verwenden Sie diese Seite, um die PCF-Werte anzuzeigen, die in der MQCFH-Struktur für eine Aktivitätstracenachricht enthalten sind.

Für eine Aktivitätstracenachricht enthält die MQCFH-Struktur die folgenden Werte:

#### **Type**

Beschreibung: Strukturtyp, der den Inhalt der Nachricht identifiziert.  
Datentyp: MQLONG.  
Wert: MQCFT\_APP\_ACTIVITY

#### **StrucLength**

Beschreibung: Länge der MQCFH-Struktur in Byte.  
Datentyp: MQLONG.  
Wert: MQCFH\_STRUC\_LENGTH

#### **Version**

Beschreibung: Strukturversionsnummer.  
Datentyp: MQLONG.  
Werte: MQCFH\_VERSION\_3

#### **Command**

Beschreibung: Befehls-ID. In diesem Feld wird die Kategorie der Nachricht angegeben.  
Datentyp: MQLONG.  
Werte: MQCMD\_ACTIVITY\_TRACE

#### **MsgSeqNumber**

Beschreibung: Nachrichtenfolgennummer. Dieses Feld ist die Folgennummer der Nachricht in einer Gruppe zusammengehöriger Nachrichten.  
Datentyp: MQLONG.  
Werte: 1

#### **Control**

Beschreibung: Steueroptionen.  
Datentyp: MQLONG.

Werte: MQCFC\_LAST.

### **CompCode**

Beschreibung: Beendigungscode.

Datentyp: MQLONG.

Werte: MQCC\_OK.

### **Reason**

Beschreibung: Beendigungscode für Ursachencode.

Datentyp: MQLONG.

Werte: MQRC\_NONE.

### **ParameterCount**

Beschreibung: Anzahl der Parameterstrukturen. Dieses Feld ist die Anzahl der Parameterstrukturen, die auf die MQCFH-Struktur folgen. Eine Gruppenstruktur (MQCFGR) und die darin enthaltenen Parameterstrukturen werden nur als eine einzige Struktur gezählt.

Datentyp: MQLONG.

Werte: 1 oder höher

### **Tracenachrichtendaten für Anwendungsaktivitäten**

Unmittelbar hinter dem PCF-Header befindet sich eine Gruppe von Parametern, die das Zeitintervall für den Aktivitätstrace beschreiben. Diese Parameter geben auch die Reihenfolge der Nachrichten im Falle von Nachrichten an, die geschrieben werden. Die Reihenfolge und die Anzahl der Felder, die auf den Header folgen, ist nicht gewährleistet, so dass weitere Informationen in die Zukunft hinzugefügt werden können.

Nachrichtenname: Die Aktivitätstracenachricht

---

Systemwarteschlange: SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

### **QueueManager**

Beschreibung: Der Name des Warteschlangenmanagers.

ID: MQCA\_Q\_MGR\_NAME

Datentyp: MQCFST

Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH

### **QSGName**



Beschreibung: Der Name der Gruppe mit gemeinsamer Warteschlange, in der der Warteschlangenmanager Mitglied ist (nur z/OS).

ID: MQCA\_QSG\_NAME

Datentyp: MQCFST

Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH

### **HostName**

Beschreibung: Der Hostname der Maschine, auf der der WS-Manager ausgeführt wird.

ID: MQCACF\_HOST\_NAME  
Datentyp: MQCFST

### ***IntervalStartDate***

Beschreibung: Das Datum des Beginns des Überwachungszeitraums.  
ID: MQCAMO\_START\_DATE  
Datentyp: MQCFST  
Maximale Länge: MQ\_DATE\_LENGTH

### ***IntervalStartTime***

Beschreibung: Der Zeitpunkt des Startzeitpunktes des Überwachungszeitraums.  
ID: MQCAMO\_START\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH

### ***IntervalEndDate***

Beschreibung: Das Datum des Endes des Überwachungszeitraums.  
ID: MQCAMO\_END\_DATE  
Datentyp: MQCFST  
Maximale Länge: MQ\_DATE\_LENGTH

### ***IntervalEndTime***

Beschreibung: Die Zeit bis zum Ende des Überwachungszeitraums.  
ID: MQCAMO\_END\_TIME  
Datentyp: MQCFST  
Maximale Länge: MQ\_TIME\_LENGTH

### ***CommandLevel***

Beschreibung: Die IBM MQ-Befehlsebene.  
ID: MQIA\_COMMAND\_LEVEL  
Datentyp: MQCFIN

### ***SeqNumber***

Beschreibung: Die Folgenummer ist normalerweise null. Dieser Wert wird für jeden nachfolgenden Datensatz für lange laufende Verbindungen erhöht.  
ID: MQIACF\_SEQUENCE\_NUMBER  
Datentyp: MQCFIN

### ***ApplicationName***

Beschreibung: Der Name der Anwendung (Programmname).  
ID: MQCACF\_APPL\_NAME  
Datentyp: MQCFST

Maximale Länge: MQ\_APPL\_NAME\_LENGTH

### ***ApplClass***

Beschreibung: Typ der Anwendung, die die Aktivität ausgeführt hat. Mögliche Werte: MQAT\_\*

ID: MQIA\_APPL\_TYPE

Datentyp: MQCFIN

### ***ApplicationPid***

Beschreibung: Die Prozess-ID des Betriebssystems der Anwendung.

ID: MQIACF\_PROCESS\_ID

Datentyp: MQCFIN

### ***UserId***

Beschreibung: Der Benutzer-ID-Kontext der Anwendung.

ID: MQCACF\_USER\_IDENTIFIER

Datentyp: MQCFST

Maximale Länge: MQ\_USER\_ID\_LENGTH

### ***APICallerType***

Beschreibung: Der Typ der Anwendung. Mögliche Werte: MQXACT\_EXTERNAL oder MQXACT\_INTERNAL.

ID: MQIACF\_API\_CALLER\_TYPE

Datentyp: MQCFIN

### ***Environment***

Beschreibung: Die Laufzeitumgebung der Anwendung. Mögliche Werte: MQXE\_\*

ID: MQIACF\_API\_ENVIRONMENT

Datentyp: MQCFIN

### ***ChannelName***

Beschreibung: Der Kanalname, der der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE\_MCA' oder 'MQXE\_MCA\_SVRCONN' ist.

ID: MQCACH\_CHANNEL\_NAME

Datentyp: MQCFST

Maximale Länge: MQ\_CHANNEL\_NAME\_LENGTH

### ***ConnectionName***

Beschreibung: Der Netzverbindungsname, der der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE\_MCA' oder 'MQXE\_MCA\_SVRCONN' ist.

ID: MQCACH\_CONNECTION\_NAME

Datentyp: MQCFST

Maximale Länge: MQ\_CONN\_NAME\_LENGTH

### **ChannelType**

Beschreibung: Der Typ des Kanals, der der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE\_MCA' oder 'MQXE\_MCA\_SVRCONN' ist. Mögliche Werte: MQCHT\_\*

ID: MQIACH\_CHANNEL\_TYPE

Datentyp: MQCFIN

### **V 9.1.0 RemoteProduct**

Beschreibung: Die ID des fernen Produkts, die der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE\_MCA' oder 'MQXE\_MCA\_SVRCONN' ist.

ID: MQCACH\_REMOTE\_PRODUCT

Datentyp: MQCFST

Maximale Länge: MQ\_REMOTE\_PRODUCT\_LENGTH

### **V 9.1.0 RemoteVersion**

Beschreibung: Die Version des fernen Produkts, die der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE\_MCA' oder 'MQXE\_MCA\_SVRCONN' ist.

ID: MQCACH\_REMOTE\_VERSION

Datentyp: MQCFST

Maximale Länge: MQ\_REMOTE\_VERSION\_LENGTH

### **FunctionName**

Beschreibung: Der Name der letzten übergeordneten Funktion, die durch den Anfangsthread eingeleitet wurde.

ID: MQCACF\_APPL\_FUNCTION

Datentyp: MQCFST

### **FunctionType**

Beschreibung: Der Typ der letzten übergeordneten Funktion, die durch den Anfangsthread eingeleitet wurde. Mögliche Werte: MQFUN\_\*

ID: MQIACF\_APPL\_FUNCTION\_TYPE

Datentyp: MQCFIN

### **Detail**

Beschreibung: Die Detailstufe, die für die Verbindung aufgezeichnet wird. Mögliche Werte: 1 = LOW 2 = MEDIUM 3 = HIGH

ID: MQIACF\_TRACE\_DETAIL

Datentyp: MQCFIN

### **TraceDataLength**

Beschreibung:	Die Länge der Nachrichtendaten (in Byte), für die ein Trace für diese Verbindung durchgeführt wird.
ID:	MQIACF_TRACE_DATA_LENGTH
Datentyp:	MQCFIN

### **PointerSize**

Beschreibung:	Die Länge (in Byte) von Zeigern auf der Plattform, auf der die Anwendung ausgeführt wird (zur Unterstützung bei der Interpretation von Binärstrukturen).
ID:	MQIACF_POINTER_SIZE
Datentyp:	MQCFIN

### **Plattform**

Beschreibung:	Die Plattform, auf der der WS-Manager ausgeführt wird. Mögliche Werte: MQPL_*
ID:	MQIA_PLATFORM
Datentyp:	MQCFIN

### **Variablenparameter für MQI-Operationen der Anwendungsaktivität**

Auf die Struktur der Anwendungsaktivitätsdaten MQCFGR folgt die Gruppe der PCF-Parameter, die der ausgeführten Operation entspricht. Die Parameter für jede Operation werden im folgenden Abschnitt definiert.

Die Tracestufe gibt die Stufe der Trace-Granularität an, die für die Parameter erforderlich ist, die in den Trace eingeschlossen werden sollen. Folgende Werte für die Tracestufe sind möglich:

#### 1. Niedrig

Der Parameter ist eingeschlossen, wenn die Traceverarbeitung für " low ", " medium " oder " high " für eine Anwendung konfiguriert ist. Diese Einstellung bedeutet, dass ein Parameter immer in der Gruppe `AppActivityData` für die Operation enthalten ist. Diese Parametergruppe reicht aus, um einen Trace für die MQI-Aufrufe, die eine Anwendung vornimmt, zu verfolgen und zu sehen, ob sie erfolgreich sind.

#### 2. Mittel

Der Parameter ist nur in der Gruppe `AppActivityData` für die Operation enthalten, wenn die Traceverarbeitung für " mittel " oder " high " für eine Anwendung konfiguriert ist. Diese Gruppe von Parametern fügt Informationen zu den Ressourcen hinzu, z. B. die von der Anwendung verwendeten Warteschlangen- und Topic-Namen.

#### 3. Hoch

Der Parameter ist nur in der Gruppe `AppActivityData` für die Operation enthalten, wenn die Traceverarbeitung für " hohe " Aktivitäten für eine Anwendung konfiguriert ist. Diese Gruppe von Parametern enthält Speicherauszüge der Strukturen, die an die MQI- und XA-Funktionen übergeben werden. Aus diesem Grund enthält er weitere Informationen zu den Parametern, die in MQI- und XA-Aufrufen verwendet werden. Die Speicherauszüge des Strukturspeichers sind flache Kopien der Strukturen. Um fehlerhafte Versuche zu vermeiden, Zeiger zu dereferenzieren, werden die Zeigerwerte in den Strukturen auf NULL gesetzt.

**Anmerkung:** Die Version der Struktur, für die ein Speicherauszug erstellt wird, ist nicht notwendigerweise identisch mit der Version, die von einer Anwendung verwendet wird. Die Struktur kann durch einen API-Überkreuzungsexit, durch den Aktivitäts-Trace-Code oder durch den Warteschlangenmanager geändert werden. Ein Warteschlangenmanager kann eine Struktur in einer späteren Version ändern,

aber der WS-Manager ändert ihn nie in eine frühere Version der Struktur. Dazu würde das Risiko eines Datenverlustes gefährdet.

### *MQBACK*

Die Anwendung hat die MQBACK-MQI-Funktion gestartet.

#### **CompCode**

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ	MQCFIN

#### **Reason**

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ	MQCFIN

#### **QMGrOpDuration**

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.  <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

### *MQBEGIN*

Die Anwendung hat die MQBEGIN-MQI-Funktion gestartet.

#### **CompCode**

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ	MQCFIN

#### **Reason**

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ	MQCFIN

## **MQBO**

Beschreibung:	Die MQBEGIN-Optionsstruktur. Dieser Parameter ist nicht eingeschlossen, wenn ein Nullzeiger für den MQBEGIN-Aufruf verwendet wird.
PCF-Parameter:	MQBACF_MQBO_STRUCT
Tracestufe:	3
Typ	MQCFBS
Länge:	Die Länge (in Byte) der MQBO-Struktur.

## **QMgrOpDuration**

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.  <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

## **MQCALLBACK**

Die Anwendung hat die Funktion MQCALLBACK gestartet.

## **ObjectHandle**

Beschreibung:	Die Objektkennung
PCF-Parameter:	MQIACF_HOBJ
Tracestufe:	1
Typ	MQCFIN

## **CallType**

Beschreibung:	Warum die Funktion aufgerufen wurde. Einer der MQCBCT_* -Werte
PCF-Parameter:	MQIACF_CALL_TYPE
Tracestufe:	1
Typ	MQCFIN

## **MsgBuffer**

Beschreibung:	Nachrichtendaten.
PCF-Parameter:	MQBACF_MESSAGE_DATA
Tracestufe:	1
Typ	MQCFBS
Länge:	Die Länge wird durch den Parameter TRACEDATA () gesteuert, der in der Konfiguration von APPTRACE festgelegt ist. Wenn TRACEDATA=NONE angegeben wird, wird dieser Parameter weggelassen.

### ***MsgLength***

Beschreibung:	Länge der Nachricht. (Taken wird aus dem Feld DataLength in der MQCBC-Struktur übernommen.)
PCF-Parameter:	MQIACF_MSG_LENGTH
Tracestufe:	1
Typ	MQCFIN

### ***HighResTime***

Beschreibung:	Zeit der Operation in Mikrosekunden seit Mitternacht, 1. Januar 1970 (UTC) <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert je nach Plattformunterstützung für hohe Auflösungszeitgeber.
PCF-Parameter:	MQIAMO64_HIGHRES_TIME
Tracestufe:	2
Typ	MQCFIN64

### ***ReportOptions***

Beschreibung:	Optionen für Berichtsnachrichten
PCF-Parameter:	MQIACF_REPORT
Tracestufe:	2
Typ	MQCFIN

### ***MsgType***

Beschreibung:	Typ der Nachricht
PCF-Parameter:	MQIACF_MSG_TYPE
Tracestufe:	2
Typ	MQCFIN

### ***Verfall***

Beschreibung:	Nachrichtenlebensdauer
PCF-Parameter:	MQIACF_EXPIRY
Tracestufe:	2
Typ	MQCFIN

### ***Format***

Beschreibung:	Name des Formats von Nachrichtendaten
PCF-Parameter:	MQCACH_FORMAT_NAME
Tracestufe:	2
Typ	MQCFST
Länge:	MQ_FORMAT_LENGTH

### ***Priorität***

Beschreibung:	Nachrichtenpriorität
---------------	----------------------

PCF-Parameter: MQIACF\_PRIORITY  
Tracestufe: 2  
Typ MQCFIN

### **Persistenz**

Beschreibung: Nachrichtenpersistenz  
PCF-Parameter: MQIACF\_PERSISTENCE  
Tracestufe: 2  
Typ MQCFIN

### **MsgId**

Beschreibung: Nachrichten-ID  
PCF-Parameter: MQBACF\_MSG\_ID  
Tracestufe: 2  
Typ MQCFBS  
Länge: MQ\_MSG\_ID\_LENGTH

### **CorrelId**

Beschreibung: Korrelations-ID  
PCF-Parameter: MQBACF\_CORREL\_ID  
Tracestufe: 2  
Typ MQCFBS  
Länge: MQ\_CORREL\_ID\_LENGTH

### **objektname**

Beschreibung: Der Name des geöffneten Objekts.  
PCF-Parameter: MQCACF\_OBJECT\_NAME  
Tracestufe: 2  
Typ MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH

### **ResolvedQName**

Beschreibung: Der lokale Name der Warteschlange, aus der die Nachricht abgerufen wurde.  
PCF-Parameter: MQCACF\_RESOLVED\_Q\_NAME  
Tracestufe: 2  
Typ MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH

### **ReplyToQueue**

Beschreibung: MQ\_Q\_NAME\_LENGTH  
PCF-Parameter: MQCACF\_REPLY\_TO\_Q

Tracestufe: 2  
Typ MQCFST

### ***ReplyToQMgr***

Beschreibung: MQ\_Q\_MGR\_NAME\_LENGTH  
PCF-Parameter: MQCACF\_REPLY\_TO\_Q\_MGR  
Tracestufe: 2  
Typ MQCFST

### ***CodedCharSetId***

Beschreibung: Zeichensatzkennung von Nachrichtendaten  
PCF-Parameter: MQIA\_CODED\_CHAR\_SET\_ID  
Tracestufe: 2  
Typ MQCFIN

### ***Encoding***

Beschreibung: Numerische Codierung von Nachrichtendaten.  
PCF-Parameter: MQIACF\_ENCODING  
Tracestufe: 2  
Typ MQCFIN

### ***PutDate***

Beschreibung: MQ\_PUT\_DATE\_LENGTH  
PCF-Parameter: MQCACF\_PUT\_DATE  
Tracestufe: 2  
Typ MQCFST

### ***PutTime***

Beschreibung: MQ\_PUT\_TIME\_LENGTH  
PCF-Parameter: MQCACF\_PUT\_TIME  
Tracestufe: 2  
Typ MQCFST

### ***ResolvedQName***

Beschreibung: Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT\_Q ist.  
PCF-Parameter: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
Tracestufe: 2  
Typ MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH.

### ***ResObjectString***

Beschreibung:	Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT_TOPIC lautet.
PCF-Parameter:	MQCACF_RESOLVED_OBJECT_STRING
Tracestufe:	2
Typ	MQCFST
Länge:	Länge variiert.

### ***ResolvedType***

Beschreibung:	Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT_Q', 'MQOT_TOPIC' oder 'MQOT_NONE'.
PCF-Parameter:	MQIACF_RESOLVED_TYPE
Tracestufe:	2
Typ	MQCFIN

### ***PolicyName***

Beschreibung:	Der Richtlinienname, der auf diese Nachricht angewendet wurde. <b>Anmerkung:</b> Nur AMS-geschützte Nachrichten
PCF-Parameter:	MQCA_POLICY_NAME
Tracestufe:	2
Typ	MQCFST
Länge:	MQ_OBJECT_NAME_LENGTH

### ***XmitqMsgId***

Beschreibung:	Die Nachrichten-ID der Nachricht im Header der Übertragungswarteschlange. <b>Anmerkung:</b> Nur bei Format: MQFMT_XMIT_Q_HEADER
PCF-Parameter:	MQBACF_XQH_MSG_ID
Tracestufe:	2
Typ	MQCFBS
Länge:	MQ_MSG_ID_LENGTH

### ***XmitqCorrelId***

Beschreibung:	Die Korrelations-ID der Nachricht im Header der Übertragungswarteschlange. <b>Anmerkung:</b> Nur bei Format: MQFMT_XMIT_Q_HEADER
PCF-Parameter:	MQBACF_XQH_CORREL_ID
Tracestufe:	2
Typ	MQCFBS
Länge:	MQ_CORREL_ID_LENGTH

### ***XmitqPutTime***

Beschreibung: Die Zeitzeit der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_PUT\_TIME

Tracestufe: 2

Typ MQCFST

Länge: MQ\_PUT\_TIME\_LENGTH

### ***XmitqPutDate***

Beschreibung: Das Datum, an dem die Nachricht in den Header der Übertragungswarteschlange gestellt wurde.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_PUT\_DATE

Tracestufe: 2

Typ MQCFST

Länge: MQ\_PUT\_DATE\_LENGTH

### ***XmitqRemoteQName***

Beschreibung: Das Ziel der fernen Warteschlange der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_REMOTE\_Q\_Name

Tracestufe: 2

Typ MQCFST

Länge: MQ\_Q\_NAME\_LENGTH

### ***XmitqRemoteQMGr***

Beschreibung: Die Nachrichten-ID der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_REMOTE\_Q\_MGR

Tracestufe: 2

Typ MQCFST

Länge: MQ\_MSG\_ID\_LENGTH

### ***MsgDescStructure***

Beschreibung: Die MQMD-Struktur. Dieser Parameter wird weggelassen, wenn ein MQGMO-Wert von Version 4 verwendet wurde, um zu fordern, dass anstelle eines MQMD eine Nachrichtenennung zurückgegeben wird.

PCF-Parameter: MQBACF\_MQMD\_STRUCT

Tracestufe: 3

Typ MQCFBS

Länge: Die Länge der MQMD-Struktur in Byte (die tatsächliche Größe hängt von der Strukturversion ab)

### ***GetMsgOptsStructure***

Beschreibung: Die MQGMO-Struktur.

PCF-Parameter: MQBACF\_MQGMO\_STRUCT

Tracestufe: 3

Typ MQCFBS

Länge: Die Länge der MQGMO-Struktur in Byte (die tatsächliche Größe ist abhängig von der Strukturversion)

### ***MQCBCContextStructure***

Beschreibung: Die MQCBC-Struktur.

PCF-Parameter: MQBACF\_MQCBC\_STRUCT

Tracestufe: 3

Typ MQCFBS

Länge: Die Länge in Byte der MQCBC-Struktur (die tatsächliche Größe hängt von der Strukturversion ab)

### ***QMGrOpDuration***

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64\_QMGR\_OP\_DURATION

Tracestufe: 2

Typ MQCFIN64

### ***MQCB***

Die Anwendung hat die Verwaltung der Callback-MQI-Funktion gestartet.

### ***CallbackOperation***

Beschreibung: Die Operation "Callback-Funktion verwalten". Setzen Sie diese Option auf einen der MQOP\_\*-Werte.

PCF-Parameter: MQIACF\_MQCB\_OPERATION

Tracestufe: 1

Typ MQCFIN

### ***CallbackType***

Beschreibung: Der Typ der Callback-Funktion (CallbackType-Feld aus der MQCBD-Struktur). Auf einen der MQCBT\_\*-Werte setzen

PCF-Parameter: MQIACF\_MQCB\_TYPE

Tracestufe: 1  
Typ MQCFIN

### **CallbackOptions**

Beschreibung: Die Callback-Optionen. Auf einen der MQCBDO\_\* -Werte setzen  
PCF-Parameter: MQIACF\_MQCB\_OPTIONS  
Tracestufe: 1  
Typ MQCFIN

### **CallbackFunction**

Beschreibung: Der Zeiger auf die Callback-Funktion, wenn er als Funktionsaufruf gestartet wurde.  
PCF-Parameter: MQBACF\_MQCB\_FUNCTION  
Tracestufe: 1  
Typ MQCFBS  
Länge: Größe von MQPTR

### **CallbackName**

Beschreibung: Der Name der Callback-Funktion, wenn er als dynamisch verlinktes Programm gestartet wird.  
PCF-Parameter: MQCACF\_MQCB\_NAME  
Tracestufe: 1  
Typ MQCFST  
Länge: Größe von MQCHAR128

### **ObjectHandle**

Beschreibung: Die Objektkennung  
PCF-Parameter: MQIACF\_HOBJ  
Tracestufe: 1  
Typ MQCFIN

### **MaxMsgLength**

Beschreibung: Maximale Nachrichtenlänge. Auf eine Ganzzahl setzen oder den Sonderwert MQCBD\_FULL\_MSG\_LENGTH  
PCF-Parameter: MQIACH\_MAX\_MSG\_LENGTH  
Tracestufe: 2  
Typ MQCFIN

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.  
PCF-Parameter: MQIACF\_COMP\_CODE  
Tracestufe: 1

Typ MQCFIN

### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.

PCF-Parameter: MQIACF\_REASON\_CODE

Tracestufe: 1

Typ MQCFIN

### **ResolvedQName**

Beschreibung: Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT\_Q ist.

PCF-Parameter: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME

Tracestufe: 2

Typ MQCFST

Länge: MQ\_Q\_NAME\_LENGTH.

### **ResObjectString**

Beschreibung: Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT\_TOPIC lautet.

PCF-Parameter: MQCACF\_RESOLVED\_OBJECT\_STRING

Tracestufe: 2

Typ MQCFST

Länge: Länge variiert.

### **ResolvedType**

Beschreibung: Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT\_Q', 'MQOT\_TOPIC' oder 'MQOT\_NONE'.

PCF-Parameter: MQIACF\_RESOLVED\_TYPE

Tracestufe: 2

Typ MQCFIN

### **Callback DescriptorStructure**

Beschreibung: Die MQCBD-Struktur. Dieser Parameter wird weggelassen, wenn ein NULL-MQCBC-Wert an den MQCB-Aufruf übergeben wird.

PCF-Parameter: MQBACF\_MQCBD\_STRUCT

Tracestufe: 3

Typ MQCFBS

Länge: Die Länge der MQCBC-Struktur in Byte.

### **MsgDescStructure**

Beschreibung: Die MQMD-Struktur. Der Parameter 'MsgDescStructure' wird übergangen, wenn ein MQMD-Wert NULL an den MQCB-Aufruf übergeben wird.

PCF-Parameter: MQBACF\_MQMD\_STRUCT

Tracestufe:	3
Typ	MQCFBS
Länge:	Die Länge der MQMD-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

### ***GetMsgOptsStructure***

Beschreibung:	Die MQGMO-Struktur. Dieser Parameter wird weggelassen, wenn ein NULL-MQGMO-Wert an den MQCB-Aufruf übergeben wird.
PCF-Parameter:	MQBACF_MQGMO_STRUCT
Tracestufe:	3
Typ	MQCFBS
Länge:	Die Länge der MQGMO-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

### ***QMGrOpDuration***

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.  <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

### ***MQCLOSE***

Die Anwendung hat die MQI-Funktion MQCLOSE gestartet.

### ***ObjectHandle***

Beschreibung:	Die Objektkennung
PCF-Parameter:	MQIACF_HOBJ
Tracestufe:	1
Typ	MQCFIN

### ***CloseOptions***

Beschreibung:	Optionen schließen
PCF-Parameter:	MQIACF_CLOSE_OPTIONS
Tracestufe:	1
Typ	MQCFIN

### ***CompCode***

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE

Tracestufe: 1  
Typ MQCFIN

### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.  
PCF-Parameter: MQIACF\_REASON\_CODE  
Tracestufe: 1  
Typ MQCFIN

### **ResolvedQName**

Beschreibung: Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT\_Q ist.  
PCF-Parameter: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
Tracestufe: 2  
Typ MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH.

### **ResObjectString**

Beschreibung: Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT\_TOPIC lautet.  
PCF-Parameter: MQCACF\_RESOLVED\_OBJECT\_STRING  
Tracestufe: 2  
Typ MQCFST  
Länge: Länge variiert.

### **ResolvedType**

Beschreibung: Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT\_Q', 'MQOT\_TOPIC' oder 'MQOT\_NONE'.  
PCF-Parameter: MQIACF\_RESOLVED\_TYPE  
Tracestufe: 2  
Typ MQCFIN

### **QMGrOpDuration**

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  
Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.  
**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.  
PCF-Parameter: MQIAMO64\_QMGR\_OP\_DURATION  
Tracestufe: 2  
Typ MQCFIN64

### *MQCMIT*

Die Anwendung hat die MQCMIT-MQI-Funktion gestartet.

#### **CompCode**

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ	MQCFIN

#### **Reason**

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ	MQCFIN

#### **QMGrOpDuration**

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.  <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

### *MQCONN und MQCONNX*

Die Anwendung hat die MQCONN-oder MQCONNX-MQI-Funktion gestartet.

#### **ConnectionId**

Beschreibung:	Die Verbindungs-ID (falls verfügbar) oder MQCONNID_NONE (falls nicht vorhanden)
PCF-Parameter:	MQBACF_CONNECTION_ID
Tracestufe:	1
Typ:	MQCFBS
Maximale Länge:	MQ_CONNECTION_ID_LENGTH

#### **QueueManagerName**

Beschreibung:	Der (nicht aufgelöste) Name des Warteschlangenmanagers, der im Aufruf MQCONN (X) verwendet wird.
PCF-Parameter:	MQCA_Q_MGR_NAME
Tracestufe:	1
Typ:	MQCFST
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.  
PCF-Parameter: MQIACF\_COMP\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.  
PCF-Parameter: MQIACF\_REASON\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **ConnectOptions**

Beschreibung: Von MQCNO\_\* -Werten abgeleitete Verbindungsoptionen  
**Anmerkung:** Nur MQCONNX  
PCF-Parameter: MQIACF\_CONNECT\_OPTIONS  
Tracestufe: 2  
Typ: MQCFIN

### **ConnectionOptionsStructure**

Beschreibung: Die MQCNO-Struktur.  
**Anmerkung:** Nur MQCONNX)  
PCF-Parameter: MQBACF\_MQCNO\_STRUCT  
Tracestufe: 3  
Typ: MQCFBS  
Maximale Länge: Die Länge in Byte der MQCNO-Struktur (die tatsächliche Größe hängt von der Strukturversion ab)

### **ChannelDefinitionStructure**

Beschreibung: Die MQCD-Struktur.  
**Anmerkung:** Nur Clientverbindungen  
PCF-Parameter: MQBACF\_MQCD\_STRUCT  
Tracestufe: 3  
Typ: MQCFBS  
Maximale Länge: Die Länge der MQCD-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

### ***QMGrOpDuration***

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.  <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

### ***MQCTL***

Die Anwendung hat die MQCTL-MQI-Funktion gestartet.

### ***CompCode***

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ:	MQCFIN

### ***Reason***

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ:	MQCFIN

### ***CtlOperation***

Beschreibung:	Einer der MQOP_* -Werte
PCF-Parameter:	MQIACF_CTL_OPERATION
Tracestufe:	1
Typ:	MQCFIN

### ***QMGrOpDuration***

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.  <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

### *MQDISC*

Die Anwendung hat die MQDISC-MQI-Funktion gestartet.

#### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.  
PCF-Parameter: MQIACF\_COMP\_CODE  
Tracestufe: 1  
Typ: MQCFIN

#### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.  
PCF-Parameter: MQIACF\_REASON\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### *MQGET*

Die Anwendung hat die MQGET MQI-Funktion gestartet.

#### **ObjectHandle**

Beschreibung: Die Objektkennung  
PCF-Parameter: MQIACF\_HOBJ  
Tracestufe: 1  
Typ: MQCFIN

#### **GetOptions**

Beschreibung: Abrufen der Optionen von MQGMO.Options  
PCF-Parameter: MQIACF\_GET\_OPTIONS  
Tracestufe: 1  
Typ: MQCFIN

#### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.  
PCF-Parameter: MQIACF\_COMP\_CODE  
Tracestufe: 1  
Typ: MQCFIN

#### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.  
PCF-Parameter: MQIACF\_REASON\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **MsgBuffer**

Beschreibung:	Nachrichtendaten. Wenn TRACEDATA=NONE angegeben wird, wird dieser Parameter nicht angegeben.
PCF-Parameter:	MQBACF_MESSAGE_DATA
Tracestufe:	1
Typ:	MQCFBS
Maximale Länge:	Die Länge wird durch den Parameter TRACEDATA () gesteuert, der in der Konfiguration von APPTRACE festgelegt ist. (In der Tracenachricht wird als MQIACF_TRACE_DATA_LENGTH eingeschlossen).

### **MsgLength**

Beschreibung:	Länge der Nachricht.
PCF-Parameter:	MQIACF_MSG_LENGTH
Tracestufe:	1
Typ:	MQCFIN

### **HighResTime**

Beschreibung:	Zeit der Operation in Mikrosekunden seit Mitternacht, 1. Januar 1970 (UTC) <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert je nach Plattformunterstützung für hohe Auflösungszeitgeber.
PCF-Parameter:	MQIAMO64_HIGHRES_TIME
Tracestufe:	2
Typ:	MQCFIN64

### **BufferLength**

Beschreibung:	Länge des Puffers, der von der Anwendung bereitgestellt wird
PCF-Parameter:	MQIACF_BUFFER_LENGTH
Tracestufe:	2
Typ:	MQCFIN

### **objektname**

Beschreibung:	Der Name des geöffneten Objekts.
PCF-Parameter:	MQCACF_OBJECT_NAME
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_Q_NAME_LENGTH

### **ResolvedQName**

Beschreibung:	Der lokale Name der Warteschlange, aus der die Nachricht abgerufen wurde.
PCF-Parameter:	MQCACF_RESOLVED_Q_NAME
Tracestufe:	2
Typ:	MQCFST

Maximale Länge: MQ\_Q\_NAME\_LENGTH

### **ReportOptions**

Beschreibung: Nachrichtenberichtsoptionen

PCF-Parameter: MQIACF\_REPORT

Tracestufe: 2

Typ: MQCFIN

### **MsgType**

Beschreibung: Typ der Nachricht

PCF-Parameter: MQIACF\_MSG\_TYPE

Tracestufe: 2

Typ: MQCFIN

### **Verfall**

Beschreibung: Nachrichtenlebensdauer

PCF-Parameter: MQIACF\_EXPIRY

Tracestufe: 2

Typ: MQCFIN

### **Format**

Beschreibung: Name des Formats von Nachrichtendaten

PCF-Parameter: MQCACH\_FORMAT\_NAME

Tracestufe: 2

Typ: MQCFST

Maximale Länge: MQ\_FORMAT\_LENGTH

### **Priorität**

Beschreibung: Nachrichtenpriorität

PCF-Parameter: MQIACF\_PRIORITY

Tracestufe: 2

Typ: MQCFIN

### **Persistenz**

Beschreibung: Nachrichtenpersistenz

PCF-Parameter: MQIACF\_PERSISTENCE

Tracestufe: 2

Typ: MQCFIN

### **MsgId**

Beschreibung: Nachrichten-ID

PCF-Parameter: MQBACF\_MSG\_ID

Tracestufe: 2  
Typ: MQCFBS  
Maximale Länge: MQ\_MSG\_ID\_LENGTH

### ***CorrelId***

Beschreibung: Korrelations-ID  
PCF-Parameter: MQBACF\_CORREL\_ID  
Tracestufe: 2  
Typ: MQCFBS  
Maximale Länge: MQ\_CORREL\_ID\_LENGTH

### ***ReplyToQueue***

Beschreibung:  
PCF-Parameter: MQCACF\_REPLY\_TO\_Q  
Tracestufe: 2  
Typ: MQCFST  
Maximale Länge: MQ\_Q\_NAME\_LENGTH

### ***ReplyToQMgr***

Beschreibung:  
PCF-Parameter: MQCACF\_REPLY\_TO\_Q\_MGR  
Tracestufe: 2  
Typ: MQCFST  
Maximale Länge: MQ\_Q\_MGR\_NAME\_LENGTH

### ***CodedCharSetId***

Beschreibung: Zeichensatzkennung von Nachrichtendaten  
PCF-Parameter: MQIA\_CODED\_CHAR\_SET\_ID  
Tracestufe: 2  
Typ: MQCFIN

### ***Encoding***

Beschreibung: Numerische Codierung von Nachrichtendaten.  
PCF-Parameter: MQIACF\_ENCODING  
Tracestufe: 2  
Typ: MQCFIN

### ***PutDate***

Beschreibung:  
PCF-Parameter: MQCACF\_PUT\_DATE  
Tracestufe: 2

Typ: MQCFST  
Maximale Länge: MQ\_PUT\_DATE\_LENGTH

### ***PutTime***

Beschreibung:  
PCF-Parameter: MQCACF\_PUT\_TIME  
Tracestufe: 2  
Typ: MQCFST  
Maximale Länge: MQ\_PUT\_TIME\_LENGTH

### ***ResolvedQName***

Beschreibung: Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT\_Q ist.  
PCF-Parameter: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH.

### ***ResObjectString***

Beschreibung: Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT\_TOPIC lautet.  
PCF-Parameter: MQCACF\_RESOLVED\_OBJECT\_STRING  
Tracestufe: 2  
Typ: MQCFST  
Länge: Länge variiert.

### ***ResolvedType***

Beschreibung: Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT\_Q', 'MQOT\_TOPIC' oder 'MQOT\_NONE'.  
PCF-Parameter: MQIACF\_RESOLVED\_TYPE  
Tracestufe: 2  
Typ: MQCFIN

### ***PolicyName***

Beschreibung: Der Richtlinienname, der auf diese Nachricht angewendet wurde.  
**Anmerkung:** Nur AMS-geschützte Nachrichten  
PCF-Parameter: MQCA\_POLICY\_NAME  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_OBJECT\_NAME\_LENGTH

### ***XmitqMsgId***

Beschreibung: Die Nachrichten-ID der Nachricht im Header der Übertragungswarteschlange.  
**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQBACF\_XQH\_MSG\_ID

Tracestufe: 2

Typ: MQCFBS

Länge: MQ\_MSG\_ID\_LENGTH

### ***XmitqCorrelId***

Beschreibung: Die Korrelations-ID der Nachricht im Header der Übertragungswarteschlange.  
**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQBACF\_XQH\_CORREL\_ID

Tracestufe: 2

Typ: MQCFBS

Länge: MQ\_CORREL\_ID\_LENGTH

### ***XmitqPutTime***

Beschreibung: Die Zeitzeit der Nachricht im Header der Übertragungswarteschlange.  
**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_PUT\_TIME

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_PUT\_TIME\_LENGTH

### ***XmitqPutDate***

Beschreibung: Das Datum, an dem die Nachricht in den Header der Übertragungswarteschlange gestellt wurde.  
**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_PUT\_DATE

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_PUT\_DATE\_LENGTH

### ***XmitqRemoteQName***

Beschreibung: Das Ziel der fernen Warteschlange der Nachricht im Header der Übertragungswarteschlange.  
**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_REMOTE\_Q\_NAME

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_Q\_NAME\_LENGTH

### ***XmitqRemoteQMgr***

Beschreibung: Die Zieladresse des fernen Warteschlangenmanagers der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_REMOTE\_Q\_MGR

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_Q\_NAME\_LENGTH

### ***MsgDescStructure***

Beschreibung: Die MQMD-Struktur.

PCF-Parameter: MQBACF\_MQMD\_STRUCT

Tracestufe: 3

Typ: MQCFBS

Maximale Länge: Die Länge der MQMD-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

### ***GetMsgOptsStructure***

Beschreibung: Die MQGMO-Struktur.

PCF-Parameter: MQBACF\_MQGMO\_STRUCT

Tracestufe: 3

Typ: MQCFBS

Maximale Länge: Die Länge der MQGMO-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

### ***QMGrOpDuration***

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbraucht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64\_QMGR\_OP\_DURATION

Tracestufe: 2

Typ: MQCFIN64

### ***MQINQ***

Die Anwendung hat die MQINQ-MQI-Funktion gestartet.

### ***ObjectHandle***

Beschreibung: Die Objektkennung

PCF-Parameter: MQIACF\_HOBJ  
Tracestufe: 1  
Typ: MQCFIN

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.  
PCF-Parameter: MQIACF\_COMP\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.  
PCF-Parameter: MQIACF\_REASON\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **SelectorCount**

Beschreibung: Die Anzahl der Selektoren, die in der Selektorenfeldgruppe bereitgestellt werden.  
PCF-Parameter: MQIACF\_SELECTOR\_COUNT  
Tracestufe: 2  
Typ: MQCFIN

### **Selectors**

Beschreibung: Die Liste der Attribute (Ganzzahl oder Zeichen), deren Werte von MQINQ zurückgegeben werden müssen.  
PCF-Parameter: MQIACF\_SELECTORS  
Tracestufe: 2  
Typ: MQCFIL

### **ResolvedQName**

Beschreibung: Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT\_Q ist.  
PCF-Parameter: MQCACF\_RESOLVED\_Q\_NAME  
Tracestufe: 2  
Typ: MQCFST  
Maximale Länge: MQ\_Q\_NAME\_LENGTH

### **ResObjectString**

Beschreibung: Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT\_TOPIC lautet.  
PCF-Parameter: MQCACF\_RESOLVED\_OBJECT\_STRING  
Tracestufe: 2

Typ: MQCFST  
Maximale Länge: Länge variiert

### ***ResolvedType***

Beschreibung: Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT\_Q', 'MQOT\_TOPIC' oder 'MQOT\_NONE'.  
PCF-Parameter: MQIACF\_RESOLVED\_TYPE  
Tracestufe: 2  
Typ: MQCFIN

### ***IntAttrCount***

Beschreibung: Die Anzahl der ganzzahligen Attribute, die von der Inquire-Operation zurückgegeben werden.  
PCF-Parameter: MQIACF\_INTATTR\_COUNT  
Tracestufe: 3  
Typ: MQCFIN

### ***IntAttr***

Beschreibung: Die ganzzahligen Attributwerte, die von der Inquire-Operation zurückgegeben werden. Dieser Parameter ist nur enthalten, wenn 'IntAttrCount' bei der Rückgabe von MQINQ größer als 0 ist.  
PCF-Parameter: MQIACF\_INT\_ATTRS  
Tracestufe: 3  
Typ: MQCFIL

### ***CharAttr***

Beschreibung: Die Zeichenattribute, die von der Inquire-Operation zurückgegeben werden. Die Werte werden miteinander verkettet. Dieser Parameter ist nur enthalten, wenn 'CharAttrLength' bei der Rückgabe von MQINQ größer als 0 ist.  
PCF-Parameter: MQCACF\_CHAR\_ATTRS  
Tracestufe: 3  
Typ: MQCFST

### ***QMGrOpDuration***

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  
Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64\_QMGR\_OP\_DURATION  
Tracestufe: 2  
Typ: MQCFIN64

## *MQOPEN*

Die Anwendung hat die MQOPEN-MQI-Funktion gestartet.

### **ObjectType**

Beschreibung:	Der in MQOT.ObjectType übergebene Objekttyp
PCF-Parameter:	MQIACF_OBJECT_TYPE
Tracestufe:	1
Typ:	MQCFIN

### **objektname**

Beschreibung:	Der Name des Objekts, das an den MQI-Aufruf übergeben wurde, bevor eine beliebige Warteschlangennamensauflösung versucht wird.
PCF-Parameter:	MQCACF_OBJECT_NAME
Tracestufe:	1
Typ:	MQCFST
Maximale Länge:	MQ_Q_NAME_LENGTH

### **ObjectQMgrName**

Beschreibung:	Der Name des Objektwarteschlangenmanagers, der an den MQI-Aufruf übergeben wurde, bevor eine beliebige Warteschlangennamensauflösung versucht wird.
PCF-Parameter:	MQCACF_OBJECT_Q_MGR_NAME
Tracestufe:	1
Typ:	MQCFST
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH

### **ObjectHandle**

Beschreibung:	Die Objektkennung
PCF-Parameter:	MQIACF_HOBJ
Tracestufe:	1
Typ:	MQCFIN

### **CompCode**

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ:	MQCFIN

### **Reason**

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ:	MQCFIN

### **OpenOptions**

Beschreibung:	Optionen zum Öffnen des Objekts
PCF-Parameter:	MQIACF_OPEN_OPTIONS
Tracestufe:	1
Typ:	MQCFIN

### **AlternateUserId**

Beschreibung:	Wird nur eingeschlossen, wenn MQOO_ALTERNATE_USER_AUTHORITY angegeben ist
PCF-Parameter:	MQCACF_ALTERNATE_USERID
Tracestufe:	2
Typ:	MQCFST
Maximale Länge:	MQ_USER_ID_LENGTH

### **RecsPresent**

Beschreibung:	Die Anzahl der vorhandenen Objektnamensdatensätze. Nur enthalten, wenn MQOD Version $\geq$ MQOD_VERSION_2
PCF-Parameter:	MQIACF_RECS_PRESENT
Tracestufe:	1
Typ:	MQCFIN

### **KnownDestCount**

Beschreibung:	Die Anzahl der erfolgreich geöffneten lokalen Warteschlangen. Nur enthalten, wenn MQOD Version $\geq$ MQOD_VERSION_2
PCF-Parameter:	MQIACF_KNOWN_DEST_COUNT
Tracestufe:	1
Typ:	MQCFIN

### **UnknownDestCount**

Beschreibung:	Die Anzahl der erfolgreich geöffneten fernen Warteschlangen. Nur enthalten, wenn MQOD Version $\geq$ MQOD_VERSION_2
PCF-Parameter:	MQIACF_UNKNOWN_DEST_COUNT
Tracestufe:	1
Typ:	MQCFIN

### **InvalidDestCount**

Beschreibung:	Die Anzahl der Warteschlangen, die nicht geöffnet werden konnten. Nur enthalten, wenn MQOD Version $\geq$ MQOD_VERSION_2
PCF-Parameter:	MQIACF_INVALID_DEST_COUNT
Tracestufe:	1
Typ:	MQCFIN

### ***DynamicQName***

Beschreibung:	Der Name der dynamischen Warteschlange, die als Eingabe an den MQOPEN-Aufruf übergeben wurde.
PCF-Parameter:	MQCACF_DYNAMIC_Q_NAME
Tracestufe:	2
Typ:	MQCFST
Maximale Länge:	MQ_Q_NAME_LENGTH

### ***ResolvedLocalQName*** 1 2

Beschreibung:	Enthält den Namen der lokalen Warteschlange, nachdem die Namensauflösung stattgefunden hat. (Bei fernen Warteschlangen ist dies beispielsweise der Name der Übertragungswarteschlange.)
PCF-Parameter:	MQCACF_RESOLVED_LOCAL_Q_NAME
Tracestufe:	2
Typ:	MQCFST
Bereich:	Wenn MQOD.Version kleiner als MQOD_VERSION_3 ist, enthält dies den Wert des Feldes MQOD.ObjectName, nachdem der MQOPEN-Aufruf abgeschlossen wurde. Wenn MQOD.Version gleich oder größer als MQOD_VERSION_3 ist, enthält dies den Wert in der MQOD-Datei. Feld "ResolvedQName".
Maximale Länge:	MQ_Q_NAME_LENGTH

### ***ResolvedLocalQMgrName*** 1 2

Beschreibung:	Der Name des lokalen WS-Managers nach der Namensauflösung wurde ausgeführt.
PCF-Parameter:	MQCACF_RESOLVED_LOCAL_Q_MGR
Tracestufe:	2
Typ:	MQCFST
Bereich:	Nur bei MQOD.Version >= MQOD_VERSION_3
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH

### ***ResolvedQName*** 1 2

Beschreibung:	Der Warteschlangennamenname nach der Namensauflösung wurde ausgeführt.
PCF-Parameter:	MQCACF_RESOLVED_Q_NAME
Tracestufe:	2
Typ:	MQCFST
Bereich:	Wenn MQOD.Version kleiner als MQOD_VERSION_3 ist, enthält dies den Wert des Feldes MQOD.ObjectName, nachdem der MQOPEN-Aufruf abgeschlossen wurde. Wenn MQOD.Version gleich oder größer als MQOD_VERSION_3 ist, enthält dies den Wert in der MQOD-Datei. Feld "ResolvedQName".
Maximale Länge:	MQ_Q_NAME_LENGTH

### **ResolvedQMgrName** <sup>1</sup> <sup>2</sup>

Beschreibung:	Enthält den Namen des Warteschlangenmanagers, nachdem die Namensauflösung ausgeführt wurde. Wenn MQOD.Version kleiner als MQOD_VERSION_3 ist, enthält dies den Wert von MQOD. Feld "ObjectQMgrName", nachdem der MQOPEN-Aufruf abgeschlossen wurde. Wenn MQOD.Version gleich oder größer als MQOD_VERSION_3 ist, enthält dies den Wert in der MQOD-Datei. Das Feld "ResolvedQMgrName".
PCF-Parameter:	MQCACF_RESOLVED_Q_MGR
Tracestufe:	2
Typ:	MQCFST
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH

### **AlternateSecurityId**

Beschreibung:	Alternative Sicherheitskennung. Nur vorhanden, wenn MQOD.Version gleich oder größer als MQOD_VERSION_3 ist, MQOO_ALTERNATE_USER_AUTHORITY angegeben ist und MQOD.AlternateSecurityId nicht gleich MQSID_NONE ist.
PCF-Parameter:	MQBACF_ALTERNATE_SECURITYID
Tracestufe:	2
Typ:	MQCFBS
Maximale Länge:	MQ_SECURITY_ID_LENGTH

### **ObjectString**

Beschreibung:	Langer Objektname. Nur wenn MQOD.Version gleich oder größer als MQOD_VERSION_4 ist und das Feld VSLength von MQOD.ObjectString MQVS_NULL_TERMINATED oder größer als null ist.
PCF-Parameter:	MQCACF_OBJECT_STRING
Tracestufe:	2
Typ:	MQCFST
Maximale Länge:	Länge variiert.

### **SelectionString**

Beschreibung:	Auswahlzeichenfolge. Wird nur eingeschlossen, wenn MQOD.Version gleich oder größer als MQOD_VERSION_4 und das Feld für die VSLength von MQOD ist. SelectionString ist MQVS_NULL_TERMINATED oder größer als null.
PCF-Parameter:	MQCACF_SELECTION_STRING
Tracestufe:	2
Typ:	MQCFST
Maximale Länge:	Länge variiert.

### **ResObjectString**

Beschreibung:	Der lange Objektname, nachdem der WS-Manager den im Feld ObjectName angegebenen Namen aufgelöst hat. Nur für Themen und Warteschlangenaliasnamen, die auf ein Themenobjekt verweisen, wenn MQOD.Version gleich oder größer als MQOD_VERSION_4 und VSLength ist, ist MQVS_NULL_TERMINATED oder größer als null.
---------------	--

PCF-Parameter: MQCACF\_RESOLVED\_OBJECT\_STRING  
 Tracestufe: 2  
 Typ: MQCFST  
 Maximale Länge: Länge variiert.

### **ResolvedType**

Beschreibung: Der Typ des aufgelösten Objekts (Basisobjekts), das geöffnet wird. Wird nur eingeschlossen, wenn MQOD.Version gleich oder größer als MQOD\_VERSION\_4 ist. Mögliche Werte sind 'MQOT\_Q', 'MQOT\_TOPIC' oder 'MQOT\_NONE'.

PCF-Parameter: MQIACF\_RESOLVED\_TYPE  
 Tracestufe: 2  
 Typ: MQCFIN

### **QMGrOpDuration**

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  
 Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64\_QMGR\_OP\_DURATION  
 Tracestufe: 2  
 Typ: MQCFIN64

### *PCF-Gruppenheader der Verteilerliste der Anwendungsaktivität*

Wenn die Funktion MQOPEN eine Verteilerliste öffnet, enthält die MQOPEN-Parameter eine AppActivityDistList-PCF-Gruppe für jede der Warteschlangen in der Verteilerliste bis zur Anzahl der Strukturen, die in RecsPresent nummeriert sind. Die PCF-Gruppe "Ap-pActivityDistList" kombiniert Informationen aus den MQOR- und MQRR-Strukturen, um den Namen der Warteschlange zu identifizieren, und zeigt das Ergebnis der offenen Operation in der Warteschlange an. Eine AppActivityDistList-Gruppe beginnt immer mit der folgenden MQCFGR-Struktur:

<i>Tabelle 29. AppActivityDistList-Gruppe MQCFGR-Struktur</i>		
<b>MQCFGR, Feld</b>	<b>Value</b>	<b>Beschreibung</b>
Typ	MQCFT_GROUP	
StrucLength	Länge der MQCFGR-Struktur in Byte	
Parameter	MQGACF_APP_DIST_LIST	Verteilerlistengruppenparameter

<sup>1</sup> Dieser Parameter ist nur enthalten, wenn das Objekt, das geöffnet wird, in eine Warteschlange aufgelöst wird und die Warteschlange für MQOO\_INPUT\_\*, MQOO\_OUTPUT oder MQOO\_BROWSE geöffnet wird.

<sup>2</sup> Der Parameter "ResolvedLocalQName" ist nur enthalten, wenn er sich vom Parameter "ResolvedQName" unterscheidet.

Tabelle 29. AppActivityDistList-Gruppe MQCFGR-Struktur (Forts.)

MQCFGR, Feld	Value	Beschreibung
Parameteranzahl	4	Die Anzahl der Parameterstrukturen im Anschluss an die MQCFGR-Struktur, die in dieser Gruppe enthalten sind.

### **objektname**

Beschreibung: Der Name einer Warteschlange in der Verteilerliste 'MQ\_Q\_NAME\_LENGTH'. Wird nur eingeschlossen, wenn MQOR-Strukturen bereitgestellt werden.

PCF-Parameter: MQCACF\_OBJECT\_NAME

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_Q\_NAME\_LENGTH. Wird nur eingeschlossen, wenn MQOR-Strukturen bereitgestellt werden.

### **ObjectQMgrName**

Beschreibung: Der Name des Warteschlangenmanagers, auf dem die in Objektname angegebene Warteschlange definiert ist.

PCF-Parameter: MQCACF\_OBJECT\_Q\_MGR\_NAME

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_Q\_MGR\_NAME\_LENGTH. Wird nur eingeschlossen, wenn MQOR-Strukturen bereitgestellt werden.

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis des geöffneten Objekts für dieses Objekt anzeigt. Nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für den MQOPEN-Befehl MQRC\_MULTIPLE\_REASONS ist

PCF-Parameter: MQIACF\_COMP\_CODE

Tracestufe: 2

Typ: MQCFIN

### **Reason**

Beschreibung: Der Ursachencode, der das Ergebnis des offenen Objekts für dieses Objekt angibt. Nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für den MQOPEN-Befehl MQRC\_MULTIPLE\_REASONS ist

PCF-Parameter: MQIACF\_REASON\_CODE

Tracestufe: 2

Typ: MQCFIN

### **MQPUT**

Die Anwendung hat die MQPUT-MQI-Funktion gestartet.

### **ObjectHandle**

Beschreibung: Die Objektkennung

PCF-Parameter: MQIACF\_HOBJ  
Tracestufe: 1  
Typ: MQCFIN

### **PutOptions**

Beschreibung: Die Optionen 'put' aus 'MQPMO.Options'  
PCF-Parameter: MQIACF\_PUT\_OPTIONS  
Tracestufe: 1  
Typ: MQCFIN

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.  
PCF-Parameter: MQIACF\_COMP\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.  
PCF-Parameter: MQIACF\_REASON\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **MsgBuffer**

Beschreibung: Nachrichtendaten.  
PCF-Parameter: MQBACF\_MESSAGE\_DATA  
Tracestufe: 1  
Typ: MQCFBS  
Länge: Die Länge wird durch den Parameter TRACEDATA () gesteuert, der in der Konfiguration von APPTRACE festgelegt ist. Wenn TRACEDATA=NONE angegeben wird, wird dieser Parameter weggelassen.

### **MsgLength**

Beschreibung: Länge der Nachricht.  
PCF-Parameter: MQIACF\_MSG\_LENGTH  
Tracestufe: 1  
Typ: MQCFIN

### **RecsPresent**

Beschreibung: Die Anzahl der vorhandenen Nachrichtendatensätze oder Antwortdatensätze. Nur enthalten, wenn MQPMO Version >= MQPMO\_VERSION\_2  
PCF-Parameter: MQIACF\_RECS\_PRESENT  
Tracestufe: 1

Typ: MQCFIN

### ***KnownDestCount***

Beschreibung: Anzahl der Nachrichten, die erfolgreich an lokale Warteschlangen gesendet wurden

PCF-Parameter: MQIACF\_KNOWN\_DEST\_COUNT

Tracestufe: 1

Typ: MQCFIN

### ***UnknownDestCount***

Beschreibung: Anzahl der Nachrichten, die erfolgreich an ferne Warteschlangen gesendet wurden

PCF-Parameter: MQIACF\_UNKNOWN\_DEST\_COUNT

Tracestufe: 1

Typ: MQCFIN

### ***InvalidDestCount***

Beschreibung: Anzahl der Nachrichten, die nicht gesendet werden konnten

PCF-Parameter: MQIACF\_INVALID\_DEST\_COUNT

Tracestufe: 1

Typ: MQCFIN

### ***HighResTime***

Beschreibung: Zeit der Operation in Mikrosekunden seit Mitternacht, 1. Januar 1970 (UTC)

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert je nach Plattformunterstützung für hohe Auflösungszeitgeber.

PCF-Parameter: MQIAMO64\_HIGHRES\_TIME

Tracestufe: 2

Typ: MQCFIN64

### ***objektname***

Beschreibung: Der Name des geöffneten Objekts.

PCF-Parameter: MQCACF\_OBJECT\_NAME

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_Q\_NAME\_LENGTH

### ***ResolvedQName***

Beschreibung: Der Name der Warteschlange, nachdem die Warteschlangennamensauflösung ausgeführt wurde.

PCF-Parameter: MQCACF\_RESOLVED\_Q\_NAME

Tracestufe: 2

Typ: MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH

### ***ResolvedQMgrName***

Beschreibung: Der Name des WS-Managers nach der Namensauflösung wurde ausgeführt.  
PCF-Parameter: MQCACF\_RESOLVED\_Q\_MGR  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_Q\_MGR\_NAME\_LENGTH

### ***ResolvedLocalQName*** <sup>3</sup>

Beschreibung: Enthält den Namen der lokalen Warteschlange, nachdem die Namensauflösung ausgeführt wurde.  
PCF-Parameter: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
Tracestufe: 2  
Typ: MQCFST

### ***ResolvedLocalQMgrName*** <sup>3</sup>

Beschreibung: Enthält den Namen des lokalen Warteschlangenmanagers, nachdem die Namensauflösung ausgeführt wurde.  
PCF-Parameter: MQCACF\_RESOLVED\_LOCAL\_Q\_MGR  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_Q\_MGR\_NAME\_LENGTH

### ***ReportOptions***

Beschreibung: Nachrichtenberichtsoptionen  
PCF-Parameter: MQIACF\_REPORT  
Tracestufe: 2  
Typ: MQCFIN

### ***MsgType***

Beschreibung: Typ der Nachricht  
PCF-Parameter: MQIACF\_MSG\_TYPE  
Tracestufe: 2  
Typ: MQCFIN

### ***Verfall***

Beschreibung: Nachrichtenlebensdauer  
PCF-Parameter: MQIACF\_EXPIRY  
Tracestufe: 2  
Typ: MQCFIN

### **Format**

Beschreibung:	Name des Formats von Nachrichtendaten
PCF-Parameter:	MQCACH_FORMAT_NAME
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_FORMAT_LENGTH

### **Priorität**

Beschreibung:	Nachrichtenpriorität
PCF-Parameter:	MQIACF_PRIORITY
Tracestufe:	2
Typ:	MQCFIN

### **Persistenz**

Beschreibung:	Nachrichtenpersistenz
PCF-Parameter:	MQIACF_PERSISTENCE
Tracestufe:	2
Typ:	MQCFIN

### **MsgId**

Beschreibung:	Nachrichten-ID
PCF-Parameter:	MQBACF_MSG_ID
Tracestufe:	2
Typ:	MQCFBS
Länge:	MQ_MSG_ID_LENGTH

### **CorrelId**

Beschreibung:	Korrelations-ID
PCF-Parameter:	MQBACF_CORREL_ID
Tracestufe:	2
Typ:	MQCFBS
Länge:	MQ_CORREL_ID_LENGTH

### **ReplyToQueue**

Beschreibung:	
PCF-Parameter:	MQCACF_REPLY_TO_Q
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_Q_NAME_LENGTH

### ***ReplyToQMgr***

Beschreibung:  
PCF-Parameter: MQCACF\_REPLY\_TO\_Q\_MGR  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_Q\_MGR\_NAME\_LENGTH

### ***CodedCharSetId***

Beschreibung: Zeichensatzkennung von Nachrichtendaten  
PCF-Parameter: MQIA\_CODED\_CHAR\_SET\_ID  
Tracestufe: 2  
Typ: MQCFIN

### ***Encoding***

Beschreibung: Numerische Codierung von Nachrichtendaten.  
PCF-Parameter: MQIACF\_ENCODING  
Tracestufe: 2  
Typ: MQCFIN

### ***PutDate***

Beschreibung:  
PCF-Parameter: MQCACF\_PUT\_DATE  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_PUT\_DATE\_LENGTH

### ***PutTime***

Beschreibung:  
PCF-Parameter: MQCACF\_PUT\_TIME  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_PUT\_TIME\_LENGTH

### ***ResolvedQName***

Beschreibung: Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT\_Q ist.  
PCF-Parameter: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH.

### ***ResObjectString***

Beschreibung:	Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT_TOPIC lautet.
PCF-Parameter:	MQCACF_RESOLVED_OBJECT_STRING
Tracestufe:	2
Typ:	MQCFST
Länge:	Länge variiert.

### ***ResolvedType***

Beschreibung:	Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT_Q', 'MQOT_TOPIC' oder 'MQOT_NONE'.
PCF-Parameter:	MQIACF_RESOLVED_TYPE
Tracestufe:	2
Typ:	MQCFIN

### ***PolicyName***

Beschreibung:	Der Richtlinienname, der auf diese Nachricht angewendet wurde. <b>Anmerkung:</b> Nur AMS-geschützte Nachrichten
PCF-Parameter:	MQCA_POLICY_NAME
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_OBJECT_NAME_LENGTH

### ***XmitqMsgId***

Beschreibung:	Die Nachrichten-ID der Nachricht im Header der Übertragungswarteschlange. <b>Anmerkung:</b> Nur bei Format: MQFMT_XMIT_Q_HEADER
PCF-Parameter:	MQBACF_XQH_MSG_ID
Tracestufe:	2
Typ:	MQCFBS
Länge:	MQ_MSG_ID_LENGTH

### ***XmitqCorrelId***

Beschreibung:	Die Korrelations-ID der Nachricht im Header der Übertragungswarteschlange. <b>Anmerkung:</b> Nur bei Format: MQFMT_XMIT_Q_HEADER
PCF-Parameter:	MQBACF_XQH_CORREL_ID
Tracestufe:	2
Typ:	MQCFBS
Länge:	MQ_CORREL_ID_LENGTH

### ***XmitqPutTime***

Beschreibung: Die Zeitzeit der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_PUT\_TIME

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_PUT\_TIME\_LENGTH

### ***XmitqPutDate***

Beschreibung: Das Datum, an dem die Nachricht in den Header der Übertragungswarteschlange gestellt wurde.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_PUT\_DATE

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_PUT\_DATE\_LENGTH

### ***XmitqRemoteQName***

Beschreibung: Das Ziel der fernen Warteschlange der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_REMOTE\_Q\_NAME

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_Q\_NAME\_LENGTH

### ***XmitqRemoteQMgr***

Beschreibung: Die Zieladresse des fernen Warteschlangenmanagers der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_REMOTE\_Q\_MGR

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_Q\_NAME\_LENGTH

### ***PutMsgOptsStructure***

Beschreibung: Die MQPMO-Struktur.

PCF-Parameter: MQBACF\_MQPMO\_STRUCT

Tracestufe: 3

Typ: MQCFBS

Länge: Die Länge der MQPMO-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

### **QMGrOpDuration**

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  
Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbraucht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64\_QMGR\_OP\_DURATION

Tracestufe: 2

Typ: MQCFIN64

### *MQPUT Application Activity Distribution List PCF-Gruppen-Header-Struktur*

Wenn die MQPUT-Funktion in eine Verteilerliste gestellt wird, enthalten die MQPUT-Parameter eine AppActivityDistList-PCF-Gruppe. Informationen zu den einzelnen Warteschlangen in der Verteilerliste finden Sie in „PCF-Gruppenheader der Verteilerliste der Anwendungsaktivität“ auf Seite 261. Die PCF-Gruppe AppActivityDistList kombiniert Informationen aus den MQPMR- und MQRR-Strukturen, um die PUT-Parameter zu identifizieren, und geben das Ergebnis der PUT-Operation für jede Warteschlange an. Für MQPUT-Operationen enthält die Gruppe AppActivityDistList einige oder alle der folgenden Parameter (der Code und der Ursachencode sind vorhanden, wenn der Ursachencode MQRC\_MULTIPLE\_REASONS lautet und die anderen Parameter durch das Feld MQPMO.PutMsgRecFields bestimmt werden):

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt. Wird nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für den MQPUT MQRC\_MULTIPLE\_REASONS ist

PCF-Parameter: MQIACF\_COMP\_CODE

Tracestufe: 2

Typ: MQCFIN

### **Reason**

Beschreibung: Der Ursachencode, der das Ergebnis der Einordnung für dieses Objekt angibt. Wird nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für den MQPUT MQRC\_MULTIPLE\_REASONS ist

PCF-Parameter: MQIACF\_REASON\_CODE

Tracestufe: 2

Typ: MQCFIN

---

<sup>3</sup> Der Parameter "ResolvedLocalQName" ist nur enthalten, wenn er sich vom Parameter "ResolvedQName" unterscheidet.

### **MsgId**

Beschreibung:	Nachrichten-ID. Nur eingeschlossen, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF_MSG_ID
PCF-Parameter:	MQBACF_MSG_ID
Tracestufe:	2
Typ:	MQCFBS
Länge:	MQ_MSG_ID_LENGTH

### **CorrelId**

Beschreibung:	Korrelations-ID Wird nur eingefügt, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF_CORREL_ID
PCF-Parameter:	MQBACF_CORREL_ID
Tracestufe:	2
Typ:	MQCFBS
Länge:	MQ_CORREL_ID_LENGTH

### **GroupId**

Beschreibung:	Gruppen-ID. Nur eingeschlossen, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF_GROUP_ID
PCF-Parameter:	MQBACF_GROUP_ID
Tracestufe:	2
Typ:	MQCFBS
Länge:	MQ_GROUP_ID_LENGTH

### **Feedback**

Beschreibung:	Rückkopplungscode. Nur eingeschlossen, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF_FEEDBACK
PCF-Parameter:	MQIACF_FEEDBACK
Tracestufe:	2
Typ:	MQCFIN

### **AccountingToken**

Beschreibung:	AccountingToken. Nur eingeschlossen, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF_ACCOUNTING_TOKEN
PCF-Parameter:	MQBACF_ACCOUNTING_TOKEN
Tracestufe:	2
Typ:	MQCFBS
Länge:	MQ_ACCOUNTING_TOKEN_LENGTH.

### **MQPUT1**

Die Anwendung hat die MQPUT1-MQI-Funktion gestartet.

### **ObjectType**

Beschreibung:	Der in MQOT.ObjectType übergebene Objekttyp
---------------	---

PCF-Parameter: MQIACF\_OBJECT\_TYPE  
Tracestufe: 1  
Typ: MQCFIN

### **objektname**

Beschreibung: Der Name des Objekts, das an den MQI-Aufruf übergeben wurde, bevor eine beliebige Warteschlangennamensauflösung versucht wird.  
PCF-Parameter: MQCACF\_OBJECT\_NAME  
Tracestufe: 1  
Typ: MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH

### **ObjectQMgrName**

Beschreibung: Der Name des Objektwarteschlangenmanagers, der an den MQI-Aufruf übergeben wurde, bevor eine beliebige Warteschlangennamensauflösung versucht wird.  
PCF-Parameter: MQCACF\_OBJECT\_Q\_MGR\_NAME  
Tracestufe: 1  
Typ: MQCFST  
Länge: MQ\_Q\_MGR\_NAME\_LENGTH

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.  
PCF-Parameter: MQIACF\_COMP\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.  
PCF-Parameter: MQIACF\_REASON\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **PutOptions**

Beschreibung: Die Optionen 'put' aus 'MQPMO.Options'  
PCF-Parameter: MQIACF\_PUT\_OPTIONS  
Tracestufe: 1  
Typ: MQCFIN

### **AlternateUserId**

Beschreibung: Wird nur eingeschlossen, wenn MQPMO\_ALTERNATE\_USER\_AUTHORITY angegeben ist.  
PCF-Parameter: MQCACF\_ALTERNATE\_USERID

Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_USER\_ID\_LENGTH

### ***RecsPresent***

Beschreibung: Die Anzahl der vorhandenen Objektnamensdatensätze.  
PCF-Parameter: MQIACF\_RECS\_PRESENT  
Tracestufe: 1  
Typ: MQCFIN

### ***KnownDestCount***

Beschreibung: Anzahl der lokalen Warteschlangen, die erfolgreich geöffnet wurden  
PCF-Parameter: MQIACF\_KNOWN\_DEST\_COUNT  
Tracestufe: 1  
Typ: MQCFIN

### ***UnknownDestCount***

Beschreibung: Anzahl der fernen Warteschlangen, die erfolgreich geöffnet wurden  
PCF-Parameter: MQIACF\_UNKNOWN\_DEST\_COUNT  
Tracestufe: 1  
Typ: MQCFIN

### ***InvalidDestCount***

Beschreibung: Anzahl der Warteschlangen, die nicht geöffnet werden konnten  
PCF-Parameter: MQIACF\_INVALID\_DEST\_COUNT  
Tracestufe: 1  
Typ: MQCFIN

### ***MsgBuffer***

Beschreibung: Nachrichtendaten.  
PCF-Parameter: MQBACF\_MESSAGE\_DATA  
Tracestufe: 1  
Typ: MQCFBS  
Länge: Die Länge wird durch den Parameter TRACEDATA () gesteuert, der in der Konfiguration von APPTRACE festgelegt ist. Wenn TRACEDATA=NONE angegeben wird, wird dieser Parameter weggelassen.

### ***MsgLength***

Beschreibung: Länge der Nachricht.  
PCF-Parameter: MQIACF\_MSG\_LENGTH  
Tracestufe: 1  
Typ: MQCFIN

### **HighResTime**

Beschreibung:	Zeit der Operation in Mikrosekunden seit Mitternacht, 1. Januar 1970 (UTC) <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert je nach Plattformunterstützung für hohe Auflösungszeitgeber.
PCF-Parameter:	MQIAMO64_HIGHRES_TIME
Tracestufe:	2
Typ:	MQCFIN64

### **ResolvedQName**

Beschreibung:	Der Name der Warteschlange, nachdem die Warteschlangennamensauflösung ausgeführt wurde.
PCF-Parameter:	MQCACF_RESOLVED_Q_NAME
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_Q_NAME_LENGTH

### **ResolvedQMgrName**

Beschreibung:	Der Name des WS-Managers nach der Namensauflösung wurde ausgeführt.
PCF-Parameter:	MQCACF_RESOLVED_Q_MGR
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_Q_MGR_NAME_LENGTH

### **ResolvedLocalQName** <sup>4</sup>

Beschreibung:	Enthält den Namen der lokalen Warteschlange, nachdem die Namensauflösung ausgeführt wurde.
PCF-Parameter:	MQCACF_RESOLVED_LOCAL_Q_NAME
Tracestufe:	2
Typ:	MQCFST

### **ResolvedLocalQMgrName** <sup>4</sup>

Beschreibung:	Enthält den Namen des lokalen Warteschlangenmanagers, nachdem die Namensauflösung ausgeführt wurde.
PCF-Parameter:	MQCACF_RESOLVED_LOCAL_Q_MGR
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_Q_MGR_NAME_LENGTH

### **AlternateSecurityId**

Beschreibung:	Alternative Sicherheits-ID. Nur vorhanden, wenn MQOD.Version gleich oder größer als MQOD_VERSION_3 ist und MQOD.AlternateSecurityId nicht mit MQSID_NONE identisch ist.
PCF-Parameter:	MQBACF_ALTERNATE_SECURITYID

Tracestufe: 2  
Typ: MQCFBS  
Länge: MQ\_SECURITY\_ID\_LENGTH

### **ObjectString**

Beschreibung: Langer Objektname. Nur wenn MQOD.Version gleich oder größer als MQOD\_VERSION\_4 ist und das Feld VSLength von MQOD.ObjectString MQVS\_NULL\_TERMINATED oder größer als null ist.

PCF-Parameter: MQCACF\_OBJECT\_STRING

Tracestufe: 2

Typ: MQCFST

Länge: Länge variiert.

### **ResObjectString**

Beschreibung: Der lange Objektname, nachdem der WS-Manager den im Feld ObjectName angegebenen Namen aufgelöst hat. Nur für Themen und Warteschlangenaliasnamen, die auf ein Themenobjekt verweisen, wenn MQOD.Version gleich oder größer als MQOD\_VERSION\_4 und VSLength ist, ist MQVS\_NULL\_TERMINATED oder größer als null.

PCF-Parameter: MQCACF\_RESOLVED\_OBJECT\_STRING

Tracestufe: 2

Typ: MQCFST

Länge: Länge variiert.

### **ResolvedType**

Beschreibung: Der Typ des aufgelösten Objekts (Basisobjekts), das geöffnet wird. Wird nur eingeschlossen, wenn MQOD.Version gleich oder größer als MQOD\_VERSION\_4 ist. Mögliche Werte sind 'MQOT\_Q', 'MQOT\_TOPIC' oder 'MQOT\_NONE'.

PCF-Parameter: MQIACF\_RESOLVED\_TYPE

Tracestufe: 2

Typ: MQCFIN

### **ReportOptions**

Beschreibung: Nachrichtenberichtsoptionen

PCF-Parameter: MQIACF\_REPORT

Tracestufe: 2

Typ: MQCFIN

### **MsgType**

Beschreibung: Typ der Nachricht

PCF-Parameter: MQIACF\_MSG\_TYPE

Tracestufe: 2

Typ: MQCFIN

### **Verfall**

Beschreibung: Nachrichtenlebensdauer  
PCF-Parameter: MQIACF\_EXPIRY  
Tracestufe: 2  
Typ: MQCFIN

### **Format**

Beschreibung: Name des Formats von Nachrichtendaten  
PCF-Parameter: MQCACH\_FORMAT\_NAME  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_FORMAT\_LENGTH

### **Priorität**

Beschreibung: Nachrichtenpriorität  
PCF-Parameter: MQIACF\_PRIORITY  
Tracestufe: 2  
Typ: MQCFIN

### **Persistenz**

Beschreibung: Nachrichtenpersistenz  
PCF-Parameter: MQIACF\_PERSISTENCE  
Tracestufe: 2  
Typ: MQCFIN

### **MsgId**

Beschreibung: Nachrichten-ID  
PCF-Parameter: MQBACF\_MSG\_ID  
Tracestufe: 2  
Typ: MQCFBS  
Länge: MQ\_MSG\_ID\_LENGTH

### **CorrelId**

PCF-Parameter: Korrelations-ID  
Beschreibung: MQBACF\_CORREL\_ID  
Tracestufe: 2  
Typ: MQCFBS  
Länge: MQ\_CORREL\_ID\_LENGTH

### **ReplyToQueue**

Beschreibung:

PCF-Parameter: MQCACF\_REPLY\_TO\_Q  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH

### ***ReplyToQMgr***

Beschreibung:  
PCF-Parameter: MQCACF\_REPLY\_TO\_Q\_MGR  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQCFST

### ***CodedCharSetId***

Beschreibung: Zeichensatzkennung von Nachrichtendaten  
PCF-Parameter: MQIA\_CODED\_CHAR\_SET\_ID  
Tracestufe: 2  
Typ: MQCFIN

### ***Encoding***

Beschreibung: Numerische Codierung von Nachrichtendaten.  
PCF-Parameter: MQIACF\_ENCODING  
Tracestufe: 2  
Typ: MQCFIN

### ***PutDate***

Beschreibung:  
PCF-Parameter: MQCACF\_PUT\_DATE  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_PUT\_DATE\_LENGTH

### ***PutTime***

Beschreibung:  
PCF-Parameter: MQCACF\_PUT\_TIME  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_PUT\_TIME\_LENGTH

### ***PolicyName***

Beschreibung: Der Richtlinienname, der auf diese Nachricht angewendet wurde.

**Anmerkung:** Nur AMS-geschützte Nachrichten

PCF-Parameter: MQCA\_POLICY\_NAME  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_OBJECT\_NAME\_LENGTH

### ***XmitqMsgId***

Beschreibung: Die Nachrichten-ID der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQBACF\_XQH\_MSG\_ID  
Tracestufe: 2  
Typ: MQCFBS  
Länge: MQ\_MSG\_ID\_LENGTH

### ***XmitqCorrelId***

Beschreibung: Die Korrelations-ID der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQBACF\_XQH\_CORREL\_ID  
Tracestufe: 2  
Typ: MQCFBS  
Länge: MQ\_CORREL\_ID\_LENGTH

### ***XmitqPutTime***

Beschreibung: Die Zeitzeit der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_PUT\_TIME  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_PUT\_TIME\_LENGTH

### ***XmitqPutDate***

Beschreibung: Das Datum, an dem die Nachricht in den Header der Übertragungswarteschlange gestellt wurde.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_PUT\_DATE  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_PUT\_DATE\_LENGTH

### ***XmitqRemoteQName***

Beschreibung: Das Ziel der fernen Warteschlange der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_REMOTE\_Q\_NAME

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_Q\_NAME\_LENGTH

### ***XmitqRemoteQMgr***

Beschreibung: Die Zieladresse des fernen Warteschlangenmanagers der Nachricht im Header der Übertragungswarteschlange.

**Anmerkung:** Nur bei Format: MQFMT\_XMIT\_Q\_HEADER

PCF-Parameter: MQCACF\_XQH\_REMOTE\_Q\_MGR

Tracestufe: 2

Typ: MQCFST

Länge: MQ\_Q\_NAME\_LENGTH

### ***PutMsgOptsStructure***

Beschreibung: Die MQPMO-Struktur.

PCF-Parameter: MQBACF\_MQPMO\_STRUCT

Tracestufe: 3

Typ: MQCFBS

Länge: Die Länge der MQPMO-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

### ***QMgrOpDuration***

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64\_QMGR\_OP\_DURATION

Tracestufe: 2

Typ: MQCFIN64

*MQPUT1 AppActivityDistList PCF-Header-Struktur*

---

<sup>4</sup> Der Parameter "ResolvedLocalQName" ist nur enthalten, wenn er sich vom Parameter "ResolvedQName" unterscheidet.

Wenn die Funktion MQPUT1 in eine Verteilerliste gestellt wird, enthalten die Variablenparameter eine AppActivityDistList-PCF-Gruppe. Informationen zu den einzelnen Warteschlangen in der Verteilerliste finden Sie in „PCF-Gruppenheader der Verteilerliste der Anwendungsaktivität“ auf Seite 261. Die PCF-Gruppe "AppActivityDistList" kombiniert Informationen aus den MQOR-, MQPMR- und MQRR-Strukturen, um die Objekte und die PUT-Parameter zu identifizieren, und geben das Ergebnis der PUT-Operation für jede Warteschlange an. Für MQPUT1-Operationen enthält die Gruppe AppActivityDistList einige oder alle der folgenden Parameter (der CompCode, Reason, ObjectName und ObjectQMgrName ist vorhanden, wenn der Ursachencode MQRC\_MULTIPLE\_REASONS ist und die anderen Parameter durch das Feld MQPMO.PutMsgRecFields bestimmt werden):

### **CompCode**

Beschreibung:	Der Beendigungscode, der das Ergebnis der Einlegung für dieses Objekt angibt. Wird nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für MQPUT1 MQRC_MULTIPLE_REASONS ist
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	2
Typ:	MQCFIN

### **Reason**

Beschreibung:	Der Ursachencode, der das Ergebnis der Einordnung für dieses Objekt angibt. Wird nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für MQPUT1 MQRC_MULTIPLE_REASONS ist
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	2
Typ:	MQCFIN

### **objektname**

Beschreibung:	Der Name einer Warteschlange in der Verteilerliste. Wird nur eingeschlossen, wenn MQOR-Strukturen bereitgestellt werden.
PCF-Parameter:	MQCACF_OBJECT_NAME
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_Q_NAME_LENGTH

### **MsgId**

Beschreibung:	Nachrichten-ID. Nur eingeschlossen, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF_MSG_ID
PCF-Parameter:	MQBACF_MSG_ID
Tracestufe:	2
Typ:	MQCFBS
Länge:	MQ_MSG_ID_LENGTH

### **CorrelId**

Beschreibung:	Korrelations-ID Wird nur eingefügt, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF_CORREL_ID
PCF-Parameter:	MQBACF_CORREL_ID

Tracestufe: 2  
Typ: MQCFBS  
Länge: MQ\_CORREL\_ID\_LENGTH

### **GroupId**

Beschreibung: Gruppen-ID. Nur eingeschlossen, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF\_GROUP\_ID  
PCF-Parameter: MQBACF\_GROUP\_ID  
Tracestufe: 2  
Typ: MQCFBS  
Länge: MQ\_GROUP\_ID\_LENGTH

### **Feedback**

Beschreibung: Rückkopplungscode. Nur eingeschlossen, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF\_FEEDBACK  
PCF-Parameter: MQIACF\_FEEDBACK  
Tracestufe: 2  
Typ: MQCFIN

### **AccountingToken**

Beschreibung: AccountingToken. Nur eingeschlossen, wenn MQPMR-Strukturen provided.und PutMsgRecFields enthält MQPMRF\_ACCOUNTING\_TOKEN  
PCF-Parameter: MQBACF\_ACCOUNTING\_TOKEN  
Tracestufe: 2  
Typ: MQCFBS  
Länge: MQ\_ACCOUNTING\_TOKEN\_LENGTH.

### **MQSET**

Die Anwendung hat die MQSET-MQI-Funktion gestartet.

### **ObjectHandle**

Beschreibung: Die Objektkennung  
PCF-Parameter: MQIACF\_HOBJ  
Tracestufe: 1  
Typ: MQCFIN

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.  
PCF-Parameter: MQIACF\_COMP\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.

PCF-Parameter: MQIACF\_REASON\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### ***SelectorCount***

Beschreibung: Die Anzahl der Selektoren, die in der Selektorenfeldgruppe bereitgestellt werden.  
PCF-Parameter: MQIACF\_SELECTOR\_COUNT  
Tracestufe: 2  
Typ: MQCFIN

### ***Selectors***

Beschreibung: Die Liste der Attribute (Ganzzahl oder Zeichen), deren Werte durch MQSET aktualisiert werden.  
PCF-Parameter: MQIACF\_SELECTORS  
Tracestufe: 2  
Typ: MQCFIL

### ***ResolvedQName***

Beschreibung: Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT\_Q ist.  
PCF-Parameter: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
Tracestufe: 2  
Typ: MQCFST  
Länge: MQ\_Q\_NAME\_LENGTH.

### ***ResObjectString***

Beschreibung: Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT\_TOPIC lautet.  
PCF-Parameter: MQCACF\_RESOLVED\_OBJECT\_STRING  
Tracestufe: 2  
Typ: MQCFST  
Länge: Länge variiert.

### ***ResolvedType***

Beschreibung: Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT\_Q', 'MQOT\_TOPIC' oder 'MQOT\_NONE'.  
PCF-Parameter: MQIACF\_RESOLVED\_TYPE  
Tracestufe: 2  
Typ: MQCFIN

### **IntAttrCount**

Beschreibung:	Die Anzahl der ganzzahligen Attribute, die durch die festgelegte Operation aktualisiert werden sollen.
PCF-Parameter:	MQIACF_INTATTR_COUNT
Tracestufe:	3
Typ:	MQCFIN

### **IntAttrrs**

Beschreibung:	Die ganzzahligen Attributwerte
PCF-Parameter:	MQIACF_INT_ATTRS
Tracestufe:	3
Typ:	MQCFIL
Bereich:	Dieser Parameter ist nur vorhanden, wenn 'IntAttrCount' größer als 0 ist.

### **CharAttrrs**

Beschreibung:	Die Zeichenattribute, die durch die festgelegte Operation aktualisiert werden sollen. Die Werte werden miteinander verkettet.
PCF-Parameter:	MQCACF_CHAR_ATTRS
Tracestufe:	3
Typ:	MQCFST
Bereich:	Dieser Parameter ist nur enthalten, wenn 'CharAttrLength' größer als 0 ist.

### **QMGrOpDuration**

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.  <b>Anmerkung:</b> Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ:	MQCFIN64

### **MQSUB**

Die Anwendung hat die MQSUB-MQI-Funktion gestartet.

### **CompCode**

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ:	MQCFIN

### **Reason**

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ:	MQCFIN

### **SubHandle**

Beschreibung:	Die Subskriptionskennung
PCF-Parameter:	MQIACF_HSUB
Tracestufe:	1
Typ:	MQCFIN

### **ObjectHandle**

Beschreibung:	Die Objektkennung
PCF-Parameter:	MQIACF_HOBJ
Tracestufe:	1
Typ:	MQCFIN

### **Options**

Beschreibung:	Subskriptionsoptionen
PCF-Parameter:	MQIACF_SUB_OPTIONS
Tracestufe:	1
Typ:	MQCFIN

### **objektname**

Beschreibung:	Der Name des Objekts.
PCF-Parameter:	MQCACF_OBJECT_NAME
Tracestufe:	1
Typ:	MQCFST
Länge:	MQ_Q_NAME_LENGTH

### **ObjectString**

Beschreibung:	Langer Objektname.
PCF-Parameter:	MQCACF_OBJECT_STRING
Tracestufe:	1
Typ:	MQCFST
Bereich:	Wird nur aufgenommen, wenn das Feld 'VSLength' von 'MQSD.ObjectString' größer als null oder 'MQVS_NULL_TERMINATED' ist.
Länge:	Länge variiert.

### **AlternateUserId**

Beschreibung:	
---------------	--

PCF-Parameter: MQCACF\_ALTERNATE\_USERID  
 Tracestufe: 2  
 Typ: MQCFST  
 Bereich: Wird nur eingeschlossen, wenn MQSO\_ALTERNATE\_USER\_AUTHORITY angegeben ist.  
 Länge: MQ\_USER\_ID\_LENGTH

### ***AlternateSecurityId***

Beschreibung: Alternative Sicherheits-ID.  
 PCF-Parameter: MQBACF\_ALTERNATE\_SECURITYID  
 Tracestufe: 2  
 Typ: MQCFBS  
 Bereich: Nur vorhanden, wenn MQSO\_ALTERNATE\_USER\_AUTHORITY angegeben ist und MQSD.AlternateSecurityId nicht mit MQSID\_NONE identisch ist.  
 Länge: MQ\_SECURITY\_ID\_LENGTH

### ***SubName***

Beschreibung: Subskriptionsname  
 PCF-Parameter: MQCACF\_SUB\_NAME  
 Tracestufe: 2  
 Typ: MQCFST  
 Bereich: Wird nur aufgenommen, wenn das Feld 'VSLength' von 'MQSD.SubName' größer als null oder 'MQVS\_NULL\_TERMINATED' ist.  
 Länge: Länge variiert.

### ***SubUserData***

Beschreibung: Subskriptions-Benutzerdaten  
 PCF-Parameter: MQCACF\_SUB\_USER\_DATA  
 Tracestufe: 2  
 Typ: MQCFST  
 Bereich: Wird nur aufgenommen, wenn das Feld 'VSLength' von 'MQSD.SubName' größer als null oder 'MQVS\_NULL\_TERMINATED' ist.  
 Länge: Länge variiert.

### ***SubCorrelId***

Beschreibung: ID der Subskriptionskorrelation  
 PCF-Parameter: MQBACF\_SUB\_CORREL\_ID  
 Tracestufe: 2  
 Typ: MQCFBS  
 Länge: MQ\_CORREL\_ID\_LENGTH

### ***SelectionString***

Beschreibung:	Auswahlzeichenfolge.
PCF-Parameter:	MQCACF_SELECTION_STRING
Tracestufe:	2
Typ:	MQCFST
Bereich:	Wird nur eingeschlossen, wenn das Feld für die VSLength-Funktion MQSD ist. SelectionString ist MQVS_NULL_TERMINATED oder größer als null.
Länge:	Länge variiert.

### ***ResolvedQName***

Beschreibung:	Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT_Q ist.
PCF-Parameter:	MQCACF_RESOLVED_LOCAL_Q_NAME
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_Q_NAME_LENGTH.

### ***ResObjectString***

Beschreibung:	Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT_TOPIC lautet.
PCF-Parameter:	MQCACF_RESOLVED_OBJECT_STRING
Tracestufe:	2
Typ:	MQCFST
Länge:	Länge variiert.

### ***ResolvedType***

Beschreibung:	Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT_Q', 'MQOT_TOPIC' oder 'MQOT_NONE'.
PCF-Parameter:	MQIACF_RESOLVED_TYPE
Tracestufe:	2
Typ:	MQCFIN

### ***SubDescriptorStructure***

Beschreibung:	Die MQSD-Struktur.
PCF-Parameter:	MQBACF_MQSD_STRUCT
Tracestufe:	3
Typ:	MQCFBS
Länge:	Die Länge (in Byte) der MQSD-Struktur.

### **QMGrOpDuration**

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.  
Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64\_QMGR\_OP\_DURATION  
Tracestufe: 2  
Typ: MQCFIN64

### **MQSUBRQ**

Die Anwendung hat die MQSUBRQ-MQI-Funktion gestartet.

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.  
PCF-Parameter: MQIACF\_COMP\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.  
PCF-Parameter: MQIACF\_REASON\_CODE  
Tracestufe: 1  
Typ: MQCFIN

### **SubHandle**

Beschreibung: Die Subskriptionskennung  
PCF-Parameter: MQIACF\_HSUB  
Tracestufe: 1  
Typ: MQCFIN

### **SubOptions**

Beschreibung: Die Unteroptionen von MQSB.Options  
PCF-Parameter: MQIACF\_SUBRQ\_OPTIONS  
Tracestufe: 2  
Typ: MQCFIN

### **Aktion**

Beschreibung: Die Aktion für die Subskriptionsanforderung (MQSR\_\*)  
PCF-Parameter: MQIACF\_SUBRQ\_ACTION  
Tracestufe: 2

Typ: MQCFIN

### **NumPubs**

Beschreibung: Die Anzahl der Veröffentlichungen, die als Ergebnis dieses Aufrufs gesendet wurden (aus MQSB.NumPubs)

PCF-Parameter: MQIACF\_NUM\_PUBS

Tracestufe: 2

Typ: MQCFIN

### **QMGrOpDuration**

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64\_QMGR\_OP\_DURATION

Tracestufe: 2

Typ: MQCFIN64

### **MQSTAT**

Die Anwendung hat die MQSTAT-MQI-Funktion gestartet.

### **CompCode**

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.

PCF-Parameter: MQIACF\_COMP\_CODE

Tracestufe: 1

Typ: MQCFIN

### **Reason**

Beschreibung: Das Ursachencode-Ergebnis der Operation.

PCF-Parameter: MQIACF\_REASON\_CODE

Tracestufe: 1

Typ: MQCFIN

### **Type**

Beschreibung: Typ der angeforderten Statusinformationen

PCF-Parameter: MQIACF\_STATUS\_TYPE

Tracestufe: 2

Typ: MQCFIN

### **StatusStructure**

Beschreibung: Die MQSTS-Struktur.

PCF-Parameter:	MQBACF_MQSTS_STRUCT
Tracestufe:	3
Typ:	MQCFBS
Länge:	Die Länge in Byte der MQSTS-Struktur (die tatsächliche Größe hängt von der Strukturversion ab)

### ***QMGrOpDuration***

**Beschreibung:** Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

**Anmerkung:** Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

### ***Variablenparameter für Anwendungsaktivitäten-XA-Operationen***

Bei XA-Operationen handelt es sich um API-Aufrufe, die Anwendungen ermöglichen, um MQ die Teilnahme an einer Transaktion zu ermöglichen. Die Parameter für jede Operation werden im folgenden Abschnitt definiert.

Die Tracestufe gibt die Stufe der Trace-Granularität an, die für die Parameter erforderlich ist, die in den Trace eingeschlossen werden sollen. Folgende Werte für die Tracestufe sind möglich:

#### 1. Niedrig

Der Parameter ist eingeschlossen, wenn die Traceverarbeitung für " low ", " medium " oder " high " für eine Anwendung konfiguriert ist. Diese Einstellung bedeutet, dass ein Parameter immer in der Gruppe `AppActivityData` für die Operation enthalten ist. Diese Parametergruppe reicht aus, um einen Trace für die MQI-Aufrufe, die eine Anwendung vornimmt, zu verfolgen und zu sehen, ob sie erfolgreich sind.

#### 2. Mittel

Der Parameter ist nur in der Gruppe `AppActivityData` für die Operation enthalten, wenn die Traceverarbeitung für " mittel " oder " high " für eine Anwendung konfiguriert ist. Diese Gruppe von Parametern fügt Informationen zu den Ressourcen hinzu, z. B. die von der Anwendung verwendeten Warteschlangen- und Topic-Namen.

#### 3. Hoch

Der Parameter ist nur in der Gruppe `AppActivityData` für die Operation enthalten, wenn die Traceverarbeitung für " hohe " Aktivitäten für eine Anwendung konfiguriert ist. Diese Gruppe von Parametern enthält Speicherauszüge der Strukturen, die an die MQI- und XA-Funktionen übergeben werden. Aus diesem Grund enthält er weitere Informationen zu den Parametern, die in MQI- und XA-Aufrufen verwendet werden. Die Speicherauszüge des Strukturspeichers sind flache Kopien der Strukturen. Um fehlerhafte Versuche zu vermeiden, Zeiger zu dereferenzieren, werden die Zeigerwerte in den Strukturen auf NULL gesetzt.

**Anmerkung:** Die Version der Struktur, für die ein Speicherauszug erstellt wird, ist nicht notwendigerweise identisch mit der Version, die von einer Anwendung verwendet wird. Die Struktur kann durch einen API-Überkreuzungsexit, durch den Aktivitäts-Trace-Code oder durch den Warteschlangenmanager geändert werden. Ein Warteschlangenmanager kann eine Struktur in einer späteren Version ändern,

aber der WS-Manager ändert ihn nie in eine frühere Version der Struktur. Dazu würde das Risiko eines Datenverlustes gefährdet.

#### *AXREG*

Anwendung hat die AXREG-AX-Funktion gestartet

##### **XID**

Beschreibung: Die XID-Struktur  
PCF-Parameter: MQBACF\_XA\_XID  
Tracestufe: 1  
Typ: MQCFBS  
Länge: Sizeof (XID)

##### **Rmid**

Beschreibung: Ressourcenmanager-ID  
PCF-Parameter: MQIACF\_XA\_RMID  
Tracestufe: 1  
Typ: MQCFIN

##### **Flags**

Beschreibung: Flags  
PCF-Parameter: MQIACF\_XA\_FLAGS  
Tracestufe: 1  
Typ: MQCFIN

##### **XARetCode**

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

#### *AXUNREG*

Anwendung hat die AXUNREG-AX-Funktion gestartet

##### **Rmid**

Beschreibung: Ressourcenmanager-ID  
PCF-Parameter: MQIACF\_XA\_RMID  
Tracestufe: 1  
Typ: MQCFIN

##### **Flags**

Beschreibung: Flags  
PCF-Parameter: MQIACF\_XA\_FLAGS  
Tracestufe: 1  
Typ: MQCFIN

### ***XARetCode***

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

### ***XACLOSE***

Die Anwendung hat die XACLOSE-AX-Funktion gestartet.

### ***Xa\_info***

Beschreibung: Informationen, die zum Initialisieren des Ressourcenmanagers verwendet werden.  
PCF-Parameter: MQCACF\_XA\_INFO  
Tracestufe: 1  
Typ: MQCFST

### ***Rmid***

Beschreibung: Ressourcenmanager-ID  
PCF-Parameter: MQIACF\_XA\_RMID  
Tracestufe: 1  
Typ: MQCFIN

### ***Flags***

Beschreibung: Flags  
PCF-Parameter: MQIACF\_XA\_FLAGS  
Tracestufe: 1  
Typ: MQCFIN

### ***XARetCode***

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

### ***XACOMMIT***

Die Anwendung hat die XACOMMIT-AX-Funktion gestartet.

### ***XID***

Beschreibung: Die XID-Struktur  
PCF-Parameter: MQBACF\_XA\_XID  
Tracestufe: 1  
Typ: MQCFBS  
Länge: Sizeof (XID)

**Rmid**

Beschreibung: Ressourcenmanager-ID  
PCF-Parameter: MQIACF\_XA\_RMID  
Tracestufe: 1  
Typ: MQCFIN

**Flags**

Beschreibung: Flags  
PCF-Parameter: MQIACF\_XA\_FLAGS  
Tracestufe: 1  
Typ: MQCFIN

**XARetCode**

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

**XACOMPLETE**

Die Anwendung hat die Funktion XACOMPLETE AX gestartet.

**Handle**

Beschreibung: Operation für asynchrone Verarbeitung handhaben  
PCF-Parameter: MQIACF\_XA\_HANDLE  
Tracestufe: 1  
Typ: MQCFIN

**Retval**

Beschreibung: Rückgabewert der asynchronen Funktion  
PCF-Parameter: MQIACF\_XA\_RETVAL  
Tracestufe: 1  
Typ: MQCFINMQCFBS

**Rmid**

Beschreibung: Ressourcenmanager-ID  
PCF-Parameter: MQIACF\_XA\_RMID  
Tracestufe: 1  
Typ: MQCFIN

**Flags**

Beschreibung: Flags  
PCF-Parameter: MQIACF\_XA\_FLAGS  
Tracestufe: 1

Typ: MQCFIN

### ***XARetCode***

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

### ***XAEND***

Die Anwendung hat die XAEND-AX-Funktion gestartet.

### ***XID***

Beschreibung: Die XID-Struktur  
PCF-Parameter: MQBACF\_XA\_XID  
Tracestufe: 1  
Typ: MQCFBS  
Länge: Sizeof (XID)

### ***Rmid***

Beschreibung: Ressourcenmanager-ID  
PCF-Parameter: MQIACF\_XA\_RMID  
Tracestufe: 1  
Typ: MQCFIN

### ***Flags***

Beschreibung: Flags  
PCF-Parameter: MQIACF\_XA\_FLAGS  
Tracestufe: 1  
Typ: MQCFIN

### ***XARetCode***

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

### ***XAFORGET***

Anwendung hat die AXREG-AX-Funktion gestartet

### ***XID***

Beschreibung: Die XID-Struktur  
PCF-Parameter: MQBACF\_XA\_XID  
Tracestufe: 1  
Typ: MQCFBS

Länge:                    Sizeof (XID)

### ***Rmid***

Beschreibung:        Ressourcenmanager-ID  
PCF-Parameter:       MQIACF\_XA\_RMID  
Tracestufe:           1  
Typ:                   MQCFIN

### ***Flags***

Beschreibung:        Flags  
PCF-Parameter:       MQIACF\_XA\_FLAGS  
Tracestufe:           1  
Typ:                   MQCFIN

### ***XARetCode***

Beschreibung:        Rückkehrcode  
PCF-Parameter:       MQIACF\_XA\_RETCODE  
Tracestufe:           1  
Typ:                   MQCFIN

### ***XAOPEN***

Die Anwendung hat die Funktion XAOPEN AX gestartet.

### ***Xa\_info***

Beschreibung:        Informationen, die zum Initialisieren des Ressourcenmanagers verwendet werden.  
PCF-Parameter:       MQCACF\_XA\_INFO  
Tracestufe:           1  
Typ:                   MQCFST

### ***Rmid***

Beschreibung:        Ressourcenmanager-ID  
PCF-Parameter:       MQIACF\_XA\_RMID  
Tracestufe:           1  
Typ:                   MQCFIN

### ***Flags***

Beschreibung:        Flags  
PCF-Parameter:       MQIACF\_XA\_FLAGS  
Tracestufe:           1  
Typ:                   MQCFIN

### ***XARetCode***

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

### ***XAPREPARE***

Die Anwendung hat die Funktion XAPREPARE AX gestartet.

### ***XID***

Beschreibung: Die XID-Struktur  
PCF-Parameter: MQBACF\_XA\_XID  
Tracestufe: 1  
Typ: MQCFBS  
Länge: Sizeof (XID)

### ***Rmid***

Beschreibung: Ressourcenmanager-ID  
PCF-Parameter: MQIACF\_XA\_RMID  
Tracestufe: 1  
Typ: MQCFIN

### ***Flags***

Beschreibung: Flags  
PCF-Parameter: MQIACF\_XA\_FLAGS  
Tracestufe: 1  
Typ: MQCFIN

### ***XARetCode***

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

### ***XARECOVER***

Anwendung hat die XARECOVER AX-Funktion gestartet

### ***Count***

Beschreibung: Anzahl der XIDs  
PCF-Parameter: MQIACF\_XA\_COUNT  
Tracestufe: 1  
Typ: MQCFIN

### ***XIDs***

Beschreibung: Die XID-Strukturen

**Anmerkung:** Es gibt mehrere Instanzen dieses PCF-Parameters-eine für jede XID-Struktur bis zu Count XIDs

PCF-Parameter: MQBACF\_XA\_XID

Tracestufe: 1

Typ: MQCFBS

Länge: Sizeof (XID)

### ***Rmid***

Beschreibung: Ressourcenmanager-ID

PCF-Parameter: MQIACF\_XA\_RMID

Tracestufe: 1

Typ: MQCFIN

### ***Flags***

Beschreibung: Flags

PCF-Parameter: MQIACF\_XA\_FLAGS

Tracestufe: 1

Typ: MQCFIN

### ***XARetCode***

Beschreibung: Rückkehrcode

PCF-Parameter: MQIACF\_XA\_RETCODE

Tracestufe: 1

Typ: MQCFIN

### ***XAROLLBACK***

Die Anwendung hat die XAROLLBACK-AX-Funktion gestartet.

### ***XID***

Beschreibung: Die XID-Struktur

PCF-Parameter: MQBACF\_XA\_XID

Tracestufe: 1

Typ: MQCFBS

Länge: Sizeof (XID)

### ***Rmid***

Beschreibung: Ressourcenmanager-ID

PCF-Parameter: MQIACF\_XA\_RMID

Tracestufe: 1

Typ: MQCFIN

### **Flags**

Beschreibung: Flags  
PCF-Parameter: MQIACF\_XA\_FLAGS  
Tracestufe: 1  
Typ: MQCFIN

### **XARetCode**

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

### **XASTART**

Anwendung hat die XASTART AX-Funktion gestartet

### **XID**

Beschreibung: Die XID-Struktur  
PCF-Parameter: MQBACF\_XA\_XID  
Tracestufe: 1  
Typ: MQCFBS  
Länge: Sizeof (XID)

### **Rmid**

Beschreibung: Ressourcenmanager-ID  
PCF-Parameter: MQIACF\_XA\_RMID  
Tracestufe: 1  
Typ: MQCFIN

### **Flags**

Beschreibung: Flags  
PCF-Parameter: MQIACF\_XA\_FLAGS  
Tracestufe: 1  
Typ: MQCFIN

### **XARetCode**

Beschreibung: Rückkehrcode  
PCF-Parameter: MQIACF\_XA\_RETCODE  
Tracestufe: 1  
Typ: MQCFIN

Systemthemen in Themenstrukturen des Warteschlangenmanagers werden für die Ressourcenüberwachung (einige davon ähneln dem Inhalt von Statistiknachrichten) und als Methode zur Verarbeitung von Anwendungsaktivitätstraces verwendet.

### \$SYS/MQ-Zweig der Themenstruktur des Warteschlangenmanagers

Die Themenstruktur jedes Warteschlangenmanagers enthält die Verzweigung \$SYS/MQ . Der Warteschlangenmanager veröffentlicht die Themenzeichenfolgen in dieser Verzweigung. Ein berechtigter Benutzer kann diese Themenzeichenfolgen subscribieren, um Informationen über den Warteschlangenmanager und die Aktivität auf ihm zu erhalten. Diese Systemthemen werden für den Trace und die Überwachung von Anwendungsaktivitäten verwendet. Weitere Informationen finden Sie unter [Themenbaumstrukturen](#).

Das Stammverzeichnis der \$SYS/MQ-Verzweigung wird durch das Themenobjekt SYSTEM.ADMIN.TOPIC dargestellt. Der Zweig '\$SYS/MQ' der Themenstruktur wird auf die folgenden Arten vom Rest der Themenstruktur isoliert:

- Eine Subskription, die mit Platzhalterzeichen an einem höheren Punkt in der Baumstruktur als \$SYS/MQ erstellt wird, stimmt nicht mit einer Themenzeichenfolge innerhalb der \$SYS/MQ-Verzweigung überein. Die Platzhalteroperation für SYSTEM.ADMIN.TOPIC ist auf " Block " gesetzt und kann nicht geändert werden. Diese Einschränkung gilt auch, wenn Sie Platzhalterzeichen mit dem **runmqsc** -Befehl DISPLAY TPSTATUS verwenden, um Knoten in der Themenstruktur anzuzeigen. Um Themenknoten in der \$SYS/MQ-Verzweigung anzuzeigen, starten Sie die Themenzeichenfolge mit \$SYS/MQ. Verwenden Sie beispielsweise \$SYS/MQ/# , um alle Knoten anzuzeigen.
- Sie müssen berechtigt sein, die Berechtigung zum Verwenden der \$SYS/MQ-Themenstruktur zu erteilen oder tiefer als \$SYS/MQ zu erteilen. Die Berechtigung zum Subscribieren einer Themenzeichenfolge basiert auf der Berechtigung, die für ein verwaltetes Themenobjekt an oder höher als die Themenzeichenfolge in der Themenstruktur erteilt wird. Berechtigungen, die im Stammverzeichnis (SYSTEM.BASE.TOPIC) erteilt werden, würden eine Benutzerberechtigung für alle Themenzeichenfolgen erteilen. Im Fall der \$SYS/MQ-Verzweigung gilt jedoch der Zugriff, der höher als \$SYS/MQ erteilt wurde, nicht auf die \$SYS/MQ-Themenzeichenfolgen.
- Die Verzweigung '\$SYS/MQ' der Themenstruktur wird von den Themenattributen isoliert, die in der Baumstruktur höher gesetzt sind. Die Warteschlange SYSTEM.ADMIN.TOPIC übernimmt keine Attribute aus einem Themenobjekt, das in der Themenstruktur höher definiert ist. Die Änderung der Attribute von SYSTEM.BASE.TOPIC hat zum Beispiel keine Auswirkungen auf das Verhalten der \$SYS/MQ-Verzweigung.

Alle Themenzeichenfolgen, die mit \$SYS/MQ beginnen, sind für die Verwendung durch IBM MQ reserviert. Für diese Themenzeichenfolgen gelten die folgenden Einschränkungen:

- Sie können Multicasting nicht aus der \$SYS/MQ-Verzweigung der Themenstruktur aktivieren.
- Clustering wird für die \$SYS/MQ-Verzweigung nicht unterstützt.
- Der Proxy-Subskriptionsmechanismus kann nicht auf " force " gesetzt werden.
- Anwendungen können nicht in einer \$SYS/MQ-Themenzeichenfolge veröffentlichen.
- Der Standardwert für Veröffentlichungs- und Subskriptionsbereich ist nur der lokale WS-Manager.
- Es können keine Platzhalterzeichen an den folgenden Punkten verwendet werden:
  - \$SYS/MQ/
  - \$SYS/MQ/INFO
  - \$SYS/MQ/INFO/QMGR
  - \$SYS/MQ/INFO/QMGR/ *queue\_manager\_name*
  - \$SYS/MQ/INFO/QMGR/ *queue\_manager\_name* /ActivityTrace

Versuche, an diesen Punkten Platzhalterzeichen zu verwenden, führen zu einem Subskriptionsfehler mit der Ursache MQRC\_ADMIN\_TOPIC\_STRING\_ERROR.

## Leistungsüberwachungs-API, die über Publish/Subscribe verfügbar ist

Ab IBM MQ 9.0 können Sie die Statistikdaten des Warteschlangenmanagers mithilfe des einfachen Publish/Subscribe-Mechanismus und entweder der amqsrua-Beispielanwendung oder Ihrer eigenen Anwendung überwachen. Statistikdaten werden in einem Systemthema unter \$SYS/MQ/INFO/QMGR veröffentlicht, das Benutzer bei der Überwachung von Ressourcen unterstützt. Diese Statistikdaten können entweder durch Ausführen der Beispielanwendung 'amqsrua' oder durch Schreiben einer Anwendung angezeigt werden, die das Thema des Ressourcenüberwachungssystems auf ähnliche Weise subskribiert wie **amqsrua**.

### Zugehörige Konzepte

„Anwendungsaktivitätstrace“ auf Seite 208

Der Aktivitätstrace der Anwendung erzeugt detaillierte Informationen zum Verhalten von Anwendungen, die mit einem Warteschlangenmanager verbunden sind. Dabei wird das Verhalten einer Anwendung verfolgt und es wird eine detaillierte Ansicht der Parameter bereitgestellt, die von einer Anwendung während der Interaktion mit IBM MQ verwendet werden. Es zeigt auch die Reihenfolge der MQI-Aufrufe, die von einer Anwendung ausgegeben werden.

## Systemressourcennutzung mit dem Befehl amqsrua überwachen

Sie können den Befehl **amqsrua** verwenden, um Leistungsdaten abzufragen, die sich auf die Systemressourcennutzung eines Warteschlangenmanagers beziehen.

### Informationen zu diesem Vorgang

Die Beispielanwendung 'amqsrua' zeigt eine Möglichkeit, wie Veröffentlichungen zur IBM MQ-Überwachung gelesen und die von Warteschlangenmanagern veröffentlichten Leistungsdaten angezeigt werden können. Diese Daten können Informationen über die CPU-, Speicher- und Plattenbelegung enthalten. Sie können auch Daten anzeigen, die äquivalent zu den STATMQI-PCF-Statistikdaten sind. Die Daten werden alle 10 Sekunden veröffentlicht und werden während der Ausführung des Befehls gemeldet.

Sie können den Befehl nur mit dem Namen des Warteschlangenmanagers ausführen und interaktiv durchgehen, um die Parameter **CLASS**, dann **TYPE** und dann **object** auszuwählen, die für den Warteschlangenmanager bei jedem Schritt verfügbar sind. Wenn Sie die **CLASS**-, **TYPE**- und **object**-Namen kennen, für die Sie Informationen anzeigen wollen, können Sie sie angeben, wenn Sie den Befehl **amqsrua** ausführen.

 Standardmäßig sucht die Anwendung 'amqsrua' nach Statistikdaten, die vom WS-Manager unter der Themenstruktur \$SYS/MQ/INFO/QMGR veröffentlicht werden. Andere Komponenten oder Anwendungen können einen ähnlichen Mechanismus verwenden, um unter einem anderen Themenstartpunkt zu veröffentlichen. Der IBM MQ Bridge to Salesforce, der auf x86-64-Linux-Plattformen verfügbar ist, veröffentlicht beispielsweise die Statistik unter \$SYS/Application/runmqsfb. Ab IBM MQ 9.1.0 können Sie mit dem Parameter **-p** angeben, wo amqsrua nach den Statistikdaten für diese anderen Komponenten in Linux und Windows sucht.

#### **-m**

Warteschlangenmanagername. Der WS-Manager muss aktiv sein. Wenn Sie keinen Warteschlangenmanagernamen angeben, werden die Statistiken für den Standardwarteschlangenmanager angezeigt.

#### **-c**

CLASS-Name. Veröffentlichungen zur IBM MQ-Ressourcennutzung sind einer Klasse zugeordnet. Die Klassen stellen die oberste Ebene in der Baumstruktur der Metadaten dar, die die verfügbaren Informationen zur Ressourcennutzung beschreibt.

#### **CPU**

Gibt Informationen zur CPU-Belastung zurück.

#### **PLATTE**

Gibt Informationen zur Plattenbelegung zurück.

## STATMQI

Gibt Informationen zur MQI-Verwendung zurück.

## STATQ

Gibt Informationen zur MQI-Verwendung pro Warteschlange zurück.

### V 9.1.5 STATAPP

Gibt Informationen zu Nutzungsstatistiken für die angegebene Anwendung zurück.

Im Abschnitt „[Eigenes Ressourcenüberwachungsanwendung entwickeln](#)“ auf Seite 302 finden Sie Informationen zur Angabe der Themenstruktur für Metadaten und unter [Anwendungsnamen in unterstützten Programmiersprachen verwenden](#) finden Sie Informationen zu den Zeichen, die Sie in den Namen von Anwendungen verwenden können.

**Anmerkung:** Mit der Option **-o** geben Sie den Anwendungsnamen an, der für STATAPP überwacht wird, und den Warteschlangennamen für STATQ.

### V 9.1.0

Informationen zu Klassen, die nur mit dem IBM MQ Bridge to Salesforce verfügbar sind, finden Sie im Abschnitt „[IBM MQ Bridge to Salesforce überwachen](#)“ auf Seite 306.

**-t**

TYPE name. Veröffentlichungen zur IBM MQ-Ressourcennutzung sind einem Typ innerhalb einer Klasse zugeordnet. Jede Veröffentlichung enthält die Klasse und den Typ, die es ermöglichen, die class/type/element-Definitionen zu finden und die resultierenden Veröffentlichungen zu bearbeiten. Die class/type/element-Beschreibungen werden als Metadaten beim Start des Warteschlangenmanagers veröffentlicht.

**-o**

Objektname Bei den Veröffentlichungen zur Ressourcennutzung handelt es sich um PCF-Nachrichten, die aus einer Folge von PCF-Elementen bestehen. Die PCF-Elemente, die für jedes Paar vom Typ/Typ veröffentlicht werden, werden in den Metadaten angezeigt. Ein Blatt wird in der Baumstruktur gespeichert, in der jedes Element beschrieben wird, wodurch die Elemente verarbeitet werden können.

### V 9.1.0

**-p**

Metadatenpräfix. Geben Sie einen Ausgangspunkt für die Themenstruktur an, an dem **amqsrua** nach Statistikdaten suchen kann, die von Warteschlangenmanagern veröffentlicht werden. Die Standardthemenbaumstruktur ist \$SYS/MQ/INFO/QMGR, aber andere Komponenten oder Anwendungen können Statistikdaten unter einem anderen Ausgangspunkt für eine Themenbaumstruktur veröffentlichen. IBM MQ Bridge to Salesforce veröffentlicht die Statistik beispielsweise unter \$SYS/Application/runmqsfb.

**-n**

Veröffentlichungszähler. Sie können angeben, wie viele Berichte zurückgegeben werden sollen, bevor der Befehl beendet wird. Die Daten werden ungefähr alle zehn Sekunden veröffentlicht. Wenn Sie also den Wert 50 eingeben, gibt der Befehl 50 Berichte über 500 Sekunden zurück. Wenn Sie diesen Parameter nicht angeben, wird der Befehl ausgeführt, bis entweder ein Fehler auftritt oder der WS-Manager heruntergefahren wird.

### V 9.1.0.10

**-s**

Modellwarteschlange. Geben Sie optional eine zu verwendende Modellwarteschlange an. (Standardmäßig verwendet **amqsrua** das SYSTEM.DEFAULT.MODEL.QUEUE Warteschlange.)

**-h**

Verwendung

## Vorgehensweise

1. Geben Sie im Verzeichnis "samples" den folgenden Befehl aus, um die verfügbaren Daten für den Warteschlangenmanager anzuzeigen:

- **Linux** Unter Linux, `MQ_INSTALLATION_PATH/samp/bin`:

```
./amqsrua -m QMgrName
```

- **Windows** Unter Windows, `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64:`

```
amqsrua -m QMgrName
```

Dabei gibt `QMgrName` den Namen des Warteschlangenmanagers an, der abgefragt werden soll. Der WS-Manager muss aktiv sein. Wenn Sie keinen Warteschlangenmanagernamen angeben, wird der Standardwarteschlangenmanager verwendet.

Folgende Optionen sind verfügbar:

```
CPU : Platform central processing units
DISK : Platform persistent data stores
STATMQI : API usage statistics
STATQ : API per-queue usage statistics
Enter Class selection
==>
```

2. Geben Sie in der Liste der CLASS-Optionen `STATMQI` ein.

```
==> STATMQI
CONNDISC : MQCONN and MQDISC
OPENCLOSE : MQOPEN and MQCLOSE
INQSET : MQINQ and MQSET
PUT : MQPUT
GET : MQGET
SYNCPOINT : Commit and rollback
SUBSCRIBE : Subscribe
PUBLISH : Publish
Enter Type selection
==>
```

3. Geben Sie in der Liste der TYPE-Optionen `PUT` ein.

```
==>PUT
Publication received PutDate:20170329 PutTime:17045485 Interval:4 minutes,13.978 seconds
Interval total MQPUT/MQPUT1 count 22
Interval total MQPUT/MQPUT1 byte count 25284 100/sec
Non-persistent message MQPUT count 22
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 25284 100/sec
Put persistent messages - byte count 0
MQSTAT count 0

Publication received PutDate:20170329 PutTime:17050485 Interval:10.001 seconds
Interval total MQPUT/MQPUT1 count 1
Interval total MQPUT/MQPUT1 byte count 524 52/sec
Non-persistent message MQPUT count 1
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 524 52/sec
Put persistent messages - byte count 0
MQSTAT count 0
```

## Ergebnisse

Sie haben die `amqsrua`-Beispielanwendung interaktiv verwendet, um Statistikdaten anzuzeigen, die Warteschlangenmanager in den Systemthemen unter dem Metadatenpräfix `$SYS/MQ/INFO/QMGR` veröffentlichten.

**Anmerkung:** Klassen und Typen von Ressourcenveröffentlichungen, die für Warteschlangenmanager verfügbar sind, können abhängig von ihrer Konfiguration, Version und Plattform unterschiedlich sein. Verwenden Sie `amqsrua` interaktiv, um die Klassen, Typen und Elemente zu suchen, die für Ihren speziellen Warteschlangenmanager verfügbar sind.

## Nächste Schritte

Informationen zum Entwickeln Ihrer eigenen Überwachungsanwendung finden Sie unter [„Eigenes Ressourcenüberwachungsanwendung entwickeln“](#) auf Seite 302.

### Zugehörige Tasks

[„IBM MQ Bridge to Salesforce überwachen“](#) auf Seite 306

Mit dem C-Beispielprogramm **amqsrua** können Sie Themen subscribieren und Veröffentlichungen für Warteschlangenmanager anzeigen, die von IBM MQ Bridge to Salesforce verwendet werden.

[Vorbereiten und Ausführen von Beispielprogrammen unter Windows](#)

[Beispielprogramme unter UNIX und Linux vorbereiten und ausführen](#)

## **Eigenes Ressourcenüberwachungsanwendung entwickeln**

Sie können eine eigene Anwendung entwickeln, um Systemressourcen zu überwachen.

Jeder WS-Manager veröffentlicht Ressourcennutzungsdaten in Themen. Diese Daten werden von Subskribenten zu diesen Themen verarbeitet. Wenn ein WS-Manager gestartet wird, veröffentlicht der Warteschlangenmanager eine Gruppe von Nachrichten zu Metathemen. In diesen Nachrichten wird beschrieben, welche Themen zur Ressourcennutzung vom Warteschlangenmanager unterstützt werden, und der Inhalt der Nachrichten, die zu diesen Themen veröffentlicht werden. Verwaltungstools können die Metadaten subscribieren, um zu ermitteln, welche Informationen zur Ressourcennutzung verfügbar sind, und zu welchen Themen und subscribieren Sie die beworbenen Themen.

Die Themenstruktur für die Metadaten hat die folgende Struktur:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/instance]/type]
```

 Verwenden Sie ab IBM MQ 9.1.5 die folgende Struktur für die Metadaten der Themenstruktur:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/resourceid]/type]
```

Damit verdeutlichen Sie die Angabe der Ressource, die über einen Anwendungsnamen mit einer Länge von maximal 28 Zeichen verfügt, um eine Verwechslung mit den Attributen des internen Metadatentyps zu vermeiden.

Eine Liste der möglichen Klassen finden Sie im Abschnitt [„Systemressourcennutzung mit dem Befehl amqsrua überwachen“](#) auf Seite 299.

Der Quelltext für die Anwendung 'amqsrua' wird als IBM MQ-Beispiel bereitgestellt. Sie können diese Musteranwendung als Leitfaden für die Erstellung einer eigenen Überwachungsanwendung verwenden. Sie können die Quelle für das Beispiel aus einer IBM MQ-Clientinstallation abrufen. Die Quellendatei heißt `amqsruaa.c` und befindet sich im Verzeichnis mit den Beispielen:

-   Auf UNIX -und Linux -Plattformen: `MQ_INSTALLATION_PATH/samp/`
-  Windows-Plattformen: `MQ_INSTALLATION_PATH\tools\c\Samples\`

Das Programm 'amqsrua' subscribiert die Themen zur IBM MQ-Ressourcennutzung und formatiert die resultierenden veröffentlichten PCF-Daten. Die Anwendungsquelle stellt ein grundlegendes Beispiel dafür dar, wie diese Art von Verwaltungsdaten subscribieren und verarbeitet werden kann. Die Anwendung 'amqsrua' führt die folgenden Tasks aus:

- Erstellt eine nicht permanente Subskription für die Themen, die durch die Eingabeparameter angegeben werden.
- Ruft MQGET wiederholt auf, um Nachrichten aus den Themen abzurufen, und schreibt in stdout.
- Schreibt eine Nachricht für jeden MQI-Ursachencode (außer MQRC\_NONE).
- Stoppt, wenn ein MQI-Beendigungscode von MQCC\_FAILED vorhanden ist, oder wenn die angeforderte Anzahl von Veröffentlichungen zur Ressourcennutzung verbraucht wurde.

Metriken werden in Klassen und in Unterkategorien in Typen eingeteilt. Es gibt verschiedene Metriken, die unter jeder Metrikkategorie und jedem Metriktyp veröffentlicht werden.

## Index

- „[CPU \(Zentrale Plattform-Verarbeitungseinheiten\)](#)“ auf Seite 303
- „[DISK \(persistente Datenspeicher der Plattform\)](#)“ auf Seite 303
- „[STATMQI \(API-Nutzungsstatistiken\)](#)“ auf Seite 304
- „[STATQ \(API-Nutzungsstatistik pro Warteschlange\)](#)“ auf Seite 305

## CPU (Zentrale Plattform-Verarbeitungseinheiten)

### SystemSummary (CPU-Leistung - plattformübergreifend)

Benutzer-CPU-Zeit in Prozent  $X$  %  
 System-CPU-Zeit in Prozent  $X$  %  
 CPU-Belastung - einminütiger Durchschnitt  $X$   
 CPU-Belastung - Fünf-Minuten-Durchschnitt  $X$   
 CPU-Belastung - Durchschnitt über fünfzehn Minuten  $X$   
 RAM freier Prozentsatz  $X$  %  
 RAM-Gesamtbytes  $X$  MB

### QMGrSummary (CPU-Leistung - laufender Warteschlangenmanager)

Benutzer-CPU-Zeit - prozentuale Schätzung für Warteschlangenmanager  $X$  %  
 System-CPU-Zeit - prozentuale Schätzung für Warteschlangenmanager  $X$  %  
 RAM-Gesamtbytes - Schätzung für Warteschlangenmanager  $X$  MB

## DISK (persistente Datenspeicher der Plattform)

### SystemSummary (Plattenbelegung - plattformübergreifend)

MQ-Fehler Dateisystem - verwendete Bytes  $X$  MB  
 MQ-Fehler Dateisystem - freier Speicherbereich  $X$  %  
 Anzahl der MQ-FDC-Dateien  $X$   
 MQ-Trace-Dateisystem - verwendete Bytes  $X$  MB  
 MQ-Trace-Dateisystem - freier Speicherbereich  $X$  %

### QMGrSummary (Plattenbelegung - laufende Warteschlangenmanager)

Warteschlangenmanager-Dateisystem - verwendete Bytes  $X$  MB  
 Warteschlangenmanager-Dateisystem - freier Speicherbereich  $X$  %

### Protokoll (Plattenbelegung - Warteschlangenmanager-Wiederherstellungsprotokoll)

Log - verwendete Bytes  $X$   
 Protokoll - Bytes max.  $X$   
 Protokolldateisystem - verwendete Bytes  $X$   
 Protokolldateisystem - Bytes max.  $X$   
 Protokoll - geschriebene physische Bytes  $X$   
 Protokoll - geschriebene logische Bytes  $X$   
 Log - Latenzzeit für Schreiboperationen  $X$  uSek

**V 9.1.0** Protokoll - Schreibgröße  $X$

**V 9.1.0** Protokoll - Bytes, die von zu archivierenden Speicherbereichen belegt werden  $X$

▶ V 9.1.0 Protokoll - für die Datenträgerwiederherstellung erforderliche Bytes X

▶ V 9.1.0 Protokoll - von wiederverwendbaren Speicherbereichen belegte Bytes X

▶ V 9.1.0 Log - aktuell verwendeter Primärspeicherbereich X %

▶ V 9.1.0 Protokoll - Workload primäre Speicherbelegung X %

## **STATMQI (API-Nutzungsstatistiken)**

### **CONNDISC (MQCONN und MQDISC)**

MQCONN/MQCONNX-Zähler X

Failed MQCONN/MQCONNX-Zähler X

Gleichzeitig bestehende Verbindungen - obere Grenze X

MQDISC-Zähler X

### **OPENCLOSE (MQOPEN und MQCLOSE)**

MQOPEN-Zähler X Y/sek

Zähler fehlgeschlagene MQOPEN X

MQCLOSE-Zähler X Y/sek

Zähler fehlgeschlagene MQCLOSE X

### **INQSET (MQINQ und MQSET)**

Zähler MQINQ X

Zähler fehlgeschlagene MQINQ X

MQSET-Zähler X

Zähler fehlgeschlagene MQSET X

### **PUT (MQPUT)**

Intervall Gesamtanzahl MQPUT/MQPUT1 X

Intervall Gesamtanzahl MQPUT/MQPUT1-Bytes X Y/sek

Nicht persistente Nachrichten MQPUT-Zähler X

Persistente Nachrichten MQPUT-Zähler X

Zähler fehlgeschlagene MQPUT X

Nicht persistente Nachrichten MQPUT1-Zähler X

Persistente Nachrichten MQPUT1-Zähler X

Zähler fehlgeschlagene MQPUT1 X

Eingereichte nicht persistente Nachrichten - Byte-Zähler X Y/sek

Eingereichte persistente Nachrichten - Byte-Zähler X

MQSTAT-Zähler X

### **GET (MQGET)**

Intervall Gesamtabrufe mit Löschen - Zähler X

Intervall Gesamtabrufe mit Löschen - Byte-Zähler X Y/sek

Nicht persistente Nachrichten Abruf mit Löschen - Zähler X

Persistente Nachrichten Abruf mit Löschen - Zähler X

Zähler fehlgeschlagene MQGET X

Abgerufene nicht persistente Nachrichten - Byte-Zähler X Y/sek

Abgerufene persistente Nachrichten - Byte-Zähler X

Nicht persistente Nachrichten anzeigen - Zähler X

Anzeige persistenter Nachrichten - Zähler X

Zähler fehlgeschlagene Anzeigen X  
Anzeige nicht persistenter Nachrichten - Zähler X Y/sek  
Anzeige persistenter Nachrichten - Byte-Zähler X  
Zähler abgelaufene Nachrichten X  
Zähler gelöschte Warteschlange X  
MQCB-Zähler X  
Zähler fehlgeschlagene MQCB X  
MQCTL-Zähler X

#### **SYNCPOINT (Festschreiben und Rollback)**

Commitzählung X  
Rollbackzählung X

#### **SUBSCRIBE (subskribieren)**

Permanenten Subskriptionszähler erstellen X  
Permanenten Subskriptionszähler ändern X  
Permanenten Subskriptionszähler fortsetzen X  
Nicht permanenten Subskriptionszähler erstellen X  
Fehlgeschlagene Erstellung/Änderung/Fortsetzung des Subskriptionszählers X  
Permanenten Subskriptionszähler löschen X  
Nicht permanenten Subskriptionszähler löschen X  
Zähler fehlgeschlagene Subskriptionslöschung X  
MQSUBRQ-Zähler X  
Zähler fehlgeschlagene MQSUBRQ X  
Permanenter Subskribent - obere Grenze X  
Permanenter Subskribent - unterer Grenzwert X  
Nicht permanenter Subskribent - obere Grenze X  
Nicht permanenter Subskribent - unterer Grenzwert X

#### **PUBLISH (veröffentlichen)**

Topic MQPUT/MQPUT1 Intervall gesamt X  
Intervall Gesamtzahl der eingereichten Topic-Bytes X Y/sek.  
Veröffentlicht an Subskribenten - Nachrichtenanzahl X  
Veröffentlicht an Subskribenten - Byte-Zähler X  
Nicht permanent - Topic MQPUT/MQPUT1-Zähler X  
Permanent - Topic MQPUT/MQPUT1-Zähler X  
Zähler fehlgeschlagene Topic MQPUT/MQPUT1 X

#### **STATQ (API-Nutzungsstatistik pro Warteschlange)**

##### **OPENCLOSE (MQOPEN und MQCLOSE)**

MQOPEN-Zähler X  
MQCLOSE-Zähler X

##### **INQSET (MQINQ und MQSET)**

Zähler MQINQ X  
MQSET-Zähler X

##### **PUT (MQPUT und MQPUT1)**

MQPUT/MQPUT1-Zähler X  
MQPUT-Byte-Zähler X

MQPUT nicht persistente Nachrichten Zähler X  
MQPUT persistente Nachrichten Zähler X  
MQPUT1 nicht persistente Nachrichten Zähler X  
MQPUT persistente Nachrichten Zähler X  
Zähler nicht persistente Nachrichten X  
Zähler permanente Byte X  
Sperrkonflikt X %  
von der Warteschlange vermiedene Einreihungen X %  
von der Warteschlange vermiedene Bytes X %

#### **GET (MQGET)**

MQGET-Zähler X  
MQGET-Bytezähler X  
Zähler gelöschte MQGET nicht persistente Nachricht X  
Zähler gelöschte MQGET persistente Nachricht X  
Zähler gelöschte MQGET nicht permanente Byte X  
Zähler gelöschte MQGET permanente Byte X  
Zähler MQGET-Anzeigen nicht persistente Nachricht X  
Zähler MQGET-Anzeigen persistente Nachricht X  
Zähler MQGET-Anzeigen nicht permanente Byte X  
Zähler MQGET-Anzeigen permanente Byte X  
abgelaufene Nachrichten X  
Anzahl bereinigter Warteschlangen X  
Durchschnittliche Warteschlangenzeit X uSec  
Warteschlangenlänge X

V 9.1.0

Linux

## **IBM MQ Bridge to Salesforce überwachen**

Mit dem C-Beispielprogramm **amqsrua** können Sie Themen subscribieren und Veröffentlichungen für Warteschlangenmanager anzeigen, die von IBM MQ Bridge to Salesforce verwendet werden.

### **Informationen zu diesem Vorgang**

Beim Start des Warteschlangenmanagers veröffentlicht der Warteschlangenmanager eine Gruppe von Nachrichten über sich selbst für Systemthemen unter dem Metadatenpräfix \$SYS/MQ/INFO/QMGR. Die Nachrichten geben an, welche Leistungsdaten für Warteschlangenmanager verfügbar sind. Verwaltungstools können Themen subscribieren, um zu ermitteln, welche Informationen zur Ressourcennutzung verfügbar sind, damit sie die Ressourcennutzungsdaten verarbeiten können. Weitere Informationen finden Sie unter „[Systemressourcennutzung mit dem Befehl amqsrua überwachen](#)“ auf Seite 299.

Standardmäßig sucht das Programm **amqsrua** nach Statistikdaten, die vom Warteschlangenmanager unter der Themenstruktur \$SYS/MQ/INFO/QMGR veröffentlicht wurden. Andere Komponenten oder Anwendungen können einen ähnlichen Mechanismus verwenden, um unter einem anderen Themenstartpunkt zu veröffentlichen. Der IBM MQ Bridge to Salesforce, der auf x86-64-Linux-Plattformen verfügbar ist, veröffentlicht beispielsweise die Statistik unter \$SYS/Application/runmqsfb.

Sie können mit dem Parameter **-p** angeben, wo **amqsrua** nach den Statistikdaten für diese anderen Komponenten sowohl in Linux als auch in Windows sucht.

Sie können den Befehl nur mit dem Namen des Warteschlangenmanagers ausführen und interaktiv durchgehen, um die Parameter **CLASS**, dann **TYPE** und dann **object** auszuwählen, die für den Warteschlangenmanager bei jedem Schritt verfügbar sind. Wenn Sie die **CLASS**-, **TYPE**- und **object**-Namen kennen,

für die Sie Informationen anzeigen wollen, können Sie sie angeben, wenn Sie den Befehl **amqsrua** ausführen.

**-m**

Warteschlangenmanagername. Der WS-Manager muss aktiv sein. Wenn Sie keinen Warteschlangenmanagernamen angeben, werden die Statistiken für den Standardwarteschlangenmanager angezeigt.

**-c**

CLASS-Name. Veröffentlichungen zur IBM MQ-Ressourcennutzung sind einer Klasse zugeordnet. Die Klassen stellen die oberste Ebene in der Baumstruktur der Metadaten dar, die die verfügbaren Informationen zur Ressourcennutzung beschreibt.

Sie können die Klassen **CHANNEL** und **STATUS** nur verwenden, um Veröffentlichungen anzuzeigen, die für Warteschlangenmanager empfangen werden, die mit IBM MQ Bridge to Salesforce verwendet werden. Sie müssen den Parameter **-p** mit dem Metadatenpräfix "**\\$SYS/Application/runmqsfb**" angeben, wenn Sie den Befehl **amqsrua** ausführen, um die Informationen für die Klassen **CHANNEL** und **STATUS** anzuzeigen.

#### **STATUS**

Gibt Informationen zum Status des IBM MQ Bridge to Salesforces zurück.

#### **CHANNEL**

Gibt Informationen zu Salesforce-Ereignissen zurück.

**-t**

TYPE name. Veröffentlichungen zur IBM MQ-Ressourcennutzung sind einem Typ innerhalb einer Klasse zugeordnet. Jede Veröffentlichung enthält die Klasse und den Typ, die es ermöglichen, die class/type/element-Definitionen zu finden und die resultierenden Veröffentlichungen zu bearbeiten. Die class/type/element-Beschreibungen werden als Metadaten beim Start des Warteschlangenmanagers veröffentlicht.

Die Klassen **STATUS** und **CHANNEL** sind verfügbar, wenn Sie das Programm **amqsrua** für einen Warteschlangenmanager ausführen, der mit der IBM MQ Bridge to Salesforce verwendet wird.

Für die Klasse **STATUS** .

#### **PUSHTOPIC**

Salesforce-Push-Themen.

#### **Ereignisse**

Salesforce-Plattformereignisse.

#### **MQPE**

Mit dieser Option können Sie Informationen zu den von IBM MQ erstellten Plattformereignissen anzeigen. Wenn Sie diese Option auswählen, werden die folgenden Daten angezeigt:

- Gesamtzahl der von IBM MQ erstellten Plattformereignisse, die in einem Intervall verarbeitet werden (in der Struktur STATUS/MQPE).
- Eindeutige Anzahl der von IBM MQ erstellten Plattformereignisse, die in diesem Intervall angezeigt werden.
- Anzahl der fehlgeschlagenen Veröffentlichungen der von IBM MQ erstellten Plattformereignisse, die in diesem Intervall angezeigt werden.

Für die Klasse **CHANNEL** .

#### **Ereignisse**

Die Gesamtzahl der Veröffentlichungen für ein bestimmtes Salesforce-Push-Thema oder -Plattformereignis. Sie müssen den Objektnamen für den Klassentyp (CHANNEL) (EVENTS) angeben, z. B. /topic/push\_topic\_name oder /event/platform\_event\_name.

**-o**

Objektnamen Bei den Veröffentlichungen zur Ressourcennutzung handelt es sich um PCF-Nachrichten, die aus einer Folge von PCF-Elementen bestehen. Die PCF-Elemente, die für jedes Paar vom Typ/Typ veröffentlicht werden, werden in den Metadaten angezeigt. Ein Blatt wird in der Baumstruktur gespeichert, in der jedes Element beschrieben wird, wodurch die Elemente verarbeitet werden können.

- p** Metadatenpräfix. Geben Sie einen Ausgangspunkt für die Themenstruktur an, an dem **amqsrua** nach Statistikdaten suchen kann, die von Warteschlangenmanagern veröffentlicht werden. Die Standardthemenbaumstruktur ist \$SYS/MQ/INFO/QMGR, aber andere Komponenten oder Anwendungen können Statistikdaten unter einem anderen Ausgangspunkt für eine Themenbaumstruktur veröffentlichen. IBM MQ Bridge to Salesforce veröffentlicht die Statistik beispielsweise unter \$SYS/Application/runmqsfb.
- n** Veröffentlichungszähler. Sie können angeben, wie viele Berichte zurückgegeben werden sollen, bevor der Befehl beendet wird. Die Daten werden ungefähr alle zehn Sekunden veröffentlicht. Wenn Sie also den Wert 50 eingeben, gibt der Befehl 50 Berichte über 500 Sekunden zurück. Wenn Sie diesen Parameter nicht angeben, wird der Befehl ausgeführt, bis entweder ein Fehler auftritt oder der WS-Manager heruntergefahren wird.
- h** Verwendung

## Vorgehensweise

1. Wechseln Sie in das Beispielverzeichnis `MQ_INSTALLATION_PATH/samp/bin`.
2. Geben Sie den Befehl **amqsrua** mit dem Parameter **-p** aus, um auf das Thema "`\$SYS/Application/runmqsfb`" zu verweisen, in dem der Warteschlangenmanager Überwachungsinformationen für IBM MQ Bridge to Salesforce-Push-Themen und -Plattformereignisse veröffentlicht:

```
./amqsrua -m QM1 -p "\$SYS/Application/runmqsfb"
```

```
STATUS : Bridge Status
CHANNEL : Salesforce Event Name
Enter Class selection
==>
```

3. Klassenauswahl eingeben CHANNEL

```
==> CHANNEL
```

```
STATUS : Bridge Status
CHANNEL : Salesforce Event Name
Enter Class selection
==> CHANNEL
EVENTS : Events
Enter Type selection
==>
```

4. Geben Sie die Typauswahl ein EVENTS

```
==> EVENTS
```

```
STATUS : Bridge Status
CHANNEL : Salesforce Event Name
Enter Class selection
==> CHANNEL
EVENTS : Events
Enter Type selection
==> EVENTS
An object name is required for Class(CHANNEL) Type(EVENTS)
Enter object name
==>
```

5. Geben Sie den Objektnamen ein, z. B.

```
==> /topic/push_topic_name
```

```

STATUS : Bridge Status
CHANNEL : Salesforce Event Name
Enter Class selection
==> CHANNEL
EVENTS : Events
Enter Type selection
==> EVENTS
An object name is required for Class(CHANNEL) Type(EVENTS)
Enter object name
==> /topic/push_topic_name
Publication received PutDate:20170215 PutTime:14513762 Interval:15.047 seconds
Total Events 1

```

## Ergebnisse

Sie haben das Programm `amqsrua` interaktiv verwendet, indem Sie die Optionen für Klasse, Typ und Objekt eingegeben haben, um die Gesamtzahl der Ereignisse für das angegebene Push-Thema anzuzeigen.

In diesem Beispiel können Sie die `amqsrua` verwenden, um die Anzahl der Salesforce-Push-Themen zu überprüfen:

```

./amqsrua -m QM1 -p "\$SYS/Application/runmqsfb"
STATUS : Bridge Status
CHANNEL : Salesforce Event Name
Enter Class selection
==> STATUS
PUSHTOPIC : Push Topics
EVENTS : Platform Events
Enter Type selection
==> PUSHTOPIC
Publication received PutDate:20170315 PutTime:10123086 Interval:15.016 seconds
Total PushTopics 6
Unique PushTopics 3

Publication received PutDate:20170315 PutTime:10124588 Interval:15.012 seconds
Total PushTopics 14 1/sec
Unique PushTopics 7

```

In diesem Beispiel können Sie die `amqsrua` verwenden, um die Anzahl der von IBM MQ erstellten Plattformereignisse zu überprüfen.

```

./amqsrua -m QM1 -p "\$SYS/Application/runmqsfb"
STATUS : Bridge Status
CHANNEL : Salesforce Event Name
Enter Class selection
==> STATUS
PUSHTOPIC : Pushtopics
EVENTS : Platform Events
MQPE : MQ-created Platform Events
Enter Type selection
==> MQPE
Publication received PutDate:20170913 PutTime:12201271 Interval:15.018 seconds
Total MQ-created Platform Events 1
Unique MQ-created Platform Events 1
Failed publications of MQ-created Platform Events 0

```

## Zugehörige Tasks

[IBM MQ für die Verwendung mit Push-Themen und Plattformereignissen für Salesforce konfigurieren](#)

[„Systemressourcennutzung mit dem Befehl `amqsrua` überwachen“ auf Seite 299](#)

Sie können den Befehl **`amqsrua`** verwenden, um Leistungsdaten abzufragen, die sich auf die Systemressourcennutzung eines Warteschlangenmanagers beziehen.

[Vorbereiten und Ausführen von Beispielprogrammen unter Windows](#)

[Beispielprogramme unter UNIX und Linux vorbereiten und ausführen](#)

## Zugehörige Verweise

[runmqsfb \(IBM MQ Bridge to Salesforce ausführen\)](#)

## Real-time Monitoring

Die Real-time-Überwachung ist eine Technik, mit der Sie den aktuellen Status von Warteschlangen und Kanälen in einem Warteschlangenmanager ermitteln können. Die zurückgegebenen Informationen sind zum Zeitpunkt der Befehlsablaufinformationen korrekt.

Es stehen eine Reihe von Befehlen zur Verfügung, die beim Zurückgeben von Echtzeitinformationen zu Warteschlangen und Kanälen verfügbar sind. Informationen können für eine oder mehrere Warteschlangen oder Kanäle zurückgegeben werden und können in der Menge variieren. Die Echtzeit-Überwachung kann in den folgenden Tasks verwendet werden:

- Systemadministratoren den stabilen Zustand ihres IBM MQ-Systems erläutern. Dies hilft bei der Fehlerdiagnose, wenn ein Problem im System auftritt.
- Ermitteln Sie die Bedingung Ihres Warteschlangenmanagers zu jedem Zeitpunkt, selbst wenn kein bestimmtes Ereignis oder Problem festgestellt wurde.
- Unterstützung bei der Bestimmung der Ursache eines Problems in Ihrem System.

Bei der Echtzeitüberwachung können Informationen für Warteschlangen oder Kanäle zurückgegeben werden. Der Umfang der zurückgegebenen Echtzeitinformationen wird durch die Attribute des Warteschlangenmanagers, der Warteschlange und des Kanals gesteuert.

- Sie überwachen eine Warteschlange, indem Sie Befehle absetzen, um sicherzustellen, dass die Warteschlange ordnungsgemäß verarbeitet wird. Bevor Sie einige der Warteschlangenattribute verwenden können, müssen Sie sie für die Echtzeitüberwachung aktivieren.
- Sie können einen Kanal überwachen, indem Sie Befehle absetzen, um sicherzustellen, dass der Kanal ordnungsgemäß ausgeführt wird. Bevor Sie einige der Kanalattribute verwenden können, müssen Sie sie für die Echtzeitüberwachung aktivieren.

Die Echtzeit-Überwachung für Warteschlangen und Kanäle erfolgt zusätzlich zu und getrennt von der Leistungsüberwachung und der Kanalereignisüberwachung.

### Attribute zur Steuerung der Echtzeitüberwachung

Einige Warteschlangen- und Kanalstatusattribute enthalten Überwachungsdaten, wenn die Echtzeitüberwachung aktiviert ist. Wenn die Echtzeitüberwachung nicht aktiviert ist, werden in diesen Überwachungsattributen keine Überwachungsdaten festgehalten. In den Beispielen wird gezeigt, wie Sie diese Warteschlangen- und Kanalstatusattribute verwenden können.

Sie können die Echtzeitüberwachung für einzelne Warteschlangen oder Kanäle oder für mehrere Warteschlangen oder Kanäle aktivieren oder inaktivieren. Wenn Sie einzelne Warteschlangen oder Kanäle steuern möchten, setzen Sie das Warteschlangenattribut MONQ oder das Kanalattribut MONCHL, um die Echtzeitüberwachung zu aktivieren oder zu inaktivieren. Um viele Warteschlangen oder Kanäle gemeinsam zu steuern, aktivieren oder inaktivieren Sie die Echtzeitüberwachung auf WS-Managerebene unter Verwendung der WS-Manager-Attribute MONQ und MONCHL. Für alle Warteschlangen- und Kanalobjekte mit einem Überwachungsattribut, das mit dem Standardwert QMGR angegeben wird, wird die Echtzeitüberwachung auf WS-Managerebene gesteuert.

Automatisch definierte Clustersenderkanäle sind keine IBM MQ-Objekte und weisen daher keine Attribute auf dieselbe Weise wie Kanalobjekte auf. Wenn Sie automatisch definierte Clustersenderkanäle steuern möchten, verwenden Sie das WS-Manager-Attribut MONACLS. Dieses Attribut bestimmt, ob automatisch definierte Clustersenderkanäle innerhalb eines Warteschlangenmanagers für die Kanalüberwachung aktiviert oder inaktiviert sind.

Für die Echtzeitüberwachung von Kanälen können Sie das Attribut MONCHL auf eine der drei Überwachungsstufen setzen: niedrig, mittel oder hoch. Sie können die Überwachungsstufe entweder auf der Objektebene oder auf der Ebene des Warteschlangenmanagers festlegen. Die Auswahl der Ebene hängt von Ihrem System ab. Für die Erfassung von Überwachungsdaten sind möglicherweise einige Anweisungen erforderlich, die relativ teuer sind, wie z. B. die Systemzeit. Um die Auswirkungen der Echtzeitüberwachung zu reduzieren, messen die Optionen "mittel" und "niedrig" eine Stichprobe der Daten in regelmäßigen Intervallen und erfassen nicht die gesamte Zeit. In [Tabelle 30 auf Seite 311](#) sind die Überwachungsstufen zusammengefasst, die für die Echtzeitüberwachung von Kanälen verfügbar sind:

Tabelle 30. Überwachungsstufen

Stufe	Beschreibung	Verwendung
Niedrig	Messen Sie in regelmäßigen Abständen eine kleine Stichprobe der Daten.	Für Objekte, die ein hohes Volumen an Nachrichten verarbeiten.
Mittel	Messen Sie in regelmäßigen Abständen ein Beispiel für die Daten.	Für die meisten Objekte.
Hoch	Messen Sie alle Daten in regelmäßigen Intervallen.	Für Objekte, die nur wenige Nachrichten pro Sekunde verarbeiten, auf denen die meisten aktuellen Informationen wichtig sind.

Für die Echtzeitüberwachung von Warteschlangen können Sie das Attribut MONQ auf eine der drei Überwachungsstufen (niedrig, mittel oder hoch) setzen. Es gibt jedoch keinen Unterschied zwischen diesen Werten. Die Werte für alle aktivieren die Datenerfassung, haben jedoch keinen Einfluss auf die Größe der Stichprobe.

## Beispiele

Die folgenden Beispiele veranschaulichen, wie die erforderlichen Warteschlangen-, Kanal- und WS-Manager-Attribute für die Steuerung der Überwachungsstufe festgelegt werden. Für alle Beispiele gilt, wenn die Überwachung aktiviert ist, die Warteschlange und die Kanalobjekte eine mittlere Überwachungsstufe aufweisen.

1. Verwenden Sie die folgenden Befehle, um die Warteschlangen- und Kanalüberwachung für alle Warteschlangen und Kanäle auf WS-Manager-Ebene zu aktivieren:

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(QMGR)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(QMGR)
```

2. Verwenden Sie die folgenden Befehle, um die Überwachung für alle Warteschlangen und Kanäle zu aktivieren, mit Ausnahme der lokalen Warteschlange, des Q1 und des Senderkanals QM1 . TO . QM2:

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(OFF)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(OFF)
```

3. Verwenden Sie die folgenden Befehle, um die Warteschlangen- und Kanalüberwachung für alle Warteschlangen und Kanäle zu inaktivieren, mit Ausnahme der lokalen Warteschlange, des Q1 s und des Senderkanals QM1 . TO . QM2:

```
ALTER QMGR MONQ(OFF) MONCHL(OFF)
ALTER QL(Q1) MONQ(MEDIUM)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(MEDIUM)
```

4. Verwenden Sie den folgenden Befehl, um die Warteschlangen- und Kanalüberwachung für alle Warteschlangen und Kanäle zu inaktivieren, und zwar unabhängig von den einzelnen Objektattributen:

```
ALTER QMGR MONQ(NONE) MONCHL(NONE)
```

5. Verwenden Sie den folgenden Befehl, um die Überwachungsfunktionen von automatisch definierten Clustersenderkanälen zu steuern:

```
ALTER QMGR MONACLS(MEDIUM)
```

6. Verwenden Sie den folgenden Befehl, um anzugeben, dass automatisch definierte Clustersenderkanäle die Warteschlangenmanagereinstellung für die Kanalüberwachung verwenden sollen:

```
ALTER QMGR MONACLS(QMGR)
```

### Zugehörige Konzepte

„Real-time Monitoring“ auf Seite 310

Die Real-time-Überwachung ist eine Technik, mit der Sie den aktuellen Status von Warteschlangen und Kanälen in einem Warteschlangenmanager ermitteln können. Die zurückgegebenen Informationen sind zum Zeitpunkt der Befehlsablaufinformationen korrekt.

„IBM MQ-Onlineüberwachung verwenden“ auf Seite 334

Sie können Überwachungsdaten für Warteschlangen und Kanäle (einschließlich automatisch definierter Cluster-Server-Kanäle) erfassen, indem Sie die Attribute MONQ, MONCHL und MONACLS definieren.

### Zugehörige Tasks

„Warteschlangen-und Kanalüberwachungsdaten anzeigen“ auf Seite 312

Wenn Sie die Echtzeitüberwachungsdaten für eine Warteschlange oder einen Kanal anzeigen möchten, verwenden Sie den Befehl IBM MQ Explorer oder den entsprechenden MQSC-Befehl. In einigen Überwachungsfeldern wird ein durch Kommas getrennter Paar von Bezugswerten angezeigt, die Ihnen helfen, die Operation Ihres WS-Managers zu überwachen. In den Beispielen wird gezeigt, wie Überwachungsdaten angezeigt werden können.

Warteschlangenmanagerattribute anzeigen und ändern

Überwachung (MONCHL)

## Warteschlangen-und Kanalüberwachungsdaten anzeigen

Wenn Sie die Echtzeitüberwachungsdaten für eine Warteschlange oder einen Kanal anzeigen möchten, verwenden Sie den Befehl IBM MQ Explorer oder den entsprechenden MQSC-Befehl. In einigen Überwachungsfeldern wird ein durch Kommas getrennter Paar von Bezugswerten angezeigt, die Ihnen helfen, die Operation Ihres WS-Managers zu überwachen. In den Beispielen wird gezeigt, wie Überwachungsdaten angezeigt werden können.

### Informationen zu diesem Vorgang

Überwachungsfelder, die ein Paar von Werten anzeigen, die durch ein Komma voneinander getrennt sind, stellen kurz- und langfristige Indikatoren für die Zeit zur Verfügung, die seit der Überwachung für das Objekt oder von dem Zeitpunkt, an dem der Warteschlangenmanager gestartet wurde, aktiviert wurde:

- Der Kurzzeitindikator ist der erste Wert im Paar und wird so berechnet, dass aktuellere Messungen eine höhere Gewichtung erhalten und einen größeren Einfluss auf diesen Wert haben. Dies gibt einen Hinweis auf den jüngsten Trend bei den durchgeführten Messungen.
- Der Langzeitindikator in dem zweiten Wert im Paar und wird so berechnet, dass aktuellere Messungen nicht so hoch gewichtet werden. Dies gibt einen Hinweis auf die längerfristige Aktivität bei der Leistung einer Ressource.

Diese Indikatorwerte sind am nützlichsten, um Änderungen in der Operation Ihres WS-Managers zu erkennen. Dies erfordert die Kenntnis der Zeit, die diese Indikatoren zeigen, wenn sie im normalen Gebrauch ist, um zu erkennen, dass in diesen Zeiten zu erkennen ist. Indem Sie diese Werte regelmäßig erfassen und überprüfen, können Sie Schwankungen in der Operation Ihres Warteschlangenmanagers erkennen. Dies kann auf eine Änderung der Leistung hinweisen.

Gehen Sie wie folgt vor, um Echtzeitüberwachungsdaten zu erhalten:

### Vorgehensweise

1. Wenn Sie die Echtzeitüberwachungsdaten für eine Warteschlange anzeigen möchten, verwenden Sie den IBM MQ Explorer oder den MQSC-Befehl `DISPLAY QSTATUS`, mit dem der optionale Parameter `MONITOR` angegeben wird.

2. Wenn Sie die Echtzeitüberwachungsdaten für einen Kanal anzeigen möchten, verwenden Sie den IBM MQ Explorer oder den MQSC-Befehl `DISPLAY CHSTATUS`, mit dem der optionale Parameter `MONITOR` angegeben wird.

### Beispiel

In der Warteschlange `Q1` ist das Attribut `MONQ` auf den Standardwert `QMGR` gesetzt, und der Warteschlangenmanager, der Eigner der Warteschlange ist, hat das Attribut `MONQ` auf `MEDIUM` gesetzt. Verwenden Sie den folgenden Befehl, um die Überwachungsfelder anzuzeigen, die für diese Warteschlange erfasst wurden:

```
DISPLAY QSTATUS(Q1) MONITOR
```

Die Überwachungsfelder und die Überwachungsstufe der Warteschlange `Q1` werden wie folgt angezeigt:

```
QSTATUS(Q1)
TYPE(Queue)
MONQ(MEDIUM)
QTIME(11892157,24052785)
MSGAGE(37)
LPUTDATE(2005-03-02)
LPUTTIME(09.52.13)
LGETDATE(2005-03-02)
LGETTIME(09.51.02)
```

Der Senderkanal (`QM1.TO.QM2`) hat das Attribut `MONCHL` auf den Standardwert `QMGR` gesetzt, und der Warteschlangenmanager, der Eigner der Warteschlange ist, hat das Attribut `MONCHL` auf `MEDIUM` gesetzt. Verwenden Sie den folgenden Befehl, um die Überwachungsfelder anzuzeigen, die für diesen Senderkanal erfasst wurden:

```
DISPLAY CHSTATUS(QM1.TO.QM2) MONITOR
```

Die Überwachungsfelder und die Überwachungsstufe des Senderkanals `QM1.TO.QM2` werden wie folgt angezeigt:

```
CHSTATUS(QM1.TO.QM2)
XMITQ(Q1)
CONNNAME(127.0.0.1)
CURRENT
CHLTYPE(SDR)
STATUS(RUNNING)
SUBSTATE(MQGET)
MONCHL(MEDIUM)
XQTIME(755394737,755199260)
NETTIME(13372,13372)
EXITTIME(0,0)
XBATCHSZ(50,50)
COMPTIME(0,0)
STOPREQ(NO)
RQNAME(QM2)
```

### Zugehörige Konzepte

„Real-time Monitoring“ auf Seite 310

Die Real-time-Überwachung ist eine Technik, mit der Sie den aktuellen Status von Warteschlangen und Kanälen in einem Warteschlangenmanager ermitteln können. Die zurückgegebenen Informationen sind zum Zeitpunkt der Befehlsablaufinformationen korrekt.

### Zugehörige Verweise

[ANZEIGEN QSTATUS](#)

## Warteschlangen überwachen

Verwenden Sie diese Seite, um Tasks anzuzeigen, die Sie bei der Behebung eines Problems mit einer Warteschlange und der Anwendung, die diese Services in Warteschlange stellt, unterstützen. Es stehen verschiedene Überwachungsoptionen zur Verfügung, um das Problem zu ermitteln.

Häufig ist das erste Anzeichen eines Problems mit einer Warteschlange, die gewartet wird, dass die Anzahl der Nachrichten in der Warteschlange (CURDEPTH) zunimmt. Wenn Sie eine Erhöhung zu bestimmten Tageszeiten oder unter bestimmten Workloads erwarten, kann eine steigende Anzahl von Nachrichten nicht auf ein Problem hinweisen. Wenn Sie jedoch keine Erklärung für die steigende Anzahl von Nachrichten haben, können Sie die Ursache möglicherweise untersuchen.

Es kann eine Anwendungswarteschlange vorhanden sein, in der ein Problem mit der Anwendung auftritt, oder eine Übertragungswarteschlange, in der ein Problem mit dem Kanal aufgetreten ist. Es sind zusätzliche Überwachungsoptionen verfügbar, wenn die Anwendung, die die Warteschlange in die Warteschlange stellt, ein Kanal ist.

Die folgenden Beispiele untersuchen Probleme mit einer bestimmten Warteschlange (Q1) und beschreiben die Felder, die in der Ausgabe der verschiedenen Befehle angezeigt werden:

### ***Feststellen, ob Ihre Anwendung die Warteschlange geöffnet hat***

Wenn Sie ein Problem mit einer Warteschlange haben, überprüfen Sie, ob Ihre Anwendung die Warteschlange geöffnet hat.

### **Informationen zu diesem Vorgang**

Führen Sie die folgenden Schritte aus, um zu ermitteln, ob Ihre Anwendung die Warteschlange geöffnet hat:

### **Vorgehensweise**

1. Stellen Sie sicher, dass die Anwendung, die für die Warteschlange ausgeführt wird, die Anwendung ist, die Sie erwarten. Geben Sie den folgenden Befehl für die Warteschlange aus:

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) ALL
```

Sehen Sie sich in der Ausgabe das Feld APPLTAG an, und überprüfen Sie, ob der Name Ihrer Anwendung angezeigt wird. Wenn der Name Ihrer Anwendung nicht angezeigt wird oder wenn überhaupt keine Ausgabe vorhanden ist, starten Sie Ihre Anwendung.

2. Wenn es sich bei der Warteschlange um eine Übertragungswarteschlange handelt, sehen Sie die Ausgabe im Feld CHANNEL an.

Wenn der Kanalname nicht im Feld CHANNEL angezeigt wird, stellen Sie fest, ob der Kanal aktiv ist.

3. Stellen Sie sicher, dass die Anwendung, die für die Warteschlange ausgeführt wird, die Warteschlange für die Eingabe geöffnet hat. Geben Sie den folgenden Befehl ein:

```
DISPLAY QSTATUS(Q1) TYPE(QUEUE) ALL
```

Sehen Sie sich in der Ausgabe das Feld IPPROCS an, um festzustellen, ob eine Anwendung die Warteschlange für die Eingabe geöffnet hat. Wenn der Wert 0 ist und es sich um eine Benutzeranwendungswarteschlange handelt, stellen Sie sicher, dass die Anwendung die Warteschlange für die Eingabe öffnet, damit die Nachrichten aus der Warteschlange abgerufen werden.

### ***Überprüfen, ob Nachrichten in der Warteschlange verfügbar sind***

Wenn Sie eine große Anzahl von Nachrichten in der Warteschlange haben und Ihre Anwendung keine dieser Nachrichten verarbeitet, überprüfen Sie, ob die Nachrichten in der Warteschlange für Ihre Anwendung verfügbar sind.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um zu untersuchen, warum Ihre Anwendung keine Nachrichten aus der Warteschlange verarbeitet:

### Vorgehensweise

1. Stellen Sie sicher, dass Ihre Anwendung nicht nach einer bestimmten Nachrichten-ID oder Korrelations-ID fragt, wenn sie alle Nachrichten in der Warteschlange verarbeiten soll.
2. Obwohl die aktuelle Länge der Warteschlange zeigen kann, dass eine wachsende Anzahl von Nachrichten in der Warteschlange vorhanden ist, sind einige Nachrichten in der Warteschlange möglicherweise nicht für eine Anwendung verfügbar, da sie nicht festgeschrieben werden. Die aktuelle Tiefe schließt die Anzahl der nicht festgeschriebenen MQPUTs von Nachrichten in die Warteschlange ein. Geben Sie den folgenden Befehl ein:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Sehen Sie sich in der Ausgabe das UNCOM-Feld an, um festzustellen, ob nicht festgeschriebene Nachrichten in der Warteschlange vorhanden sind.

3. Wenn Ihre Anwendung versucht, Nachrichten aus der Warteschlange abzurufen, überprüfen Sie, ob die Putting-Anwendung die Nachrichten ordnungsgemäß festschreibt. Geben Sie den folgenden Befehl aus, um die Namen der Anwendungen zu ermitteln, die Nachrichten in diese Warteschlange einreihen:

```
DISPLAY QSTATUS(Q1) TYPE(Handle) OPENTYPE(OUTPUT)
```

4. Geben Sie dann den folgenden Befehl aus und setzen Sie in *appltag* den Wert APPLTAG aus der Ausgabe des vorherigen Befehls ein:

```
DISPLAY CONN(*) WHERE(APPLTAG EQ appltag) UOWSTDA UOWSTTI
```

Dies zeigt, wann die Arbeitseinheit gestartet wurde, und hilft Ihnen dabei zu ermitteln, ob die Anwendung eine lange Arbeitseinheit erstellt. Wenn es sich bei der Putting-Anwendung um einen Kanal handelt, möchten Sie möglicherweise untersuchen, warum eine Stapelverarbeitung eine lange Zeit in Anspruch nimmt.

## Überprüfen, ob die Anwendung Nachrichten aus der Warteschlange erhält

Wenn Sie ein Problem mit einer Warteschlange und der Anwendung, die diese Warteschlange enthält, haben, überprüfen Sie, ob Ihre Anwendung Nachrichten aus der Warteschlange abstellt.

## Informationen zu diesem Vorgang

Führen Sie die folgenden Prüfungen aus, um zu überprüfen, ob Ihre Anwendung Nachrichten aus der Warteschlange erhält:

### Vorgehensweise

1. Stellen Sie sicher, dass die Anwendung, die für die Warteschlange ausgeführt wird, tatsächlich Nachrichten aus der Warteschlange verarbeitet. Geben Sie den folgenden Befehl ein:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

In der Ausgabe die Felder LGETDATE und LGETTIME anzeigen, die anzeigen, wann der letzte Versuch aus der Warteschlange ausgeführt wurde.

2. Wenn das letzte Abrufen aus dieser Warteschlange zurückliegt als erwartet, stellen Sie sicher, dass die Anwendung Nachrichten ordnungsgemäß verarbeitet.

Wenn es sich bei der Anwendung um einen Kanal handelt, überprüfen Sie, ob die Nachrichten durch diesen Kanal verschoben werden.

## ***Feststellen, ob die Anwendung Nachrichten schnell genug verarbeiten kann***

Wenn Nachrichten in der Warteschlange erstellt werden, die anderen Prüfungen jedoch keine Verarbeitungsfehler gefunden haben, überprüfen Sie, ob die Anwendung Nachrichten schnell genug verarbeiten kann. Wenn es sich bei der Anwendung um einen Kanal handelt, überprüfen Sie, ob der Kanal Nachrichten schnell genug verarbeiten kann.

### **Informationen zu diesem Vorgang**

Führen Sie die folgenden Tests aus, um festzustellen, ob die Anwendung Nachrichten schnell genug verarbeitet:

### **Vorgehensweise**

1. Geben Sie den folgenden Befehl in regelmäßigen Abständen aus, um Leistungsdaten zu der Warteschlange zu erfassen:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Wenn die Werte in den QTIME-Indikatoren hoch sind oder sich im Laufe des Zeitraums erhöhen und Sie bereits die Möglichkeit von Arbeitseinheiten mit langer Laufzeit ausgeschlossen haben, indem Sie überprüfen, ob Nachrichten in der Warteschlange verfügbar sind, wird die Anwendung unter Umständen nicht mit den Putting-Anwendungen in Einklang stehen.

2. Wenn Ihre Anwendung nicht mit den Eintragsanwendungen Schritt halten kann, fügen Sie eine weitere Anwendung zum Verarbeiten der Warteschlange hinzu.

Ob Sie eine weitere Anwendung hinzufügen können, hängt vom Design der Anwendung ab und davon, ob die Warteschlange von mehreren Anwendungen gemeinsam genutzt werden kann. Funktionen wie die Nachrichtengruppierung oder die Korrelations-ID können dazu beitragen, dass zwei Anwendungen gleichzeitig eine Warteschlange verarbeiten können.

## ***Warteschlange überprüfen, wenn die aktuelle Tiefe nicht zunimmt***

Auch wenn die aktuelle Länge der Warteschlange nicht zunimmt, kann es sinnvoll sein, die Warteschlange zu überwachen, um zu überprüfen, ob Ihre Anwendung Nachrichten ordnungsgemäß verarbeitet.

### **Informationen zu diesem Vorgang**

Gehen Sie wie folgt vor, um Leistungsdaten für die Warteschlange zu erfassen: Geben Sie den folgenden Befehl in regelmäßigen

### **Vorgehensweise**

Geben Sie den folgenden Befehl in regelmäßigen Abständen aus:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) MSGAGE QTIME
```

Wenn in der Ausgabe der Wert in MSGAGE im Verlauf der Zeit zunimmt und Ihre Anwendung so konzipiert ist, dass alle Nachrichten verarbeitet werden, kann dies darauf hinweisen, dass einige Nachrichten überhaupt nicht verarbeitet werden.

## **Kanäle überwachen**

Auf dieser Seite können Sie Tasks anzeigen, die Sie bei der Behebung eines Problems mit einer Übertragungswarteschlange und dem Kanal, in dem die Warteschlange ausgeführt wird, unterstützen. Es stehen verschiedene Optionen zur Kanalüberwachung zur Verfügung, um das Problem zu bestimmen.

Häufig ist das erste Anzeichen eines Problems mit einer Warteschlange, die gewartet wird, dass die Anzahl der Nachrichten in der Warteschlange (CURDEPTH) zunimmt. Wenn Sie eine Erhöhung zu bestimmten Tageszeiten oder unter bestimmten Workloads erwarten, kann eine steigende Anzahl von Nachrichten

nicht auf ein Problem hinweisen. Wenn Sie jedoch keine Erklärung für die steigende Anzahl von Nachrichten haben, können Sie die Ursache möglicherweise untersuchen.

Möglicherweise liegt ein Problem mit dem Kanal vor, der eine Übertragungswarteschlange für Services enthält. Es stehen verschiedene Optionen für die Kanalüberwachung zur Verfügung, mit deren Hilfe Sie das Problem bestimmen können.

In den folgenden Beispielen werden Probleme mit einer Übertragungswarteschlange mit dem Namen QM2 und einem Kanal mit dem Namen QM1.TO.QM2 untersucht. Dieser Kanal wird verwendet, um Nachrichten vom WS-Manager QM1 an den Warteschlangenmanager QM2 zu senden. Die Kanaldefinition auf WS-Manager QM1 ist entweder ein Sender- oder Serverkanal, und die Kanaldefinition im Warteschlangenmanager QM2 ist entweder ein Empfänger- oder Requesterkanal.

### **Feststellen, ob der Kanal aktiv ist**

Wenn Sie ein Problem mit einer Übertragungswarteschlange haben, überprüfen Sie, ob der Kanal aktiv ist.

### **Informationen zu diesem Vorgang**

Führen Sie die folgenden Schritte aus, um den Status des Kanals zu überprüfen, der die Übertragungswarteschlange bedient:

### **Vorgehensweise**

1. Geben Sie den folgenden Befehl aus, um herauszufinden, welcher Kanal die Übertragungswarteschlange QM2 verarbeiten soll:

```
DIS CHANNEL(*) WHERE(XMITQ EQ QM2)
```

In diesem Beispiel zeigt die Ausgabe dieses Befehls an, dass der Kanal, der die Übertragungswarteschlange bedient, QM1.TO.QM2 ist.

2. Geben Sie den folgenden Befehl aus, um den Status des Kanals, QM1.TO.QM2, zu bestimmen:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

3. Überprüfen Sie das Feld STATUS in der Ausgabe des Befehls **CHSTATUS** :
  - Wenn der Wert im Feld STATUS den Wert RUNNING hat, überprüfen Sie, ob der Kanal Nachrichten bewegt.
  - Wenn die Ausgabe des Befehls keinen Status anzeigt oder der Wert des Felds STATUS den Wert STOPPED, RETRY, BINDING oder REQUESTING hat, führen Sie den entsprechenden Schritt wie folgt aus:
4. Optional: Wenn der Wert im Feld STATUS keinen Status anzeigt, ist der Kanal inaktiv. Führen Sie daher die folgenden Schritte aus:
  - a) Wenn der Kanal automatisch durch einen Auslöser gestartet worden sein sollte, überprüfen Sie, ob die Nachrichten in der Übertragungswarteschlange verfügbar sind.  
Wenn in der Übertragungswarteschlange Nachrichten vorhanden sind, überprüfen Sie, ob die Auslöseereinstellungen in der Übertragungswarteschlange richtig sind.
  - b) Geben Sie den folgenden Befehl aus, um den Kanal erneut manuell zu starten:

```
START CHANNEL(QM1.TO.QM2)
```

5. Optional: Wenn der Wert des Felds STATUS STOPPED lautet, führen Sie die folgenden Schritte aus:
  - a) Überprüfen Sie die Fehlerprotokolle, um festzustellen, warum der Kanal gestoppt wurde. Wenn der Kanal aufgrund eines Fehlers gestoppt wurde, beheben Sie das Problem.  
Stellen Sie sicher, dass für den Kanal auch Werte angegeben sind, die für die Wiederholungsattribute angegeben sind: *SHORTRTY* und *LONGRTY*. Bei transienten Fehlern, wie z. B. Netzfehlern, wird der Kanal dann automatisch erneut gestartet.

b) Geben Sie den folgenden Befehl aus, um den Kanal erneut manuell zu starten:

```
START CHANNEL(QM1.TO.QM2)
```

**z/OS** Unter IBM MQ for z/OS können Sie ermitteln, ob ein Benutzer einen Kanal mithilfe von Befehlsereignisnachrichten stoppt.

6. Optional: Wenn der Wert des Felds STATUS RETRY lautet, führen Sie die folgenden Schritte aus:

- a) Überprüfen Sie die Fehlerprotokolle, um den Fehler zu identifizieren, und beheben Sie das Problem.
- b) Geben Sie den folgenden Befehl aus, um den Kanal erneut manuell zu starten:

```
START CHANNEL(QM1.TO.QM2)
```

oder Warten Sie, bis der Kanal bei der nächsten Wiederholung erfolgreich eine Verbindung hergestellt hat.

7. Optional: Wenn der Wert des Felds STATUS BINDING oder REQUESTING ist, ist der Kanal noch nicht erfolgreich mit dem Partner verbunden. Führen Sie die folgenden Schritte aus:

- a) Geben Sie den folgenden Befehl an beiden Enden des Kanals aus, um den Unterstatus des Kanals zu bestimmen:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

**Anmerkung:**

- i) In einigen Fällen kann es zu einem Substatus nur an einem Ende des Kanals kommen.
  - ii) Viele Unterzustände sind vorübergehend, daher setzen Sie den Befehl ein paar Mal ab, um festzustellen, ob ein Kanal in einem bestimmten Unterzustand blockiert ist.
- b) Überprüfen Sie [Tabelle 31 auf Seite 318](#), um festzustellen, welche Aktion zu ergreifen ist:

<i>Tabelle 31. Substatus mit Statusbindung oder Anforderung angezeigt</i>		
<b>MCA-Unterzustand initialisieren <sup>1</sup></b>	<b>Antwortet MCA-Unterstatus <sup>2</sup></b>	<b>Anmerkungen</b>
NAMESERVER		Der einleitende MCA wartet darauf, dass eine Name-Server-Anforderung ausgeführt wird. Stellen Sie sicher, dass der richtige Hostname im Kanalattribut CONNAME angegeben wurde und dass Ihre Name-Server ordnungsgemäß konfiguriert sind.
SCYEXIT	SCYEXIT	Die Nachrichtenkanalagenten (MCAs) sind momentan <i>im Datenaustausch</i> über einen Sicherheitsexit. Weitere Informationen finden Sie im Abschnitt „Feststellen, ob der Kanal Nachrichten schnell genug verarbeiten kann“ auf Seite 321.
	CHADEXIT	Der Exit für die Kanalautodefinition wird gerade ausgeführt. Weitere Informationen finden Sie im Abschnitt „Feststellen, ob der Kanal Nachrichten schnell genug verarbeiten kann“ auf Seite 321.
RCVEXIT SENDEXIT MSGEXIT MREXIT	RCVEXIT SENDEXIT MSGEXIT MREXIT	Die Exits werden beim Kanalstart für MQXR_INIT aufgerufen. Sehen Sie sich die Verarbeitung in diesem Teil des Exits an, wenn dies lange dauert. Weitere Informationen finden Sie im Abschnitt „Feststellen, ob der Kanal Nachrichten schnell genug verarbeiten kann“ auf Seite 321.
SERIALIZE	SERIALIZE	Dieser Unterzustand gilt nur für Kanäle mit der Disposition SHARED.

Tabelle 31. Substatus mit Statusbindung oder Anforderung angezeigt (Forts.)		
MCA-Unterzustand initialisieren <sup>1</sup>	Antwortet MCA-Unterstatus <sup>2</sup>	Anmerkungen
NETZVERBIND		Dieser Unterzustand wird angezeigt, wenn die Verbindung aufgrund einer falschen Netzkonfiguration verzögert wird.
SSLHANDSHAKE	SSLHANDSHAKE	Ein TLS-Handshake besteht aus einer Reihe von gesendeten und empfangenen Nachrichten. Wenn die Netzzeiten langsam sind oder die Verbindung zu den Lookup-CRLs langsam ist, wirkt sich dies auf die Zeit aus, die für den Handshake zur Hand genommen wird.   Unter IBM MQ for z/OS kann dieser Unterzustand auch anzeigen, dass der Wert von SSLTASKS nicht ausreicht.

#### Anmerkungen:

- i) Der einleitende MCA ist das Ende des Kanals, der den Dialog gestartet hat. Dabei kann es sich um Absender, Cluster-Absender, vollständig qualifizierte Server und Anforderer handeln. In einem Server-Requester-Paar ist es das Ende, von dem aus Sie den Kanal gestartet haben.
- ii) Der antwortende MCA ist das Ende des Kanals, das auf die Anforderung zum Starten des Datenaustauschs geantwortet hat. Dies kann Empfänger, Cluster-Empfänger, Requester (wenn der Server oder Sender gestartet wird), Server (wenn der Requester gestartet wird) und Absender (in einem Requester-Sender-Callback-Paar von Kanälen).

### Überprüfen, ob der Kanal Nachrichten bewegt

Wenn Sie ein Problem mit einer Übertragungswarteschlange haben, überprüfen Sie, ob der Kanal Nachrichten bewegt.

### Vorbereitende Schritte

Geben Sie den Befehl `DIS CHSTATUS(QM1.TO.QM2) ALL` aus. Wenn der Wert des Felds STATUS RUNNING ist, hat der Kanal erfolgreich eine Verbindung zum Partnersystem hergestellt.

Stellen Sie sicher, dass keine nicht festgeschriebenen Nachrichten in der Übertragungswarteschlange vorhanden sind, wie in [„Überprüfen, ob Nachrichten in der Warteschlange verfügbar sind“](#) auf Seite 314 beschrieben.

### Informationen zu diesem Vorgang

Wenn es Nachrichten gibt, die für den Kanal verfügbar sind, um zu senden und zu empfangen, führen Sie die folgenden Prüfungen durch:

### Vorgehensweise

1. Zeigen Sie in der Ausgabe des Statusbefehls für den Anzeigekanal (`DIS CHSTATUS(QM1.TO.QM2) ALL`) die folgenden Felder an:

#### MSGS

Die Anzahl der gesendeten oder empfangenen Nachrichten (oder für Serververbindungskanäle die Anzahl der verarbeiteten MQI-Aufrufe) während dieser Sitzung (seit dem Kanalstart).

#### BUFSENT

Anzahl der gesendeten Übertragungspuffer. Dazu gehören auch Übertragungen reiner Steuerinformationen.

#### BYTSENT

Gibt die Anzahl Bytes an, die in der aktuellen Sitzung (seit dem Kanalstart) bereits gesendet wurden. Dazu gehören auch Steuerinformationen, die von dem Nachrichtenkanal gesendet wurden.

### LSTMSGDA

Gibt das Datum an, an dem die letzte Nachricht gesendet oder der letzte MQI-Aufruf ausgeführt wurde (siehe LSTMSGTI).

### LSTMSGTI

Gibt den Zeitpunkt an, zu dem die letzte Nachricht gesendet oder der letzte MQI-Aufruf ausgeführt wurde. Für einen Sender- oder Serverkanal ist dies der Zeitpunkt, zu dem die letzte Nachricht (bzw. im Falle einer Segmentierung das letzte Nachrichtensegment) gesendet wurde. Für einen Requester- oder Empfängerkanal ist dies der Zeitpunkt, zu dem die letzte Nachricht in die entsprechende Bestimmungwarteschlange gestellt wurde. Für einen Serververbindungskanal ist dies der Zeitpunkt, zu dem der letzte MQI-Aufruf abgeschlossen wurde.

### AKTUELLUNGEN

Für einen Senderkanal gibt dieses Attribut die Anzahl der Nachrichten an, die in dem aktuellen Stapel übertragen wurden. Für einen Empfängerkanal gibt dieses Attribut die Anzahl der Nachrichten an, die in dem aktuellen Stapel empfangen wurden. Bei Festschreibung des Stapels wird dieser Wert für Sender- und Empfängerkanäle auf null zurückgesetzt.

### STATUS

Der Status des Kanals. Folgende Werte sind möglich: Wird gestartet, Wird gebunden, Wird initialisiert, Aktiv, Wird gestoppt, Wird wiederholt, Angehalten, Gestoppt oder Wird angefordert.

### SUBSTATE

Die Aktion, die vom Kanal derzeit ausgeführt wird.

### INDOUBT

Gibt an, ob der Kanal momentan unbestätigt ist. Der Wert YES wird hier nur zurückgegeben, solange der sendende Nachrichtenkanalagent auf die Empfangsbestätigung für einen erfolgreich gesendeten Stapel wartet. Andernfalls wird NO zurückgegeben, auch für den Zeitraum, in dem Nachrichten gesendet wurden, jedoch noch keine Empfangsbestätigung angefordert wurde. Für Empfangskanäle wird immer der Wert NO zurückgegeben.

2. Stellen Sie fest, ob der Kanal seit dem Start alle Nachrichten gesendet hat. Wenn eine Nachricht gesendet wurde, stellen Sie fest, wann die letzte Nachricht gesendet wurde.
3. Der Kanal hat möglicherweise einen Stapel gestartet, der noch nicht abgeschlossen wurde, wie durch einen Wert ungleich null in CURMSGs angezeigt wird. Wenn INDOUBT den Wert YES hat, wartet der Kanal auf die Bestätigung, dass der Stapel am anderen Ende des Kanals empfangen wurde. Zeigen Sie das Feld SUBSTATE der Ausgabe an und lesen Sie die Informationen unter [Tabelle 32 auf Seite 320](#):

<b>AbsenderSUBSTATE</b>	<b>Empfänger-SUBSTATE</b>	<b>Hinweise</b>
MQGET	RECEIVE	Normale Zustände eines Kanals in Ruhe.
SENDEN	RECEIVE	SEND ist in der Regel ein transitorischer Status. Wenn SEND angezeigt wird, zeigt dies an, dass die Kommunikationsprotokollpuffer gefüllt sind. Dies kann auf ein Netzproblem hinweisen.
RECEIVE		Wenn der Sender im Unterzustand RECEIVE für einen beliebigen Zeitraum angezeigt wird, wartet er auf eine Antwort, entweder auf einen Batch-Abschluss oder auf einen Heartbeat. Möglicherweise möchten Sie prüfen, warum eine Stapelverarbeitung lange dauert.

**Anmerkung:** Es kann auch festgestellt werden, ob der Kanal Nachrichten schnell genug verarbeiten kann, insbesondere wenn der Kanal einen Unterstatus hat, der der Exitverarbeitung zugeordnet ist.

## **Prüfen, warum ein Stapel lange dauert, bis er abgeschlossen ist**

Gründe, warum ein Stapel lange Zeit in Anspruch nehmen kann, um ein langsames Netz zu schließen, oder ein Kanal verwendet die Nachrichtenwiederholungsverarbeitung.

### **Informationen zu diesem Vorgang**

Wenn ein Senderkanal einen Stapel von Nachrichten gesendet hat, wartet er auf die Bestätigung des Stapels vom Empfänger, es sei denn, der Kanal ist pipeliniert. Die in dieser Task beschriebenen Faktoren können sich darauf auswirken, wie lange der Senderkanal wartet.

### **Prozedur**

- Überprüfen Sie, ob das Netz langsam ist.

Der NETTIME-Wert ist die Zeit, die in Mikrosekunden angezeigt wird, um ein Ende der Stapelanforderung an das ferne Ende des Kanals zu senden und eine Antwort abzüglich der Zeit zu empfangen, bis das Ende der Stapelanforderung verarbeitet wird. Dieser Wert kann aus einem der folgenden Gründe groß sein:

- Das Netz ist langsam. Ein langsames Netz kann sich auf die Zeit auswirken, die benötigt wird, um einen Stapel zu beenden. Die Messungen, die zu den Indikatoren für das NETTIME-Feld führen, werden am Ende eines Stapels gemessen. Der erste von einer Verlangsamung im Netz betroffene Stapel wird jedoch nicht mit einer Änderung des NETTIME-Werts angezeigt, da er am Ende des Stapels gemessen wird.
- Anforderungen werden am fernen Endpunkt in die Warteschlange gestellt. Beispielsweise kann ein Kanal eine Anforderung PUT wiederholen oder eine Anforderung PUT wird möglicherweise aufgrund einer Seitengruppenein-/ausgabe langsam ausgeführt. Sobald alle in der Warteschlange befindlichen Anforderungen vollständig ausgeführt worden sind, wird die Dauer für die Beendigung der Stapelanforderung gemessen. Wenn Sie also einen großen NETTIME-Wert erhalten, überprüfen Sie, ob am fernen Ende eine ungewöhnliche Verarbeitung vorhanden ist.

- Überprüfen Sie, ob der Kanal die Nachrichtenwiederholung verwendet.

Wenn es dem Empfängerkanal nicht gelingt, eine Nachricht in eine Zielwarteschlange zu stellen, kann er die Nachricht erneut verarbeiten, anstatt sie sofort in eine Warteschlange für nicht zustellbare Nachrichten einzureihen. Die Wiederholungsverarbeitung kann dazu führen, dass die Stapelverarbeitung verlangsamt wird. In zwischen MQPUT-Versuchen hat der Kanal STATUS (PAUSED) und gibt an, dass er darauf wartet, dass das Nachrichtenwiederholungsintervall übergeben wird.

## **Feststellen, ob der Kanal Nachrichten schnell genug verarbeiten kann**

Wenn sich Nachrichten in der Übertragungswarteschlange befinden, Sie jedoch keine Verarbeitungsfehler gefunden haben, stellen Sie fest, ob der Kanal Nachrichten schnell genug verarbeiten kann.

### **Vorbereitende Schritte**

Geben Sie den folgenden Befehl mehrmals in einem bestimmten Zeitraum aus, um Leistungsdaten zum Kanal zu erfassen:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

### **Informationen zu diesem Vorgang**

Vergewissern Sie sich, dass keine nicht festgeschriebenen Nachrichten in der Übertragungswarteschlange vorhanden sind, wie in „Überprüfen, ob Nachrichten in der Warteschlange verfügbar sind“ auf Seite 314 beschrieben, und überprüfen Sie anschließend das Feld XQTIME in der Ausgabe des Statusbefehls für den Anzeigenkanal. Wenn die Werte der XQTIME-Indikatoren konstant hoch sind oder sich über den Messzeitraum erhöhen, deutet dies darauf hin, dass der Kanal nicht mit den Putting-Anwendungen mithalten kann.

Führen Sie die folgenden Tests aus:

## Vorgehensweise

### 1. Prüfen Sie, ob die Exits verarbeitet werden.

Wenn auf dem Kanal, der diese Nachrichten liefert, Exits verwendet werden, können sie zu den Nachrichten hinzugefügt werden, die für die Verarbeitung von Nachrichten benötigt werden. Führen Sie die folgenden Prüfungen durch, um zu ermitteln, ob dies der Fall ist:

- a) Überprüfen Sie in der Ausgabe des Befehls `DIS CHSTATUS(QM1.TO.QM2) ALL` das Feld `EXIT-TIME`.

Wenn die in den Exits verbrachte Zeit höher ist als erwartet, überprüfen Sie die Verarbeitung in Ihren Exits auf unnötige Schleifen oder zusätzliche Verarbeitungszeiten, insbesondere in Nachrichten-, Sende- und Empfangsexits. Diese Verarbeitung wirkt sich auf alle Nachrichten aus, die über den Kanal übertragen werden.

- b) Überprüfen Sie in der Ausgabe des Befehls `DIS CHSTATUS(QM1.TO.QM2) ALL` das Feld `SUBSTATE`.

Wenn der Kanal einen der folgenden Unterzustände für einen signifikanten Zeitraum aufweist, überprüfen Sie die Verarbeitung in Ihren Exits:

- SCYEXIT
- RCVEXIT
- SENDEXIT
- MSGEXIT
- MREXIT

Weitere Informationen zum Teilstatus eines Kanals finden Sie in der Tabelle [Tabelle 32 auf Seite 320](#).

### 2. Überprüfen Sie, ob das Netz langsam ist.

Wenn sich die Nachrichten nicht schnell genug über einen Kanal bewegen, kann es sein, dass das Netz langsam ist. Führen Sie die folgenden Prüfungen durch, um zu ermitteln, ob dies der Fall ist:

- a) Überprüfen Sie in der Ausgabe des Befehls `DIS CHSTATUS(QM1.TO.QM2) ALL` das Feld `NET-TIME`.

Diese Indikatoren werden gemessen, wenn der sendende Kanal seinen Partner für eine Antwort auffordert. Dies geschieht am Ende jedes Stapels und, wenn ein Kanal während des Überwachungssignals im Leerlauf ist.

- b) Wenn dieser Indikator zeigt, dass Rundreisen länger als erwartet sind, verwenden Sie andere Netzüberwachungstools, um die Leistung Ihres Netzes zu untersuchen.

### 3. Überprüfen Sie, ob der Kanal die Komprimierung verwendet.

Wenn der Kanal die Komprimierung verwendet, wird dadurch die Zeit für die Verarbeitung von Nachrichten hinzugefügt. Wenn der Kanal nur einen Komprimierungsalgorithmus verwendet, führen Sie die folgenden Prüfungen durch:

- a) Überprüfen Sie in der Ausgabe des Befehls `DIS CHSTATUS(QM1.TO.QM2) ALL` das Feld `COMP-TIME`.

Diese Anzeiger zeigen die Zeit an, die während der Komprimierung oder Dekomprimierung verbracht wurde

- b) Wenn die ausgewählte Komprimierung die Datenmenge, die von der erwarteten Menge gesendet werden soll, nicht reduziert, ändern Sie den Komprimierungsalgorithmus.

### 4. Wenn der Kanal mehrere Komprimierungsalgorithmen verwendet, führen Sie die folgenden Prüfungen durch:

- a) Überprüfen Sie in der Ausgabe des Befehls `DIS CHSTATUS(QM1.TO.QM2) ALL` die Felder `COMP-TIME`, `COMPHDR` und `COMPMSG`.

- b) Ändern Sie die Komprimierungsalgorithmen, die in der Kanaldefinition angegeben sind, oder erwägen Sie, einen Nachrichtenexit zu schreiben, um die Auswahl des Komprimierungsalgorithmus

des Kanals für bestimmte Nachrichten zu überschreiben, wenn die Komprimierungsrate oder die Auswahl des Algorithmus die erforderliche Komprimierung oder Leistung nicht bereitstellt.

### **Probleme mit Clusterkanälen beheben**

Wenn Sie einen Aufbau von Nachrichten in der Warteschlange SYSTEM.CLUSTER.TRANSMIT.QUEUE haben, wird der erste Schritt bei der Diagnose des Problems erkannt, welcher Kanal oder welche Kanäle ein Problem mit der Zustellung von Nachrichten haben.

### **Informationen zu diesem Vorgang**

Um zu ermitteln, welche Kanäle oder Kanäle die SYSTEM.CLUSTER.TRANSMIT.QUEUE verwenden, ist ein Problem beim Zustellen von Nachrichten aufgetreten. Führen Sie die folgenden Prüfungen durch:

### **Vorgehensweise**

1. Geben Sie den folgenden Befehl ein:

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

**Anmerkung:** Wenn Sie einen ausgelasteten Cluster haben, der viele Nachrichten in Bewegung hat, sollten Sie diesen Befehl mit einer höheren Zahl ausgeben, um die Kanäle zu eliminieren, die nur wenige Nachrichten zur Verfügung haben.

2. Sehen Sie sich die Ausgabe für den Kanal oder die Kanäle an, die über große Werte im Feld XQMSGSA verfügen. Stellen Sie fest, warum der Kanal keine Nachrichten bewegt, oder versetzen Sie sie nicht schnell genug. Verwenden Sie die in „Kanäle überwachen“ auf Seite 316 beschriebenen Tasks, um die Probleme mit den gefundenen Kanälen zu diagnostizieren, die die Erstellung des Builds verursachen.

### **Windows Windows-Leistungsüberwachung**

In IBM WebSphere MQ 7.0 und früheren Versionen konnte die Leistung lokaler Warteschlangen auf Windows-Systemen mithilfe der Windows-Leistungsüberwachung überwacht werden. Ab IBM WebSphere MQ 7.1 ist diese Methode zur Leistungsüberwachung nicht mehr verfügbar.

Sie können Warteschlangen mit den im Abschnitt „Real-time Monitoring“ auf Seite 310 beschriebenen Methoden auf allen unterstützten Plattformen überwachen.

## **Cluster überwachen**

In einem Cluster können Sie Anwendungsnachrichten, Steuernachrichten und Protokolle überwachen. Es gibt spezielle Überwachungsaspekte, wenn die Clusterauslastung zwischen zwei oder mehr Instanzen einer Warteschlange verteilt wird.

### **Anwendungsnachrichten im Cluster überwachen**

In der Regel durchlaufen alle Clusternachrichten, die den Warteschlangenmanager verlassen, den SYSTEM.CLUSTER.TRANSMIT.QUEUE, unabhängig davon, welcher Clustersenderkanal für die Übertragung der Nachricht verwendet wird. Jeder Kanal enthält Nachrichten, die für diesen Kanal parallel zu allen anderen Clustersenderkanälen ausgerichtet sind. Ein wachsender Aufbau von Nachrichten in dieser Warteschlange kann auf ein Problem mit einem oder mehreren Kanälen hinweisen und muss untersucht werden:

- Die Tiefe der Warteschlange muss entsprechend für das Clusterdesign überwacht werden.
- Der folgende Befehl gibt alle Kanäle zurück, die mehr als eine Nachricht haben, die auf die Übertragungswarteschlange wartet:

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

Bei allen Clusternachrichten in einer einzigen Warteschlange ist es nicht immer leicht zu sehen, welcher Kanal Probleme hat, wenn er beginnt, sich zu füllen. Die Verwendung dieses Befehls ist eine einfache Möglichkeit, zu sehen, welcher Kanal verantwortlich ist.

Sie können einen Cluster-WS-Manager so konfigurieren, dass er mehrere Übertragungswarteschlangen hat. Wenn Sie das Warteschlangenmanagerattribut DEFCLXQ in CHANNEL ändern, wird jeder Clustersenderkanal einer anderen Clusterübertragungswarteschlange zugeordnet. Alternativ können Sie separate Übertragungswarteschlangen manuell konfigurieren. Führen Sie den folgenden Befehl aus, um alle Cluster-Übertragungswarteschlangen anzuzeigen, die Clustersenderkanälen zugeordnet sind:

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

Definieren Sie die Clusterübertragungswarteschlangen so, dass sie dem Muster des festen Stamms des Warteschlangennamens auf der linken Seite folgen. Anschließend können Sie die Länge aller Clusterübertragungswarteschlangen abfragen, die vom Befehl **DISPLAY CLUSMGR** zurückgegeben wurden, indem Sie einen generischen Warteschlangennamen verwenden:

```
DISPLAY QUEUE (qname *) CURDEPTH
```

## Überwachen von Steuernachrichten im Cluster

Die Warteschlange SYSTEM.CLUSTER.COMMAND.QUEUE wird für die Verarbeitung aller Clustersteuerungsnachrichten für einen Warteschlangenmanager verwendet, die entweder vom lokalen Warteschlangenmanager generiert oder von anderen Warteschlangenmanagern im Cluster an diesen Warteschlangenmanager gesendet wurden. Wenn ein WS-Manager seinen Clusterstatus korrekt aufrechterhält, neigt diese Warteschlange zu null. Es gibt Situationen, in denen die Tiefe der Nachrichten in dieser Warteschlange vorübergehend zunehmen kann:

- Wenn viele Nachrichten in der Warteschlange vorhanden sind, wird das Churn im Clusterstatus angezeigt.
- Wenn Sie wichtige Änderungen vornehmen, können Sie die Warteschlange zwischen diesen Änderungen einlagern. Wenn z. B. Repositories verschoben werden, können Sie die Warteschlange vor dem Verschieben des zweiten Repositories auf null setzen.

Während ein Nachrichtenrückstand in dieser Warteschlange vorhanden ist, werden Aktualisierungen für den Clusterstatus oder die clusterbezogenen Befehle nicht verarbeitet. Wenn Nachrichten über einen längeren Zeitraum nicht aus dieser Warteschlange entfernt werden, ist eine weitere Untersuchung erforderlich, die zunächst durch die Überprüfung der Fehlerprotokolle des Warteschlangenmanagers (oder der CHINIT-Protokolle unter z/OS) ausgeführt wird. Dadurch kann möglicherweise der Prozess ermittelt werden, durch den diese Situation verursacht wird.

Der SYSTEM.CLUSTER.REPOSITORY.QUEUE enthält die Informationen zum Cluster-Repository-Cache als Anzahl von Nachrichten. Es ist üblich, dass Nachrichten immer in dieser Warteschlange vorhanden sind, und mehr für größere Cluster. Daher ist die Tiefe der Nachrichten in dieser Warteschlange kein Problem für die Problemmeldung.

## Protokolle überwachen

Probleme, die im Cluster auftreten, zeigen möglicherweise keine externen Symptome an Anwendungen für viele Tage (und sogar Monate) an, nachdem das Problem aufgrund des Zwischenspeicherns von Informationen und der verteilten Clusterbildung ursprünglich aufgetreten ist. Das ursprüngliche Problem wird jedoch häufig in den IBM MQ-Fehlerprotokollen (und CHINIT-Protokollen unter z/OS) dokumentiert. Aus diesem Grund ist es wichtig, diese Protokolle für alle Nachrichten, die sich auf das Clustering beziehen, aktiv zu überwachen. Diese Nachrichten müssen gelesen und verstanden werden, wobei gegebenenfalls Maßnahmen ergriffen werden müssen.

Beispiel: Eine Unterbrechung der Kommunikation mit einem WS-Manager in einem Cluster kann dazu führen, dass bestimmte Clusterressourcen gelöscht werden, da Cluster die Clusterressourcen regelmäßig erneut überprüfen, indem sie die Informationen erneut veröffentlichen. Die Nachricht [AMQ9465](#) oder

CSQX465I auf z/OS-Systemen enthält eine Warnung über ein solches Ereignis, das möglicherweise auftritt. Diese Nachricht weist darauf hin, dass das Problem untersucht werden muss.

## Besondere Hinweise zum Lastausgleich

Wenn die Clusterauslastung zwischen zwei oder mehr Instanzen einer Warteschlange verteilt wird, müssen konsumierende Anwendungen die Nachrichten in jeder der Instanzen verarbeiten. Wenn eine oder mehrere der konsumierenden Anwendungen die Verarbeitung von Nachrichten beenden oder stoppt, ist es möglich, dass das Clustering weiterhin Nachrichten an diese Instanzen der Warteschlange sendet. In dieser Situation werden diese Nachrichten erst verarbeitet, wenn die Anwendungen wieder ordnungsgemäß funktionieren. Aus diesem Grund ist die Überwachung der Anwendungen ein wichtiger Bestandteil der Lösung, und es müssen Maßnahmen ergriffen werden, um die Nachrichten in dieser Situation erneut zu leiten. Ein Beispiel für einen Mechanismus zum Automatisieren einer solchen Überwachung finden Sie in diesem Beispiel: [The Cluster Queue Monitoring sample program \(AMQSCLM\)](#).

### Zugehörige Konzepte

„Verteilte Publish/Subscribe-Netze optimieren“ auf Seite 380

Verwenden Sie die Optimierungstipps in diesem Abschnitt, um die Leistung der verteilten Publish/Subscribe-Cluster und -Hierarchien von IBM MQ zu verbessern.

„Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen“ auf Seite 386

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

## Übertragungswarteschlangenwechsel überwachen

Es ist wichtig, dass Sie den Prozess der Clustersenderkanäle überwachen, die die Übertragungswarteschlangen wechseln, damit die Auswirkungen auf Ihr Unternehmen minimiert werden. Sie sollten diesen Prozess beispielsweise nicht versuchen, wenn die Auslastung hoch ist oder wenn Sie viele Kanäle gleichzeitig wechseln.

### Der Prozess zum Wechseln von Kanälen

Der Prozess, der zum Wechseln von Kanälen verwendet wird:

1. Der Kanal öffnet die neue Übertragungswarteschlange für die Eingabe und beginnt mit dem Abrufen von Nachrichten aus ihr (mithilfe des Abrufvorgangs nach Korrelations-ID).
2. Ein Hintergrundprozess wird vom Warteschlangenmanager eingeleitet, um alle Nachrichten, die für den Kanal eingereicht wurden, aus seiner alten Übertragungswarteschlange in seine neue Übertragungswarteschlange zu verschieben. Beim Verschieben von Nachrichten werden alle neuen Nachrichten für den Kanal in die alte Übertragungswarteschlange eingereicht, um die Reihenfolgeplanung beizubehalten. Dieser Prozess kann einige Zeit dauern, wenn eine große Anzahl von Nachrichten für den Kanal in der alten Übertragungswarteschlange vorhanden ist oder neue Nachrichten schnell ankommen.
3. Wenn keine festgeschriebenen oder nicht festgeschriebenen Nachrichten für den Kanal in der alten Übertragungswarteschlange verbleiben, wird der Wechsel abgeschlossen. Neue Nachrichten werden jetzt direkt in die neue Übertragungswarteschlange eingereicht.

Um den Fall zu vermeiden, dass mehrere Kanäle gleichzeitig wechseln, bietet IBM MQ die Möglichkeit, die Übertragungswarteschlange eines oder mehrerer Kanäle zu wechseln, die nicht aktiv sind. Unter:

- IBM MQ for Multiplatforms heißt der Befehl **runswch1**
- IBM MQ for z/OS Das Dienstprogramm CSQUTIL kann verwendet werden, um einen Befehl SWITCH CHANNEL zu verarbeiten.

## Status von Switchoperationen überwachen

Um den Status von Switchoperationen zu verstehen, können Administratoren die folgenden Aktionen ausführen:

- Überwachen Sie das Fehlerprotokoll des Warteschlangenmanagers (AMQERR01.LOG), in dem Nachrichten ausgegeben werden, um die folgenden Phasen während der Operation anzugeben:
  - Die Switchoperation wurde gestartet.
  - Das Verschieben von Nachrichten wurde gestartet.
  - Regelmäßige Aktualisierungen zur Anzahl der noch zu verschiebenden Nachrichten (wenn die Switchoperation nicht schnell abgeschlossen wird)
  - Das Verschieben von Nachrichten ist abgeschlossen.
  - Die Switchoperation ist abgeschlossen

Unter z/OS werden diese Nachrichten im Jobprotokoll des Warteschlangenmanagers ausgegeben, nicht im Jobprotokoll des Kanalinitiators, obwohl eine einzelne Nachricht von einem Kanal in das Jobprotokoll des Kanalinitiators ausgegeben wird, wenn beim Start ein Wechsel eingeleitet wird.

- Mit dem Befehl `DISPLAY CLUSQMGR` können Sie die Übertragungswarteschlange abfragen, die momentan von jedem Clustersenderkanal verwendet wird.
- Führen Sie den Befehl `runswchl` (oder `CSQUTIL` unter z/OS) im Abfragemodus aus, um den Wechselstatus eines oder mehrerer Kanäle zu ermitteln. Die Ausgabe dieses Befehls gibt Folgendes für jeden Kanal an:
  - Gibt an, ob für den Kanal eine Switchoperation ansteht
  - Von und zu welcher Übertragungswarteschlange der Kanal wechselt
  - Wie viele Nachrichten in der alten Übertragungswarteschlange verbleiben

Jeder Befehl ist wirklich nützlich, da Sie in einem Aufruf den Status jedes Kanals, die Auswirkung einer Konfigurationsänderung und die Beendigung aller Switchoperationen ermitteln können.

## Potenzielle Probleme, die auftreten können

Im Abschnitt Mögliche Probleme beim Wechseln von Übertragungswarteschlangen finden Sie eine Liste einiger Probleme, die beim Wechseln der Übertragungswarteschlange auftreten können, ihre Ursachen und die wahrscheinlichsten Lösungen.

### Zugehörige Konzepte

„Verteilte Publish/Subscribe-Netze optimieren“ auf Seite 380

Verwenden Sie die Optimierungstipps in diesem Abschnitt, um die Leistung der verteilten Publish/Subscribe-Cluster und -Hierarchien von IBM MQ zu verbessern.

„Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen“ auf Seite 386

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

V 9.1.3

Multi

## Lastausgleich von Anwendungen überwachen

Beschreibung des Befehls `DISPLAY APSTATUS` zur Überwachung des Lastausgleichsstatus von Anwendungen in einem Uniform-Cluster sowie zur Ermittlung des Grunds, weshalb eine Anwendung wider Erwarten nicht im Lastausgleich eingeschlossen ist.

## Aktuellen Status von Anwendungen über mehrere Warteschlangenmanager hinweg in einem Cluster überwachen

Sie können sich von jedem Warteschlangenmanager in einem Uniform-Cluster aus einen Überblick über den aktuellen Status der Anwendungen über alle Warteschlangenmanager eines Clusters hinweg verschaffen, indem Sie den Befehl `DIS APSTATUS` ausführen.

Nachdem ein Warteschlangenmanager kurz zuvor gestartet wurde, ist beispielsweise eine Ausgabe mit ungefähr folgendem Text möglich:

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(UNKNOWN)
```

**V 9.1.4** Ab IBM MQ 9.1.4 enthält die Ausgabe auch das Feld **TYPE**:

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(UNKNOWN)
TYPE (APPL)
```

Dies zeigt an, dass der Uniform-Cluster eine Anwendung namens MYAPP enthält und es derzeit acht Instanzen gibt, von denen alle acht als im Uniform-Cluster verschiebbar gelten. Der Laustausgleiswert UNKNOWN ist ein temporärer Wert, der angibt, dass der Warteschlangenmanager noch nicht versucht hat, die Anwendung im Bedarfsfall neu zu verteilen.

Nach einer kurzen Zeit ist folgende Ausgabe wahrscheinlicher:

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(NO)
```

**V 9.1.4**

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(NO)
TYPE (APPL)
```

Diese Ausgabe zeigt, dass die Anwendung acht Instanzen hat, zwischen denen im Uniform-Cluster jedoch kein Lastausgleich erfolgt. An diesem Punkt lohnt es sich, einen Blick auf die Verteilung der Anwendungen im Cluster zu werfen.

Führen Sie hierzu erneut den Befehl **DIS APSTATUS** aus. Beachten Sie, dass Sie diesen Befehl für jeden beliebigen Warteschlangenmanager im Uniform-Cluster ausführen können:

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(6)                      MOVCCOUNT(6)
BALSTATE(HIGH)                 LMSGDATE(2019-05-24)
LMSGTIME(13:11:10)             QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                      MOVCCOUNT(1)
BALSTATE(LOW)                  LMSGDATE(2019-05-24)
LMSGTIME(13:11:03)            QMNAME(UNID002)
```

```

QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                       MOVCCOUNT(1)
BALSTATE(LOW)                   LMSGDATE(2019-05-24)
LMSGTIME(13:11:07)             QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)

```

#### V 9.1.4

```

1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(6)                       MOVCCOUNT(6)
BALSTATE(HIGH)                 LMSGDATE(2019-05-24)
LMSGTIME(13:11:10)           QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                       MOVCCOUNT(1)
BALSTATE(LOW)                   LMSGDATE(2019-05-24)
LMSGTIME(13:11:03)           QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                       MOVCCOUNT(1)
BALSTATE(LOW)                   LMSGDATE(2019-05-24)
LMSGTIME(13:11:07)           QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)

```

Daraus wird zu diesem Zeitpunkt ersichtlich, dass der Warteschlangenmanager UNID001 über sechs Instanzen verfügt, die Warteschlangenmanager UNID0002 und UNID0003 jedoch nur über jeweils eine Instanz. Die BALSTATE-Ausgabe zeigt den Laustausgleichsstatus an, der zuletzt vom Warteschlangenmanager gemeldet wurde. Beachten Sie jedoch, dass die Instanzanzahl möglicherweise aktueller als das Feld BALSTATE ist.

Diese Ausgabe gibt auch einen hilfreichen Hinweis darauf, dass der Uniform-Cluster Informationen zu dieser Anwendung verteilt. Die Anwendung ist allen Warteschlangenmanagerinstanzen in im Uniform-Cluster bekannt, und das Datum und die Uhrzeit der letzten Nachricht sind sehr aktuell.

Außerdem gibt das Feld AKTIV an, dass alle Warteschlangenmanager im Cluster als kommunizierend miteinander betrachtet werden; wenn ACTIVE auf NO auf einem beliebigen Warteschlangenmanager gesetzt ist, zeigt dies an, dass die reguläre Kommunikation mit dem Warteschlangenmanager unterbrochen wurde.

Wenn der Lastausgleich sich selbst überlassen wird, würde die Ausgabe letztendlich etwa folgendermaßen aussehen:

```

1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(3)                       MOVCCOUNT(3)
BALSTATE(OK)                   LMSGDATE(2019-05-24)
LMSGTIME(13:14:22)           QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(3)                       MOVCCOUNT(3)
BALSTATE(OK)                   LMSGDATE(2019-05-24)
LMSGTIME(13:13:53)           QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(2)                       MOVCCOUNT(2)
BALSTATE(OK)                   LMSGDATE(2019-05-24)
LMSGTIME(13:13:47)           QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)

```

#### V 9.1.4

```

1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.

```

```

APPLNAME(MYAPP)           ACTIVE(YES)
COUNT(3)                 MOVCCOUNT(3)
BALSTATE(OK)              LMSGDATE(2019-05-24)
LMSGTIME(13:14:22)       QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)           ACTIVE(YES)
COUNT(3)                 MOVCCOUNT(3)
BALSTATE(OK)              LMSGDATE(2019-05-24)
LMSGTIME(13:13:53)       QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)           ACTIVE(YES)
COUNT(2)                 MOVCCOUNT(2)
BALSTATE(OK)              LMSGDATE(2019-05-24)
LMSGTIME(13:13:47)       QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)

```

Der BALSTATE der Warteschlangenmanager ist jetzt OK, was darauf hinweist, dass ein stabiler Zustand erreicht wurde.

## Einzelne Anwendungsinstanzen überwachen

Sie können die einzelnen Anwendungsinstanzen betrachten, tun dies allerdings für jeden Warteschlangenmanager. Sehen Sie sich die Ausgabe unter UNID001 an:

```

1 : DIS APSTATUS(*) type(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02EA0A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                  IMMREASN(NONE)
IMMCOUNT(0)               IMMDATE( )
IMMTIME( )                MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E50A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                  IMMREASN(NONE)
IMMCOUNT(0)               IMMDATE( )
IMMTIME( )                MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E60A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                  IMMREASN(NONE)
IMMCOUNT(0)               IMMDATE( )
IMMTIME( )                MOVABLE(YES)
One MQSC command read.

```

### V 9.14

```

1 : DIS APSTATUS(*) type(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02EA0A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                  IMMREASN(NONE)
IMMCOUNT(0)               IMMDATE( )
IMMTIME( )                MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E50A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                  IMMREASN(NONE)
IMMCOUNT(0)               IMMDATE( )
IMMTIME( )                MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E60A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1)                  IMMREASN(NONE)
IMMCOUNT(0)               IMMDATE( )
IMMTIME( )                MOVABLE(YES)
TYPE (LOCAL)
One MQSC command read.

```

Das Vorhandensein von MOVABLE(YES) zeigt an, dass diese Instanz bei Bedarf in einen anderen Warteschlangenmanager im Cluster verschoben werden kann. Im folgenden Beispiel ist die Anwendungsinstanz nicht verschiebbar, da sie sich nicht als Client verbindet:

```

3 : DISPLAY APSTATUS('ServerApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ServerApp)
CONNTAG(MQCT02DFE75C01800B20UNID001_2019-05-24_13.09.35ServerApp)
CONNS(1) IMMREASN(NOTCLIENT)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)

```

#### V 9.1.4

```

3 : DISPLAY APSTATUS('ServerApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ServerApp)
CONNTAG(MQCT02DFE75C01800B20UNID001_2019-05-24_13.09.35ServerApp)
CONNS(1) IMMREASN(NOTCLIENT)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)

```

Im Feld **CONNTAG** können Sie die einzelnen Warteschlangenmanagerverbindungen dieser betreffenden Instanz sehen. Dies kann hilfreich sein, wenn für die Anwendungsinstanz IMMREASN(NOTRECONN) angegeben ist. Im folgenden Beispiel ist die Clientanwendung nicht verschiebbar, und die Untersuchung der Verbindungsoptionen zeigt, dass MQCNO\_RECONNECT im Feld **CONNOPTS** nicht vorhanden ist.

```

1 : DISPLAY APSTATUS('ClientApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ClientApp)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
CONNS(1) IMMREASN(NOTRECONN)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)

2 : DISPLAY CONN(*) TYPE(CONN) WHERE(CONNTAG eq
'MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp') ALL
AMQ8276I: Display Connection details.
CONN(02DFE75C01CB0B20)
EXTCONN(414D5143554E49443030312020202020)
TYPE(CONN)
PID(14656) TID(20)
APPLDESC(IBM MQ Channel) APPLTAG(ClientApp)
APPLTYPE(USER) ASTATE(NONE)
CHANNEL(SYSTEM.DEF.SVRCONN) CLIENTID( )
CONNNAME(127.0.0.1)
CONNOPTS(MQCNO_HANDLE_SHARE_BLOCK,MQCNO_SHARED_BINDING)
USERID(MyUserid) UOWLOG( )
UOWSTDA( ) UOWSTTI( )
UOWLOGDA( ) UOWLOGTI( )
URTYPE(QMGR)
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
QMURID(0.0) UOWSTATE(NONE)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)

```

#### V 9.1.4

```

1 : DISPLAY APSTATUS('ClientApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ClientApp)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
CONNS(1) IMMREASN(NOTRECONN)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)

2 : DISPLAY CONN(*) TYPE(CONN) WHERE(CONNTAG eq
'MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp') ALL
AMQ8276I: Display Connection details.
CONN(02DFE75C01CB0B20)
EXTCONN(414D5143554E49443030312020202020)
TYPE(CONN)
PID(14656) TID(20)

```

APPLDESC(IBM MQ Channel)	APPLTAG(ClientApp)
APPLTYPE(USER)	ASTATE(NONE)
CHANNEL(SYSTEM.DEF.SVRCONN)	CLIENTID( )
CONNNAME(127.0.0.1)	
CONNOPTS(MQCN0_HANDLE_SHARE_BLOCK,MQCN0_SHARED_BINDING)	
USERID(MyUserid)	UOWLOG( )
UOWSTDA( )	UOWSTTI( )
UOWLOGDA( )	UOWLOGTI( )
URTYPE(QMGR)	
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])	
QMURID(0,0)	UOWSTATE(NONE)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)	
TYPE (CONN)	

## Zugehörige Konzepte

[Automatischer Ausgleich von Anwendungen](#)

## z/OS Überwachen der Leistung und Ressourcennutzung

Dieser Abschnitt enthält Informationen zu den Funktionen, die für die Überwachung der Leistung und die Ressourcennutzung Ihrer IBM MQ for z/OS-Subsysteme zur Verfügung stehen.

### Zugehörige Tasks

[z/OS konfigurieren](#)

[IBM MQ for z/OS verwalten](#)

## z/OS Einführung in die Überwachung

Verwenden Sie diesen Abschnitt als Übersicht über die Überwachungsfunktionen, die für IBM MQ for z/OS verfügbar sind. Dabei handelt es sich beispielsweise um den Abruf von Momentaufnahmen mithilfe des IBM MQ-Trace, der Onlineüberwachung und von Ereignissen.

In diesem Abschnitt wird beschrieben, wie die Leistung und Ressourcennutzung von IBM MQ überwacht wird.

- Sie beschreibt einige der Informationen, die Sie abrufen können, und beschreibt kurz einen allgemeinen Ansatz zur Untersuchung von Leistungsproblemen.
- Sie beschreibt einige der Informationen, die Sie abrufen können, und beschreibt kurz einen allgemeinen Ansatz zur Untersuchung von Leistungsproblemen. Weitere Informationen finden Sie unter [„Leistungsprobleme untersuchen“](#) auf Seite 338.
- 
- Es wird beschrieben, wie Sie Statistikdaten zur Leistung von IBM MQ mithilfe von SMF-Datensätzen erfassen können.
- Es wird beschrieben, wie Sie Abrechnungsdaten erfassen können, mit denen Sie die Nutzung Ihrer IBM MQ-Systeme durch Ihre Kunden berechnen können.
- Es wird beschrieben, wie Sie mit IBM MQ-Ereignissen (Benachrichtigungen) Ihre Systeme überwachen können.

Im Folgenden sind einige Tools aufgeführt, mit denen Sie IBM MQ überwachen können. Sie werden in den folgenden Abschnitten beschrieben:

- Von IBM MQ bereitgestellte Tools:
  - [DISPLAY-Befehle verwenden](#)
  - [„CICS-Adapterstatistik verwenden“](#) auf Seite 333
  - [„IBM MQ-Ereignisse verwenden“](#) auf Seite 335
- z/OS-Servicehilfen:
  - [„Systemverwaltungsfunktion verwenden“](#) auf Seite 335
- Andere IBM Lizenzprogramme:
  - [Ressourcenmessungsfunktion verwenden](#)

- Tivoli Decision Support für z/OS verwenden
- Funktion zur Überwachung von CICS verwenden

Informationen zum Interpretieren der Daten, die im Trace zur Leistungsstatistik erfasst werden, finden Sie im Abschnitt „IBM MQ-Leistungsstatistik interpretieren“ auf Seite 339.

Informationen zum Interpretieren der Daten, die im Abrechnungstrace erfasst werden, finden Sie im Abschnitt „IBM MQ-Abrechnungsdaten interpretieren“ auf Seite 366.

### **Momentaufnahmen von IBM MQ mit dem Befehl DISPLAY abrufen**

IBM MQ stellt die MQSC-Funktion bereit, die eine Momentaufnahme der Leistung und Ressourcennutzung mithilfe der DISPLAY-Befehle ermöglicht.

Sie können eine Vorstellung des aktuellen Status von IBM MQ erhalten, indem Sie die DISPLAY-Befehle und für den CICS-Adapter die CICS-Adapteranzeige verwenden.

#### **DISPLAY-Befehle verwenden**

Mit den Befehlen IBM MQ MQSC DISPLAY oder PCF Inquire können Sie Informationen zum aktuellen Status von IBM MQ abrufen. Sie stellen Informationen zum Status des Befehlsservers, Prozessdefinitionen, Warteschlangen, des Warteschlangenmanagers und der zugehörigen Komponenten bereit. Diese Befehle sind:

<b>MQSC-Befehl</b>	<b>PCF-Befehl</b>
ANZEIGEN ARCHIVE	Archiv erfragen
DISPLAY AUTHINFO	Inquire Authentication Information Object
DISPLAY CFSTATUS	CF-Strukturstatus abgefragt
ANZEIGEN CFSTRUCT	CF-Struktur erfragen
ANZEIGEN CHANNEL	Kanalinquire
ANZEIGEN CHINIT	Kanalinitiator inquire
ANZEIGEN CHSTATUS	Kanalstatus abgefragt
ANZEIGEN CMDSERV	
DISPLAY CLUSQMGR	Clusterwarteschlangenmanager anfragen
DISPLAY CONN	Verbindung anfragen
ANZEIGEN G	Inquire-Gruppe
ANZEIGEN LOG	Protokoll anfragen
ANZEIGEN PROZ	Prozess inquire
ANZEIGEN QMGR	Warteschlangenmanager abfragen
ANZEIGEN QSTATUS	Warteschlangenstatus abgefragt
ANZEIGEN QUEUE	Warteschlange einfragen
DISPLAY SECURITY	Sicherheit inquire
ANZEIGEN STGKLASSE	Speicherklasse inquire
ANZEIGEN SYSTEM	System abgefragt
ANZEIGEN TRACE	
ANZEIGEN SYNTA	Belegung inquire

Die detaillierte Syntax der einzelnen Befehle finden Sie in den [MQSC-Befehlen](#) oder [PCF-Befehlen](#). Alle Funktionen dieser Befehle (außer DISPLAY CMDSERV und DISPLAY TRACE) sind auch über die Operations- und Steuerkonsolen verfügbar.

Diese Befehle stellen eine Momentaufnahme des Systems nur zur Verfügung, wenn der Befehl verarbeitet wurde. Wenn Sie die Trends im System untersuchen möchten, müssen Sie einen IBM MQ-Trace starten und die Ergebnisse über einen Zeitraum analysieren.

### **CICS-Adapterstatistik verwenden**

Wenn Sie ein berechtigter CICS-Benutzer sind, können Sie die Anzeige für die CICS-Adaptersteuerung verwenden, um die CICS-Adapterstatistik dynamisch anzuzeigen.

Diese Statistikdaten stellen eine Momentaufnahme der Informationen dar, die sich auf die Nutzung des CICS-Threads und auf Situationen beziehen, in denen alle Threads ausgelastet sind. Die Anzeige "Verbindungsanzeige" kann durch Drücken der Eingabetaste aktualisiert werden. Weitere Informationen finden Sie im Abschnitt "The CICS-IBM MQ Adapter" in der Produktdokumentation zu CICS Transaction Server für z/OS 4.1 unter [CICS Transaction Server für z/OS 4.1, dem CICS-IBM MQ -Adapter](#).

### **IBM MQ-Trace verwenden**

Sie können mithilfe der IBM MQ-Tracefunktion eine Leistungsstatistik und Abrechnungsdaten für IBM MQ erfassen. In diesem Abschnitt wird beschrieben, wie der IBM MQ-Trace gesteuert wird.

Die von IBM MQ generierten Daten werden gesendet an:

- SMF (System Management Facility), insbesondere als SMF-Datensatztyp 115, Subtypen 1 und 2 für den Leistungsstatistiktrace
- Der SMF, insbesondere der SMF-Satztyp 116, die Subtypen Null, 1 und 2 für den Accounting-Trace.

Falls gewünscht, können die vom IBM MQ-Abrechnungstrace generierten Daten auch an die allgemeine Tracefunktion (GTF) gesendet werden.

## **IBM MQ-Trace starten**

Sie können die IBM MQ-Tracefunktion jederzeit starten, indem Sie den Befehl IBM MQ [START TRACE](#) absetzen.

Abrechnungsdaten können verloren gehen, wenn der Abrechnungstrace gestartet oder gestoppt wird, während Anwendungen aktiv sind. Damit die Abrechnungsdaten erfolgreich erfasst werden können, müssen die folgenden Bedingungen erfüllt sein:

- Der Accounting-Trace muss aktiv sein, wenn eine Anwendung gestartet wird, und sie muss aktiv sein, wenn die Anwendung abgeschlossen ist.
- Wenn der Accounting-Trace gestoppt ist, wird jede abrechnungsbezogene Datensammlung gestoppt, die aktiv war.

Sie können auch die Erfassung einiger Trace-Informationen automatisch starten, wenn Sie in den Parametern SMFSTAT (SMF STATISTICS) und SMFACCT (SMF ACCOUNTING) des Makros CSQ6SYSP YES angeben.  Diese Parameter werden im Abschnitt [CSQ6SYSP verwenden](#) beschrieben.

Vor dem Start eines IBM MQ-Trace lesen Sie den Abschnitt „[Systemverwaltungsfunktion verwenden](#)“ auf Seite 335.

## **IBM MQ-Trace steuern**

Um die IBM MQ-Tracedatenerfassung beim Start zu steuern, müssen Sie Werte für die Parameter im Makro CSQ6SYSP angeben, wenn Sie IBM MQ anpassen.  Einzelheiten finden Sie im Abschnitt [CSQ6SYSP verwenden](#).

Sie können die Tracefunktion für IBM MQ steuern, wenn der Warteschlangenmanager mit diesen Befehlen ausgeführt wird:

- START TRACE
- ALTER TRACE
- STOP-TRACE

Sie können die Zieladresse auswählen, an die die Trace-Daten gesendet werden sollen. Folgende Ziele sind möglich:

**SMF**

Systemverwaltungsfunktion

**GTF**

Generalized Trace Facility (nur Accounting-Trace)

**SRV**

Funktionsfähigkeitsroutine, die vom IBM-Kundendienst zur Diagnose verwendet wird

Für die tägliche Überwachung werden Informationen an SMF gesendet (das Standardziel). SMF-Dateien enthalten in der Regel Informationen von anderen Systemen. Diese Informationen sind erst dann für die Berichterstellung verfügbar, wenn ein Speicherauszug der SMF-Datei erstellt wird.

Sie können auch Accounting-Trace-Informationen an den GTF senden. Diese Informationen enthalten eine Ereignis-ID von 5EE. **z/OS** Mit dem MQI-Aufruf, dem Benutzerparameter und der allgemeine Tracefunktion von z/OS (GTF) wird beschrieben, wie IBM MQ-Traceinformationen verarbeitet werden, die an die GTF gesendet werden.

Informationen zu IBM MQ-Befehlen finden Sie im Abschnitt [MQSC-Befehle](#).

**Auswirkung des Trace auf IBM MQ-Leistung**

Die Verwendung der IBM MQ-Tracefunktion kann erhebliche Auswirkungen auf IBM MQ und die Transaktionsleistung haben. Wenn Sie beispielsweise einen globalen Trace für die Klasse 1 oder für alle Klassen starten, erhöht sich die Wahrscheinlichkeit, dass die Prozessorbelegung und die Antwortzeiten von Transaktionen um etwa 50% erhöht werden. Wenn Sie jedoch einen globalen Trace nur für die Klassen 2 bis 4 starten, erhöhen sich die Prozessorauslastung und die Transaktionsantwortzeiten um weniger als 1% zusätzlicher Prozessorauslastung im Vergleich zur Auslastung bei IBM MQ-Aufrufen. Dasselbe gilt für einen Statistik-oder Accounting-Trace.

**z/OS IBM MQ-Onlineüberwachung verwenden**

Sie können Überwachungsdaten für Warteschlangen und Kanäle (einschließlich automatisch definierter Cluster-Server-Kanäle) erfassen, indem Sie die Attribute MONQ, MONCHL und MONACLS definieren.

Tabelle 33 auf Seite 334 fasst die Befehle zur Einstellung dieser Attribute auf verschiedenen Ebenen und zur Anzeige der Überwachungsdaten zusammen.

<i>Tabelle 33. Attribute festlegen und anzeigen, um die Onlineüberwachung zu steuern</i>			
<b>Attribut</b>	<b>Anwendbar auf dieser Ebene</b>	<b>Mit Befehl festlegen</b>	<b>Überwachungsdaten mit dem Befehl anzeigen</b>
MONQ	Warteschlange	QLOCAL DEFINIER QMODEL DEFINIER ALTER QLOCAL ALTER QMODEL	ANZEIGEN QSTATUS
	Warteschlangenmanager	ALTER QMGR	

Tabelle 33. Attribute festlegen und anzeigen, um die Onlineüberwachung zu steuern (Forts.)

Attribut	Anwendbar auf dieser Ebene	Mit Befehl festlegen	Überwachungsdaten mit dem Befehl anzeigen
MONCHL	Kanal	CHANNEL DEFINE CHANNEL ALTER CHANNEL	ANZEIGEN CHSTATUS
	Warteschlangenmanager	ALTER QMGR	
MONACLS	Warteschlangenmanager	ALTER QMGR	

Ausführliche Informationen zu diesen Befehlen finden Sie im Abschnitt [MQSC-Befehle](#). Weitere Informationen zur Onlineüberwachung finden Sie im Abschnitt [„IBM MQ-Netz überwachen“](#) auf Seite 5.

### **IBM MQ-Ereignisse verwenden**

IBM MQ-Instrumentierungsereignisse stellen Informationen zu Fehlern, Warnungen und anderen wichtigen Vorkommnissen in einem Warteschlangenmanager bereit. Sie können die Operation all Ihrer Warteschlangenmanager überwachen, indem Sie diese Ereignisse in Ihre eigene Systemmanagementanwendung integrieren.

IBM MQ-Instrumentierungsereignisse schlagen in den folgenden Kategorien fehl:

#### **WS-Manager-Ereignisse**

Diese Ereignisse beziehen sich auf die Definitionen von Ressourcen in Warteschlangenmanagern. Eine Anwendung versucht z. B., eine Nachricht in eine Warteschlange zu stellen, die nicht vorhanden ist.

#### **Durchsatzereignisse**

Bei diesen Ereignissen handelt es sich um Benachrichtigungen, bei denen eine Schwellenwertbedingung von einer Ressource erreicht wurde. Beispielsweise wurde eine Warteschlangenlänge erreicht, oder die Warteschlange wurde nicht innerhalb eines vordefinierten Zeitlimits verarbeitet.

#### **Kanalereignisse**

Diese Ereignisse werden von Kanälen aufgrund von Bedingungen gemeldet, die während ihrer Operation erkannt wurden. Beispielsweise wird eine Kanalinstanz gestoppt.

#### **Konfigurationsereignisse**

Bei diesen Ereignissen handelt es sich um Benachrichtigungen, die ein Objekt erstellt, geändert oder gelöscht wurde.

Wenn ein Ereignis eintritt, stellt der Warteschlangenmanager eine *Ereignisnachricht* in die entsprechende *Ereigniswarteschlange* (falls definiert). Die Ereignisnachricht enthält Informationen zu dem Ereignis, die von einer geeigneten IBM MQ-Anwendung abgerufen werden können.

IBM MQ-Ereignisse können mit den IBM MQ-Befehlen oder über die Operations- und Steueranzeigen aktiviert werden.

Der Abschnitt [„Ereignistypen“](#) auf Seite 8 enthält Informationen zu den IBM MQ-Ereignissen, die Nachrichten generieren, sowie zum Format dieser Nachrichten. Weitere Informationen zum Aktivieren der Ereignisse finden Sie unter [Ereignisnachrichtenreferenz](#).

### **Systemverwaltungsfunktion verwenden**

Sie können SMF verwenden, um Statistikdaten und Abrechnungsdaten zu erfassen. Für die Verwendung von SMF müssen bestimmte Parameter in z/OS und IBM MQ festgelegt werden.

SMF (System Management Facility, Systemverwaltungsfunktion) ist eine z/OS-Servicehilfe, mit der Informationen von verschiedenen z/OS-Subsystemen erfasst werden. Diese Informationen werden in regelmäßigen Abständen erstellt und berichtet, z. B. stündlich. Sie können SMF mit der IBM MQ-Tracefunktion verwenden, um Daten aus IBM MQ zu erfassen. Auf diese Weise können Sie *Trends* überwachen, bei-

spielsweise in der Systemauslastung und -leistung. Sie können außerdem Abrechnungsdaten zu jeder Benutzer-ID erfassen, die IBM MQ verwendet.

Wenn Sie die Leistungsstatistik (Datensatztyp 115) in SMF aufzeichnen möchten, geben Sie Folgendes im Member SMFPRMxx von SYS1.PARMLIB oder mit dem z/OS-Bedienerbefehl SETSMF an.

```
SYS(TYPE(115))
```

Wenn Sie Abrechnungsdaten (Datensatztyp 116) in SMF aufzeichnen möchten, geben Sie Folgendes im Member SMFPRMxx von SYS1.PARMLIB oder mit dem z/OS-Bedienerbefehl SETSMF an.

```
SYS(TYPE(116))
```

Für die Verwendung des z/OS-Befehls SETSMF muss im SMFPRM-Member xx entweder PROMPT(ALL) oder PROMPT(LIST) angegeben sein. Weitere Informationen finden Sie im Handbuch [z/OS MVS Initialization and Tuning Reference](#) und im Handbuch [z/OS MVS System Commands](#).

Sie können mit der Erfassung einiger Trace-Informationen automatisch beginnen, wenn Sie YES in den Parametern SMFSTAT (SMF STATISTICS) und SMFACCT (SMF ACCOUNTING) des Makros CSQ6SYSP angeben. Dies wird im Abschnitt [CSQ6SYSP](#) beschrieben.

Wenn Sie YES in den Parametern SMFSTAT und SMFACCT angeben, können Sie Trace-Informationen als Warteschlangenmanager erfassen.

Sie können die Datenerfassung auch starten, wenn der WS-Manager mit dem Befehl **START TRACE** unter Angabe von START TRACE (A) oder START TRACE (S) ausgeführt wird.

Sie können die Aufzeichnung von Abrechnungsdaten auf Warteschlangen- oder Warteschlangenmanager-ebene mit dem Parameter ACCTQ der Befehle **DEFINE QLOCAL**, **DEFINE QMODEL**, **ALTER QLOCAL**, **ALTER QMODEL** oder **ALTER QMGR** aktivieren oder inaktivieren. Ausführliche Informationen zu diesen Befehlen finden Sie in den [MQSC-Befehlen](#).

Die Erfassung von Kanalabrechnungsdaten können Sie mit dem Parameter **STATCHL** der Befehle **DEFINE CHANNEL**, **ALTER CHANNEL** oder **ALTER QMGR** auf Kanal- oder Warteschlangenmanagerebene steuern.

Sie können das Intervall angeben, in dem IBM MQ die Statistik- und Abrechnungsdaten auf eine der beiden folgenden Arten erfasst:

- Sie können Statistik- und Abrechnungsdaten in demselben Intervall erfassen, indem Sie in Ihren Systemparametern einen Wert für "STATIME" angeben (siehe [CSQ6SYSP](#) verwenden).
- Sie können Statistikdaten und Abrechnungsdaten [unter Verwendung des Vorkommens von Abrechnungseignissen](#) erfassen, indem Sie für STATIME null angeben.

SMF muss bereits aktiv sein, damit Sie Daten an die Funktion senden können. Weitere Informationen zu SMF enthält das Handbuch [z/OS MVS System Management Facilities \(SMF\)](#).

Für die Statistik- und Abrechnungsdaten, die zurückgesetzt werden sollen, muss mindestens ein MQI-Aufruf während des Abrechnungsintervalls abgesetzt werden.

## Zuordnen zusätzlicher SMF-Puffer

Wenn Sie einen Trace starten, müssen Sie sicherstellen, dass Sie genügend SMF-Puffer zuordnen. Geben Sie die SMF-Pufferung im Parameter VSAM BUFSP der Anweisung DEFINE CLUSTER des Zugriffsmethodeservices an. Geben Sie in der Anweisung **DEFINE CLUSTER** für jede SMF-VSAM-Dateigruppe CISZ(4096) und BUFSP(81920) an.

Wenn ein SMF-Puffermangel auftritt, weist SMF alle Tracesätze zurück, die an sie gesendet wurden. IBM MQ sendet in diesem Fall eine CSQW133I-Nachricht an die z/OS-Konsole. IBM MQ behandelt den Fehler als temporär und bleibt aktiv, auch wenn die SMF-Daten möglicherweise verloren gehen. Nachdem der Engpass behoben wurde und die Traceaufzeichnung wieder aufgenommen worden ist, sendet IBM MQ eine Nachricht CSQW123I an die z/OS-Konsole.

## Berichtsdaten in SMF

Sie können das SMF-Programm IFASMFDP (bzw. bei Verwendung von Protokollströmen das SMF-Programm IFASMFDP) verwenden, um einen Speicherauszug von SMF-Datensätzen in einer sequenziellen Datei zu erstellen, damit sie verarbeitet werden können.

Es gibt mehrere Möglichkeiten, um über diese Daten zu berichten, z. B.:

- Schreiben Sie ein Anwendungsprogramm, um Informationen aus der SMF-Dateigruppe zu lesen und zu berichten. Anschließend können Sie den Bericht entsprechend Ihren Anforderungen anpassen.
- Verwenden Sie Performance Reporter, um die Datensätze zu verarbeiten. Weitere Informationen finden Sie unter „Andere Produkte mit IBM MQ verwenden“ auf Seite 337.

## Keine Intervallklasse (03) SMF-Abrechnungsdatensätze, die während Prozessen mit langer Laufzeit erzeugt werden

Sie erfassen CLASS (3) SMF116 -Abrechnungsdatensätze für IBM MQ, erhalten aber keine Datensätze, während ein Prozess mit langer Laufzeit ausgeführt wird.

Die Abrechnungsdatensätze des Typs CLASS (3) SMF116 werden normalerweise nur erstellt, wenn ein Prozess beendet wird. For long running processes, for example CICS, this might not produce a sufficient number of records, as the process can run for a month or longer. Sie können jedoch SMF116 -Datensätze in festgelegten Zeitintervallen zusammenstellen, während ein Prozess ausgeführt wird.

Für die Erfassung von Abrechnungsdatensätzen des Typs CLASS (3) SMF116 müssen Sie Folgendes festlegen:

### SMFACCT

= JA

### SMFSTAT

=YES oder NO, wobei

#### YES

Bewirkt, dass Datensätze erzeugt werden, wenn ein Sammlungsbroadcast empfangen wird.

#### Nein

Bewirkt, dass ein Datensatz des Typs CLASS (3) SMF116 nur erzeugt wird, wenn ein Prozess beendet wird.

und setzen Sie den folgenden Befehl ab:

```
START TRACE(ACCTG) DEST(SMF) CLASS(03)
```

Wenn Sie SMFSTAT = YES festgelegt haben und ein Erfassungs-Broadcast auftritt, wird ein Intervallklassendatensatz (3) SMF116 für jeden Prozess erzeugt, der momentan ausgeführt wird und der zum Zeitpunkt des vorherigen Erfassungs-Broadcast ebenfalls ausgeführt wurde.

Sie können festlegen, dass der Erfassungs-Broadcast in einem regelmäßigen Zeitintervall stattfinden soll, indem Sie STATIME in CSQ6SYSP wie folgt festlegen:

- Wenn STATIME auf einen Wert größer als 0 gesetzt wurde, ist dies Ihr Broadcast-Intervall in Minuten.
- Wenn STATIME = 0 ist, wird der SMF-Broadcast Ihres Systems verwendet (SMF INTVAL)
- Wenn STATIME = 0 und SMF INTVAL nicht festgelegt ist, findet kein Broadcast statt und es werden keine Intervalldatensätze erzeugt.

## Andere Produkte mit IBM MQ verwenden

Sie können andere Produkte verwenden, um die Darstellung von zu verbessern oder die Statistiken in Bezug auf Leistung und Abrechnung zu erweitern. Beispiel: Resource Measurement Facility, Tivoli Decision Support und Überwachung von CICS.

## Verwenden der Ressourcenmessungsfunktion

Resource Measurement Facility (RMF) ist ein IBM Lizenzprogramm (Programmnummer 5685-029), das systemweite Informationen zur Prozessorauslastung, zur E/A-Aktivität, zum Speicher und zum Paging bereitstellt. Sie können RMF verwenden, um die Auslastung der physischen Ressourcen im gesamten System dynamisch zu überwachen. Weitere Informationen finden Sie im [z/OS Resource Measurement Facility Benutzerhandbuch](#).

## Tivoli Decision Support für z/OS verwenden

Mit Tivoli Decision Support für z/OS können Sie RMF- und SMF-Datensätze interpretieren.

Der Tivoli Decision Support für z/OS ist ein IBM Lizenzprogramm (Programmnummer 5698-B06), mit dem Sie die Leistung Ihres System verwalten können, indem Sie Leistungsdaten in einer Db2-Datenbank erfassen und die Daten in verschiedenen Formaten für die Verwendung in der Systemverwaltung darstellen. Mit Tivoli Decision Support können grafische und tabellarische Berichte mithilfe von Systemmanagementdaten generiert und in der zugehörigen Db2-Datenbank gespeichert werden. Es enthält einen Verwaltungsdialo, einen Berichtsdialog und einen Protokollcollector, die alle mit einer Db2-Standarddatenbank interagieren.

Dies wird in [IBM Tivoli Decision Support for z/OS: Administration Guide and Reference](#) beschrieben.

## CICS-Überwachungsfunktion verwenden

Die CICS-Überwachungsfunktion stellt Leistungsdaten zu jeder aktiven CICS-Transaktion bereit. Es kann verwendet werden, um die verwendeten Ressourcen und die Zeit für die Verarbeitung von Transaktionen zu untersuchen. Hintergrundinformationen finden Sie in der Veröffentlichung [CICS Performance Guide](#) und [Developing CICS System Programs](#) zusammen mit den beiden Referenzhandbüchern, die früher als [CICS Customization Guide](#) bezeichnet wurden.

### **Leistungsprobleme untersuchen**

Leistungsprobleme können sich aus verschiedenen Faktoren ergeben. Beispiel: falsche Ressourcenzuordnung, schlechte Anwendungsgestaltung und E/A-Randbedingungen. Verwenden Sie dieses Thema, um einige der möglichen Ursachen von Leistungsproblemen zu untersuchen.

Die Leistung kann beeinträchtigt werden durch:

- Pufferpools, die eine falsche Größe haben
- Mangel an Realspeicher
- E/A-Konkurrenzsituation für Seitengruppen oder Protokolle
- Protokollpufferschwellenwerte, die falsch festgelegt sind
- Falsche Einstellung der Anzahl Protokollpuffer
- Große Nachrichten
- Einheiten mit Wiederherstellung, die lange Zeit in Anspruch sind und viele Nachrichten für jeden Synchronisationspunkt enthalten
- Nachrichten, die lange Zeit in einer Warteschlange verbleiben
- RACF-Prüfung
- Unnötige Sicherheitsprüfungen
- Ineffizientes Programmdesign

Wenn Sie Leistungsdaten analysieren, beginnen Sie immer mit dem Blick auf das Gesamtsystem, bevor Sie sich entscheiden, dass ein bestimmtes IBM MQ-Problem vorliegt. Denken Sie daran, dass bei einer Konkurrenzsituation fast alle Symptome einer verminderten Leistung vergrößert werden. Wenn z. B. eine Konkurrenzsituation für DASD vorliegt, können die Antwortzeiten der Transaktionen erhöht werden.

Darüber hinaus sind die mehr Transaktionen im System vorhanden, die Prozessorauslastung und die Nachfrage nach virtueller und realer Speicherung.

In solchen Situationen zeigt das System die hohe Verwendung von *all* Ressourcen an. Das System hat jedoch einen normalen Systemstress zu verzeichnen, und dieser Stress könnte die Ursache für eine Leistungsreduzierung sein. Um die Ursache eines solchen Leistungsverlusts zu ermitteln, müssen Sie alle Elemente berücksichtigen, die sich auf Ihre aktiven Tasks auswirken könnten.

## Untersuchung des Gesamtsystems

In IBM MQ ist das Leistungsproblem entweder eine erhöhte Antwortzeit oder eine unerwartete und unerklärliche Verwendung von Ressourcen. Überprüfen Sie zuerst die Faktoren wie die gesamte Prozessorauslastung, die DASD-Aktivität und das Paging. Ein IBM-Tool zur Überprüfung der gesamten Prozessorauslastung ist Resource Management Facility (RMF). Im Allgemeinen müssen Sie das System in einigen Details prüfen, um festzustellen, warum die Tasks langsam vorankommen oder warum eine bestimmte Ressource stark genutzt wird.

Beginnen Sie mit der allgemeinen Taskaktivität und konzentrieren Sie sich dann auf bestimmte Aktivitäten, wie z. B. bestimmte Tasks oder ein bestimmtes Zeitintervall.

Eine andere Möglichkeit besteht darin, dass das System über einen begrenzten Realspeicher verfügt. Daher werden die Tasks aufgrund von Paging-Interrupts langsamer ausgeführt als erwartet.

## Einzeltasks untersuchen

Mit dem Abrechnungstrace können Sie Informationen zu IBM MQ-Tasks erfassen. Diese Trace-Sätze zeigen Ihnen sehr viel über die Aktivität, die die Task ausgeführt hat, und darüber, wie lange die Task ausgesetzt wurde, und wartet auf die Latches. Der Tracesatz enthält außerdem Informationen darüber, wie viele Db2- und Coupling-Facility-Aktivitäten von den Tasks ausgeführt wurden.

Das Interpretieren von IBM MQ-Abrechnungsdaten wird im Abschnitt [„IBM MQ-Abrechnungsdaten interpretieren“](#) auf Seite 366 beschrieben.

Arbeitseinheiten mit langer Laufzeit können durch das Vorhandensein der Nachricht CSQR026I im Jobprotokoll identifiziert werden. Diese Nachricht gibt an, dass eine Task für mehr als drei WS-Manager-Prüfpunkte vorhanden ist und die zugehörigen Protokollsätze nicht mehr angezeigt wurden.  Eine Beschreibung der Protokollsatzverzögerung finden Sie unter [Die Protokolldateien](#).

## IBM MQ-Leistungsstatistik interpretieren

Verwenden Sie diesen Abschnitt als Index für die verschiedenen SMF-Datensätze, die von IBM MQ für z/OS erstellt werden.

IBM MQ-Leistungsstatistik wird als SMF-Datensatz des Typs 115 geschrieben. Statistikdatensätze werden in regelmäßigen Abständen in einem Zeitintervall erstellt, das durch den Parameter STATIME des Systemparametermoduls CSQ6SYSP oder im globalen SMF-Accounting-Intervall angegeben wird, wenn Sie für STATIME den Wert null angeben. Die in den SMF-Datensätzen bereitgestellten Informationen stammen aus den folgenden Komponenten von IBM MQ:

<b>Puffermanager</b>	Verwaltet die Pufferpools im virtuellen Speicher und das Schreiben von Seiten in Seitengruppen, da die Pufferpools voll sind. verwaltet auch das Lesen von Seiten aus Seitengruppen.
<b>Coupling Facility Manager</b>	Verwaltet die Schnittstelle mit der Coupling-Facility.
<b>Datenmanager</b>	Verwaltet die Links zwischen Nachrichten und Warteschlangen. Er ruft den Puffermanager auf, um die Seiten mit Nachrichten auf ihnen zu verarbeiten.
<b>Db2-Manager</b>	Verwaltet die Schnittstelle mit der Db2-Datenbank, die als gemeinsam genutztes Repository verwendet wird.

<b>Sperrenmanager</b>	Verwaltet Sperren für IBM MQ for z/OS.
<b>Protokollmanager</b>	Verwaltet das Schreiben von Protokollsätzen, die für die Aufrechterhaltung der Integrität des Systems von wesentlicher Bedeutung sind, wenn es eine Rückanforderung oder eine Wiederherstellung gibt, wenn ein System- oder Datenträgerfehler aufgetreten ist.
<b>Nachrichtenmanager</b>	Verarbeitet alle IBM MQ-API-Anforderungen
<b>Speichermanager</b>	Verwaltet den Speicher für IBM MQ for z/OS, z. B. Speicherpoolzuordnung, Erweiterung und Freigabe.
<b>Themenmanager</b>	Verwaltet die Themen- und Subskriptionsinformationen für IBM MQ for z/OS.
<b>Coupling-Facility-SMDS-Manager</b>	Verwaltet die gemeinsam genutzten Nachrichtendatengruppen (SMDS) für große Nachrichten, die in der Coupling Facility gespeichert sind.

IBM MQ-Statistikdaten werden in SMF als SMF-Datensätze des Typs 115 geschrieben. Es können die folgenden Subtypen vorhanden sein:

**1**

Systeminformationen, z. B. im Zusammenhang mit den Protokollen und dem Speicher.

**2**

Informationen über die Anzahl der Nachrichten und Paging-Informationen. Informationen zur Gruppe mit gemeinsamer Warteschlange, die sich auf die Coupling-Facility und Db2 beziehen.

**5 und 6**

Ausführliche Informationen zur internen Speicherbelegung im Adressraum des Warteschlangenmanagers. Zwar können Sie diese Informationen anzeigen, einige dieser Informationen sind jedoch nur zur Verwendung durch IBM vorgesehen.

**7**

Übersichtsinformationen zum Speichermanager. Zwar können Sie diese Informationen anzeigen, einige dieser Informationen sind jedoch nur zur Verwendung durch IBM vorgesehen.

**201**

Eingabe-/Ausgabeinformationen für Seitengruppe

**215**

Pufferpoolinformationen

**231**

Systeminformationen für den Adressraum des Kanalinitiators.

Dabei ist Folgendes zu beachten:

- Datensätze des Subtyps 1, 2, 201 und 215 werden mit Statistiktraceklasse 1 erstellt.
- Datensätze des Subtyps 7 werden mit Statistiktraceklasse 2 erstellt.
- Datensätze des Subtyps 5 und 6 werden mit Statistiktraceklasse 3 erstellt.
- Datensätze des Subtyps 231 werden mit Statistiktraceklasse 4 erstellt.

Der Subtyp wird im Feld SM115STF (siehe [Tabelle 34 auf Seite 341](#)) angegeben.

### **Layout eines SMF-Datensatzes vom Typ 115**

Sie können diesen Abschnitt als Referenz für das Format eines SMF-Datensatzes vom Typ 115 verwenden.

Das Standardlayout für SMF-Datensätze umfasst drei Teile:

#### **SMF-Header**

Stellt das Format, die Identifikation und die Uhrzeit- und Datuminformationen zum Datensatz selbst bereit.

## Selbstdefinierendes Kapitel

Definiert die Position und Größe der einzelnen Datensätze innerhalb des SMF-Datensatzes.

### Datensätze

Die tatsächlichen Daten von IBM MQ, die Sie analysieren möchten.

Weitere Informationen zu SMF-Satzformaten finden Sie unter [z/OS MVS System Management Facilities \(SMF\)](#).

### Zugehörige Verweise

„SMF-Header“ auf Seite 341

Verwenden Sie dieses Thema als Referenz für das Format des SMF-Headers.

„Selbstdefinierende Abschnitte“ auf Seite 342

Verwenden Sie dieses Thema als Referenz für das Format der selbstdefinierenden Abschnitte des SMF-Datensatzes.

„Beispiele für SMF-Statistikdatensätze“ auf Seite 343

Verwenden Sie dieses Thema, um einige Beispiele für SMF-Datensätze zu verstehen.

### SMF-Header

Verwenden Sie dieses Thema als Referenz für das Format des SMF-Headers.

Tabelle 34 auf Seite 341 zeigt das Format der SMF-Datensatzkopfdaten (SM115).

Offset: Dez.	Offset: Hex	Typ	Len	Name	Beschreibung	Beispiel
0	0	Struktur	28	SM115	SMF-Datensatz-Header.	
0	0	Integer	2	SM115LEN	SMF-Satzlänge.	14A0
2	2		2		Reserviert.	
4	4	Integer	1	SM115FLG	Systemanzeiger.	5E
5	5	Integer	1	SM115RTY	Satztyp. Der SMF-Datensatztyp für IBM MQ-Statistikdatensätze ist immer 115 (X'73').	73
6	6	Integer	4	SM115TME	Zeit, wenn ein SMF-Datensatz verschoben wurde.	00355575
10	A	Integer	4	SM115DTE	Datum, an dem der SMF-Datensatz verschoben wurde	0100223F
14	E	Zeichen	4	SM115SID	z/OS-Subsystem-ID. Definiert das z/OS-Subsystem, auf dem die Datensätze erfasst wurden.	D4E5F4F1 (MV41)
18	12	Zeichen	4	SM115SSI	IBM MQ-Subsystem-ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM115STF	Satzsubtyp.	0002
24	18	Zeichen	3	SM115REL	IBM MQ-Version.	F6F0F0 (600)
27	1B		1		Reserved	
28	1C	Zeichen	0	SM115END	Ende des SMF-Headers und Beginn des selbstdefinierenden Abschnitts.	

## z/OS **Selbstdefinierende Abschnitte**

Verwenden Sie dieses Thema als Referenz für das Format der selbstdefinierenden Abschnitte des SMF-Datensatzes.

Ein selbstdefinierter Abschnitt eines SMF-Satzes vom Typ 115 teilt Ihnen mit, wo ein Statistikdatensatz zu finden ist, wie lange es ist und wie oft dieser Datensatztyp wiederholt wird (mit unterschiedlichen Werten). Die selbstdefinierenden Abschnitte folgen dem Header, bei fixen Offsets vom Anfang des SMF-Datensatzes. Jeder Statistikdatensatz kann durch eine Blickfängerzeichenfolge identifiziert werden.

Die folgenden Typen von selbstdefinierenden Schnitten stehen Benutzern für Typen 115-Datensätze zur Verfügung. Jeder selbstdefinierende Abschnitt verweist auf Statistikdaten, die sich auf eine der IBM MQ-Komponenten beziehen. Tabelle 35 auf Seite 342 fasst die Quellen der Statistiken, die Strukturkennungszeichenfolge und die Offsets der selbstdefinierenden Abschnitte vom Beginn der SMF-Datensatzkopfdaten zusammen.

*Tabelle 35. Offsets für selbstdefinierende Abschnitte*

Quelle der Statistik	Satzsubtyp (SM115STF)	Offset des selbstdefinierenden Schnittabschnitts		Eye-Catcher von Daten
		Dez.	Hex.	
Speichermanager	1	100	X'64 '	QSST
Protokollmanager	1	116	X'74 '	QJST
Nachrichtenmanager	2	36	X'24 '	QMST
Datenmanager	2	44	X'2C '	QIST
 Nicht mehr verwendet. Der selbstdefinierende Abschnitt wird binäre Nullen sein.	2	52	X'34 '	
Sperrmanager	2	60	X'3C '	QLST
Db2-Manager	2	68	X'44 '	Q5ST
Coupling Facility-Manager	2	76	X'4C '	QEST
Themenmanager	2	84	X'54 '	QTST
SMDS-Verwendung	2	92	X'5C '	QESD
Puffermanager-eine für jeden Pufferpool	215	36	X'24 '	QPST
Kanalinitiator	231			QWSX
Seitengruppe des Datenmanagers-eine für jede Seitengruppe	201	36	X'24 '	QIS1
Speichermanager	5	36	X'24 '	QSPH
Speichermanager	6	36	X'24 '	QSGM
Speichermanager	7	36	X'24 '	QSRS

**Anmerkung:** Einige Speichermanagerinformationen in Datensätzen des Subtyps 5, 6 und 7 sind nur für die Verwendung durch IBM gedacht. Andere nicht aufgeführte selbstdefinierende Abschnitte enthalten nur Daten zur Verwendung durch IBM.

Jeder selbstdefinierende Abschnitt ist zwei Vollwörter lang und hat das folgende Format:

sssssssllllnnnn

Dabei gilt:

- ssssssss ist ein Vollwort, das den Offset vom Anfang des SMF-Datensatzes enthält.
- llll ist ein Halbwort, das die Länge dieses Datensatzes ergibt.
- nnnn ist ein Halbwort, das die Anzahl der Datensätze in diesem SMF-Datensatz angegeben hat.

Weitere Informationen finden Sie in „Beispiele für SMF-Statistikdatensätze“ auf Seite 343.

**Anmerkung:** Verwenden Sie in den selbstdefinierenden Abschnitten immer Offsets, um die Statistikdatensätze zu lokalisieren.

### **Beispiele für SMF-Statistikdatensätze**

Verwenden Sie dieses Thema, um einige Beispiele für SMF-Datensätze zu verstehen.

Abbildung 20 auf Seite 343 zeigt ein Beispiel für einen Teil des SMF-Datensatzes für Subtyp 1. Subtyp 1 enthält die Statistikdatensätze des Speicheranagers und des Protokollanagers. Der SMF-Datensatz-Header wird unterstrichen angezeigt.

Der selbstdefinierende Abschnitt an der relativen Position X'64' bezieht sich auf die Speicheranagertatistik, und der selbstdefinierende Abschnitt an der relativen Position X'74' bezieht sich auf Protokollanagertatistiken, die beide in **Fettdruck** angezeigt werden.

Der Speicheranagertatistikdatensatz befindet sich bei der relativen Position X'0000011C' am Anfang des Headers und ist X'48' Byte lang. Es gibt eine Gruppe von Speicheranagertatistikdaten, die durch die Blickfängerfolge QSST angegeben wird. Der Anfang dieses Statistikdatensatzes wird auch im Beispiel gezeigt.

Der Protokollanagertatistikdatensatz befindet sich bei der relativen Position X'00000164' am Anfang des Headers und ist X'78' Byte lang. Es gibt eine Gruppe von Protokollanagertatistikdaten, die durch die Blickfang-Zeichenfolge QJST identifiziert wird.

```
000000 02000000 5E730035 55750100 223FD4E5 *.....;.....MV*
000010 F4F1D4D8 F0F70001 F6F0F000 000001DC *41MQ07..600....*
000020 00240001 00000000 00000000 00000000 *.....*
000030 00000000 00000000 00000000 0000007C *.....@*
000040 00400001 000000BC 00600001 00000000 *.....~.....*
000050 00000000 00000000 00000000 00000000 *.....*
000060 00000000 000011C 00480001 00000000 *.....*
000070 00000000 0000164 00780001 00000000 *.....*
000080 00000000 00000000 00000000 00000000 *.....*
:
000110 00000000 00000000 00000000 003C0048 *.....*
000120 D8E2E2E3 0000004F 00000003 00000002 *QSST...|.....*
```

Abbildung 20. SMF-Datensatz 115, Subtyp 1

Abbildung 21 auf Seite 344 zeigt ein Beispiel für einen Teil des SMF-Datensatzes für den Subtyp 2. Subtyp 2 enthält die Statistikdatensätze für den Nachrichten-, Daten-, Sperren-, Coupling-Facility-, Themen- und Db2-Manager. Der SMF-Datensatz-Header ist unterstrichen dargestellt. Die selbstdefinierenden Abschnitte werden abwechselnd **fett** und *kursiv* angezeigt.

- Der selbstdefinierende Abschnitt an der relativen Position X'24' bezieht sich auf die Statistik des Nachrichtenmanagers. Der Statistikdatensatz für den Nachrichtenmanager befindet sich bei der relativen Position X'00000064' am Anfang des Headers und ist X'48' Byte lang. Es gibt eine Gruppe dieser Statistiken, die durch die Blickfang-Zeichenfolge QMST identifiziert wird.

- Der selbstdefinierende Abschnitt bei Offset X'2C ' bezieht sich auf die Datenmanagerstatistik. Der Statistikdatensatz für den Datenmanager befindet sich bei der relativen Position X'00000AC 'am Anfang des Headers und ist X'50' Byte lang. Es gibt eine Gruppe dieser Statistiken, die durch die Blickfolgerzeichenfolge QIST identifiziert wird.
- **V 9.1.0** In früheren Releases hat der selbstdefinierende Abschnitt an der relativen Position X'34 ' auf die Statistik des Puffermanagers verwiesen. Da dieser SMF-Datensatz von einem IBM MQ 9.1.0-Warteschlangenmanager übernommen wurde, wird der selbstdefinierende Abschnitt des Puffermanagers auf Nullen gesetzt, um anzuzeigen, dass keine Puffermanagerstatistik vorhanden ist. Stattdessen befinden sich diese Statistiken in SMF 115 -Datensätzen des Subtyps 215.
- Der selbstdefinierende Abschnitt an der relativen Position X'3C ' bezieht sich auf Sperrmanagerstatistiken. Der Sperrmanager-Statistikdatensatz befindet sich bei der relativen Position X'00000FC 'am Anfang des Headers und ist X'20' Byte lang. Es gibt eine Gruppe dieser Statistiken, die durch die Blickfang-Zeichenfolge QLST identifiziert wird.
- Der selbstdefinierende Abschnitt bei Offset X'44' bezieht sich auf die Db2-Managerstatistik. Der Statistikdatensatz für den Db2-Manager befindet sich an der relativen Position X'0000011C' am Anfang des Headers und ist X'2A0' Byte lang. Es gibt eine Gruppe dieser Statistiken, die durch die Blickfang-Zeichenfolge Q5ST gekennzeichnet ist.
- Der selbstdefinierende Abschnitt bei der relativen Position X'4C ' bezieht sich auf Coupling Facility Manager-Statistiken. Der Coupling Facility-Manager-Statistikdatensatz befindet sich bei der relativen Position X'000003BC 'vom Anfang des Headers und ist X'1008' Byte lang. Es gibt eine Gruppe dieser Statistiken, die durch die Blickfängerzeichenfolge QEST angegeben wird.
- Der selbstdefinierende Abschnitt an der relativen Position X'54 ' bezieht sich auf die Statistik des Topic-Managers. Der Topic-Manager-Statistikdatensatz befindet sich bei der relativen Position X'000013C4 'am Anfang des Headers und ist X'64' Byte lang. Es gibt eine Gruppe dieser Statistiken, die durch die Blickfängerzeichenfolge QTST angegeben wird.
- Der selbstdefinierende Abschnitt an der relativen Position X'5C ' ist für SMDS-Statistiken bestimmt. Dieser selbstdefinierende Abschnitt wird auf Nullen gesetzt, die angeben, dass SMDS nicht verwendet wird.

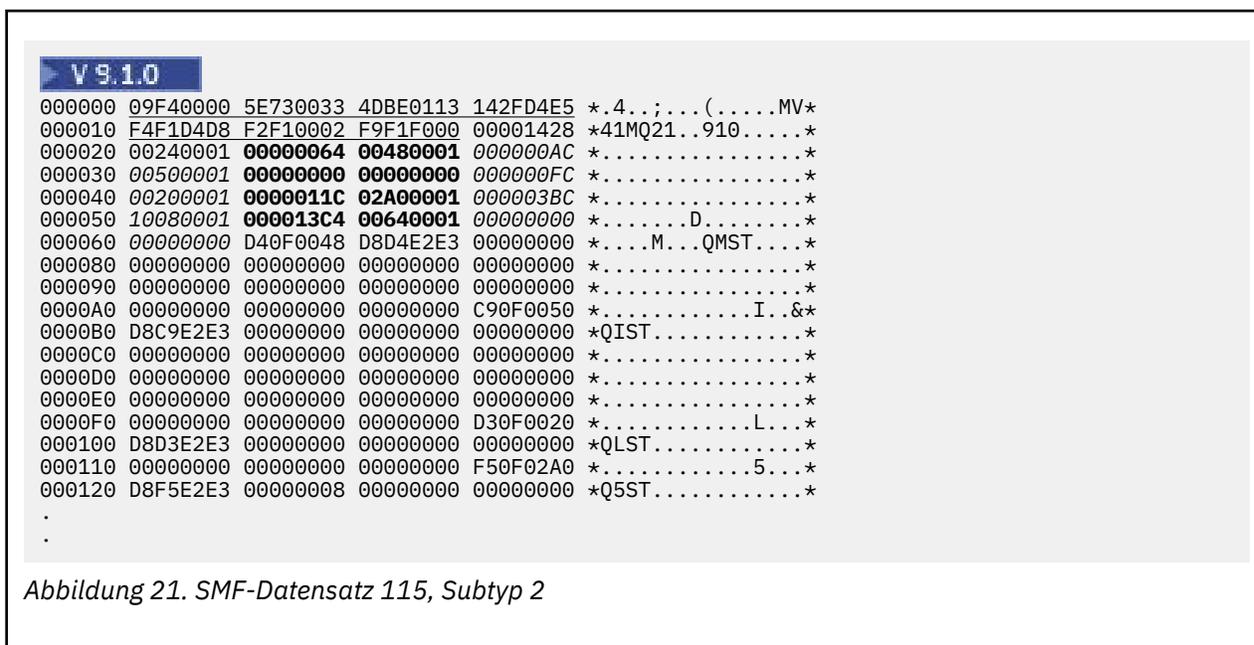


Abbildung 21. SMF-Datensatz 115, Subtyp 2

## **Z/OS** Verarbeitung von SMF-Datensätzen des Typs 115

Verwenden Sie dieses Thema als Referenz für die Verarbeitung von SMF-Datensätzen des Typs 115.

Sie müssen alle Daten, die Sie von SMF erfassen, verarbeiten, um nützliche Informationen zu extrahieren. Überprüfen Sie beim Verarbeiten der Daten, dass die Datensätze aus IBM MQ stammen und es sich dabei um die von Ihnen erwarteten Datensätze handelt.

Validieren Sie die Werte der folgenden Felder:

- SM115RTY, die SMF-Satznummer, muss X'73 ' (115) sein.
- SM115STF, der Satzsubtyp, muss 0001, 0002, 0005, 0006, 0007, 0201, 0215 oder 0231 sein.

Das Lesen aus den aktiven SMF-Dateigruppen (oder SMF-Protokolldatenströmen) wird nicht unterstützt. Sie müssen das SMF-Programm IFASMFDP (bzw. bei Verwendung von Protokolldatenströmen das SMF-Programm IFASMF DL) verwenden, um einen Speicherauszug von SMF-Datensätzen in einer sequenziellen Datei zu erstellen, damit sie verarbeitet werden können. Weitere Informationen finden Sie im Abschnitt „Systemverwaltungsfunktion verwenden“ auf Seite 335.

Einzelheiten zu den Strukturen und Feldern finden Sie im IBM MQ-SupportPac [MP1B](#).

Es gibt ein C-Beispielprogramm namens CSQ4SMFD, das den Inhalt von SMF-Datensätzen des Typs 115 und 116 aus der sequenziellen Datei ausgibt. Das Programm wird als Quelle in thlqual.SCSQC37S und im ausführbaren Format in thlqual.SCSQLOAD. bereitgestellt. Die Beispiel-JCL wird in thlqual.SCSQPROC (CSQ4SMFJ) bereitgestellt.

## **Speichermanagerdatensätze**

Verwenden Sie dieses Thema als Referenz für Speichermanagerdatensätze.

Das Format des Statistikdatensatzes für den Speichermanager wird im Assemblermakro thlqual.SCSQMACS (CSQDQSST) beschrieben.

Die Daten enthalten Informationen zur Anzahl der festen und variablen Speicherpools, die der WS-Manager während des Statistikintervalls zugeordnet, erweitert, komprimiert und gelöscht hat, sowie die Anzahl der GETMAIN-, FREEMAIN- und STORAGE-Anforderungen an z/OS, einschließlich der Anzahl der nicht erfolgreichen Anforderungen. Zu den zusätzlichen Informationen gehören die Anzahl der festgestellten Speicherkennungen und die Anzahl der abnormalen Beendigungen, die als Ergebnis dieser Bedingung aufgetreten sind.

Zusätzliche Daten zur Speicherbelegung im Warteschlangenmanager werden vom Statistiktrace der Klasse 2 und 3 erstellt. Zwar können Sie diese Informationen anzeigen, einige dieser Informationen sind jedoch nur zur Verwendung durch IBM vorgesehen.

- Das Format des Statistikdatensatzes für den Speichermanagerpool-Header, der in Datensätzen des Subtyps 5 vorhanden ist, wird im Assemblermakro thlqual.SCSQMACS (CSQDQSPH) beschrieben.
- Das Format des getmain-Statistikdatensatzes für den Speichermanager, der in Datensätzen des Subtyps 6 vorhanden ist, wird im Assemblermakro thlqual.SCSQMACS (CSQDQSGM) beschrieben.
- Das Format des region-Summensatzes für den Speichermanager, der in Datensätzen des Subtyps 7 vorhanden ist, wird im Assemblermakro thlqual.SCSQMACS (CSQDQSR S) beschrieben.

## **Protokollmanagerdatensätze**

Verwenden Sie dieses Thema als Referenz für das Format von Protokollmanagerdatensätzen.

Das Format des Protokollmanagerstatistikdatensatzes wird im Assemblermakro thlqual.SCSQMACS (CSQDQJST) beschrieben.

In den Statistiken sind diese Zahlen wichtig:

1. Die Gesamtzahl der Protokollschreibanforderungen:

$$N_{\logwrite} = QJSTWRNW + QJSTWRF$$

2. Die Gesamtzahl der Protokolleleseanforderungen:

$$N_{\logread} = QJSTRBUF + QJSTRACT + QJSTRARH$$

Die Fehlersymptome, die anhand der Protokollmanagerstatistik untersucht werden können, werden in der folgenden Tabelle beschrieben.

**Symptom 1**

QJSTWTB ist ungleich null.

**Ursache**

Tasks werden ausgesetzt, während der In-Storage-Puffer in das aktive Protokoll geschrieben wird.

Möglicherweise gibt es Probleme beim Schreiben in das aktive Protokoll.

Der Parameter OUTBUFF in CSQ6LOGP ist zu klein.

**Action**

Schreiben Sie die Probleme auf, die in das aktive Protokoll geschrieben werden.

Erhöhen Sie den Wert des Parameters OUTBUFF innerhalb von CSQ6LOGP.

**Symptom 2**

Das Verhältnis:  $QJSTWTL/N_{logread}$  ist größer als 1%.

**Ursache**

Es wurden Protokollesevorgänge eingeleitet, bei denen aus einem Archivprotokoll gelesen werden muss. IBM MQ konnte allerdings kein Dataset zuordnen, da MAXRTU-Datasets bereits zugeordnet waren.

**Action**

Erhöhen Sie MAXRTU.

**Symptom 3**

Das Verhältnis:  $QJSTRARH/N_{logread}$  ist größer als normal.

**Ursache**

Die meisten Protokolleseanforderungen sollten aus dem Ausgabepuffer oder dem aktiven Protokoll stammen. Um Anforderungen für ein Zurückstellen zu erfüllen, werden die Arbeitseinheitenwiederherstellungsdatensätze aus dem speicheraktiven Puffer, dem aktiven Protokoll und den archivierten Protokollen gelesen.

Eine lange Arbeitseinheit mit Wiederherstellung, die sich über einen Zeitraum von mehreren Minuten erstreckt, enthält möglicherweise Protokollsätze, die sich über viele verschiedene Protokolle erstrecken. Dies beeinträchtigt die Leistung, da zusätzliche Arbeiten zur Wiederherstellung der Protokollsätze ausgeführt werden müssen.

**Action**

Ändern Sie die Anwendung, um die Länge einer Arbeitseinheit mit Wiederherstellung zu reduzieren. Darüber hinaus sollten Sie die Größe des aktiven Protokolls erhöhen, um die Möglichkeit einer einzelnen Arbeitseinheit zu reduzieren, die sich auf mehr als ein Protokoll ausbreitet.

**Andere Zeiger**

Das Verhältnis  $N_{logread} / N_{logwrite}$  gibt einen Hinweis darauf, wie viel Arbeit zurückgesetzt werden muss.

**Symptom 4**

QJSTLLCP ist mehr als 10 Stunden pro Stunde.

**Ursache**

Auf einem ausgelasteten System erwarten Sie in der Regel 10 Prüfpunkte pro Stunde. Wenn der Wert für QJSTLLCP größer als dieser Wert ist, weist er auf ein Problem in der Konfiguration des Warteschlangenmanagers hin.

Der wahrscheinlichste Grund hierfür ist, dass der Parameter LOGLOAD in CSQ6SYSP zu klein ist. Das andere Ereignis, das einen Prüfpunkt verursacht, ist, wenn ein aktives Protokoll voll ist, und wechselt zu der nächsten aktiven Protokolldatei. Wenn Ihre Protokolle zu klein sind, kann dies zu häufigen Prüfpunkten führen.

QJSTLLCP ist die Anzahl der Gesamtanzahl der Prüfpunkte.

**Action**

Erhöhen Sie den Wert für den Parameter LOGLOAD oder erhöhen Sie die Größe der Protokolldatensätze nach Bedarf.

**Symptom 5**

QJSTCmpFail > 0 oder QJSTCmpComp nicht viel kleiner als QJSTCmpUncmp

**Ursache**

Der Warteschlangenmanager kann die Protokollsätze nicht wesentlich komprimieren.

QJSTCmpFail ist die Anzahl der Male, die der Warteschlangenmanager nicht in der Lage war, eine Reduzierung der Satzlänge zu erreichen. Vergleichen Sie die Zahl mit QJSTCmpReq (Anzahl der Komprimierungsanforderungen), um festzustellen, ob die Anzahl der Fehler signifikant ist.

QJSTCmpComp ist die Summe der komprimierten Byte, die in das Protokoll geschrieben werden, und QJSTCmpUncmp ist die Summe der Byte vor der Komprimierung. Die Summe enthält keine Byte, die für Protokollsätze geschrieben wurden, die nicht für die Komprimierung auswählbar waren. Wenn die Zahlen ähnlich sind, hat die Komprimierung wenig Nutzen erzielt.

**Action**

Schalten Sie die Protokollkomprimierung aus. Setzen Sie den Befehl SET LOG COMPLOG (NONE) ab. Weitere Informationen finden Sie im Befehl [SET LOG](#).

**Anmerkung:** In der ersten Gruppe von Statistiken, die nach dem Systemstart erstellt wurden, kann es aufgrund der Auflösung von In-Flight-Einheiten der Wiederherstellung zu einer erheblichen Protokollaktivität kommen.

**z/OS Nachrichtenmanagerdatensätze**

Verwenden Sie dieses Thema als Referenz für die Datensätze des Nachrichtenmanagers.

Das Format des Statistikdatensatzes für den Nachrichtenmanager wird im Assemblermakro thlqual.SCSQMACS (CSQDQMST) beschrieben.

Die Daten geben die Anzahl von verschiedenen IBM MQ-API-Anforderungen an.

**z/OS Datendatensätze des Datenmanagers**

Verwenden Sie dieses Thema als Referenz für das Format der Data Manager-Datendatensätze.

Das Format des Statistikdatensatzes für den Datenmanager wird im Assemblermakro thlqual.SCSQMACS (CSQDQIST) beschrieben.

Die Daten geben Ihnen die Anzahl unterschiedlicher Objektanforderungen an.

**z/OS Datensatzdatensätze für den Datenmanager**

Verwenden Sie diesen Abschnitt als Referenz für das Format der Datensatzdatensätze für den Datenmanager.

Das Format des Statistikdatensatzes für den Datenmanagerseitensatz wird im Assemblermakro `thlqual.SCSQMACS (CSQDQIS1)` beschrieben.

Die Nutzungsinformationen zur Seitengruppe erleichtern die Verwaltung von lokalen Warteschlangen in einem Warteschlangenmanager, indem Informationen wie Ein-/Ausgaberraten von Seitengruppen und die höchste Auslastung aufgezeichnet werden.

Die Daten stellen dieselben grundlegenden Seitenseiteninformationen wie die Ausgabe des MQSC-Befehls **DISPLAY USAGE TYPE (PAGESET)** oder der PCF-Befehl `Inquire Usage ( MQCMD_INQUI-RE_USAGE )` bereit.

Beispiel:

- Gesamtseiten
- Die aktuell verwendeten Seiten
- Nicht verwendete persistente und nicht persistente Seiten
- Erweiterungsmethode
- Anzahl der Erweiterung
- Anzahl Stripekonfigur

Die Daten stellen auch einige Leistungsindikatoren zusammen mit anderen Leistungsinformationen zur Verfügung. Beispiel:

- Wie viele verzögerte Schreib-, Sofortschreibungs- und Lese-Seiten-E/A-Anforderungen während des SMF-Intervalls aufgetreten sind,
- Die Anzahl der umbewegten Seiten, die abgelaufene Zeit und die Anzahl der Lese- und Schreiboperationen.
- Wie viele Seiten in Prüfpunkten geschrieben wurden.
- Ist die Erweiterung während des SMF-Intervalls aufgetreten?
- Gibt an, wie oft die Seitengruppe voll wurde.
- Ein Hinweis darauf, wo innerhalb der Seitengruppe neuer Speicherbereich zugeordnet wird.

Aus den angezeigten Informationen sollten Sie den allgemeinen Status der einzelnen Seitengruppe verstehen und überlegen, ob Sie das System neu einstellen müssen.

### **Zugehörige Verweise**

„SMF-Header“ auf Seite 341

Verwenden Sie dieses Thema als Referenz für das Format des SMF-Headers.

„Selbstdefinierende Abschnitte“ auf Seite 342

Verwenden Sie dieses Thema als Referenz für das Format der selbstdefinierenden Abschnitte des SMF-Datensatzes.

„Beispiele für SMF-Statistikdatensätze“ auf Seite 343

Verwenden Sie dieses Thema, um einige Beispiele für SMF-Datensätze zu verstehen.

### **Puffermanager-Datensätze**

Verwenden Sie dieses Thema als Referenz für das Format von Puffermanagerdatensätzen.

Das Format des Statistikdatensatzes für den Puffermanager wird im Assemblermakro `thlqual.SCSQMACS (CSQDQPST)` beschrieben.

**Anmerkung:** Puffermanager-Statistikdatensätze werden nur für Pufferpools erstellt, die definiert sind. Wenn ein Pufferpool definiert, aber nicht verwendet wird, werden keine Werte festgelegt, und der zugehörige Statistikdatensatz für den Puffermanager enthält keine Daten.

Informationen zur effizienten Verwaltung Ihrer Pufferpools finden Sie unter [„Pufferpools verwalten“](#) auf Seite 350.

Bei der Interpretation der Statistikdaten wird empfohlen, die folgenden Faktoren zu berücksichtigen, da die Werte dieser Felder verwendet werden können, um die Leistung Ihres Systems zu verbessern:

1. Wenn QPSTSOS, QPSTDMC oder QPSTIMW größer als null ist, sollten Sie entweder die Größe des Pufferpools erhöhen oder die Seitengruppen auf verschiedene Pufferpools umstellen.
  - QPSTSOS ist die Anzahl der Male, die keine Puffer für die Seitenaufforderungen zur Verfügung standen. Wenn QPSTSOS jemals ungleich null ist, wird angezeigt, dass IBM MQ unter starker Belastung steht. Die Pufferpoolgröße sollte erhöht werden. Wenn der Wert von QPSTSOS 0 durch eine Erhöhung der Pufferpoolgröße nicht erhöht wird, kann es zu einer E/A-Konkurrenzsituation auf den DASD-Seitengruppen kommen.
  - QPSTDMC ist die Anzahl der Aktualisierungen, die synchron ausgeführt wurden, da entweder mehr als 95% der Seiten im Pufferpool auf Schreib-E/A warteten, oder es waren weniger als 5% des Pufferpools für Leseanforderungen verfügbar. Wenn diese Zahl nicht null ist, ist der Pufferpool möglicherweise zu klein und sollte vergrößert werden. Wenn die Größe des Pufferpools nicht die Größe von QPSTDMC auf Null reduziert, kann es zu E/A-Konflikten auf den DASD-Seitengruppen kommen.
  - QPSTIMW ist die Anzahl der Male, die die Seiten synchron geschrieben wurden. Wenn QPSTDMC null ist, ist QPSTIMW die Anzahl der Male, die Seiten in der Warteschlange gefunden wurden, die auf Schreib-E/A warteten, die für mindestens zwei Prüfpunkte vorhanden war.
2. Für Pufferpoolnull und Pufferpools, die kurzlebige Nachrichten enthalten:
  - QPSTDWT muss null sein, und der Prozentsatz QPSTCBSL/QPSTNBUF sollte größer als 15% sein.  
QPSTDWT gibt an, wie oft der asynchrone Schreibprozessor gestartet wurde, da entweder mehr als 85% der Seiten im Pufferpool auf Schreib-E/A warteten, oder es waren weniger als 15% des Pufferpools für Leseanforderungen verfügbar. Eine Erhöhung der Pufferpoolgröße sollte diesen Wert verringern. Ist dies nicht der Fall, ist das Muster des Zugriffs einer der langen Verzögerungen zwischen den Einreisungen und den Abrufe.
  - QPSTTPW kann aufgrund einer Prüfpunktaktivität größer als null sein.
  - QPSTRIO sollte null sein, es sei denn, die Nachrichten werden von einer Seitengruppe gelesen, die nach einem Neustart des Warteschlangenmanagers festgelegt wurde.  
Das Verhältnis von QPSTRIO zu QPSTGETP zeigt die Effizienz des Seitenabrufs innerhalb des Pufferpools an. Eine Erhöhung der Pufferpoolgröße sollte dieses Verhältnis verringern und somit die Effizienz bei der Auslagerungseffizienz erhöhen. Wenn dies nicht der Fall ist, wird angezeigt, dass die Seiten nicht häufig erneut aufgerufen werden. Dies impliziert ein Transaktionsmuster, bei dem eine lange Verzögerung zwischen den Nachrichten, die gestellt werden, und dem späteren Abruf besteht.  
Das Verhältnis von QPSTGETN zu QPSTGETP gibt die Häufigkeit an, mit der eine leere Seite angefordert wurde, im Gegensatz zu einer nicht leeren Seite. Dieses Verhältnis ist eher ein Hinweis auf ein Transaktionsmuster als ein Wert, der verwendet werden kann, um das System zu optimieren.
  - Wenn QPSTSTL einen Wert größer als null hat, bedeutet dies, dass die zuvor noch nicht verwendeten Seiten verwendet werden. Dies kann durch eine höhere Nachrichtenrate verursacht werden, Nachrichten, die nicht so schnell verarbeitet werden, wie sie zuvor (zu einem Aufbau von Nachrichten) waren, oder größere Nachrichten, die verwendet werden.  
QPSTSTL ist die Anzahl der Male, die eine Seitenzugriffsanforderung die Seite, die sich bereits im Pufferpool befindet, nicht gefunden hat. Erneut ist das Verhältnis von QPSTSTL zu (QPSTGETP + QPSTGETN) um so größer, dass die Auslagerungs-Effizienz höher ist. Eine Erhöhung der Pufferpoolgröße sollte dieses Verhältnis verringern, aber wenn dies nicht der Fall ist, deutet dies darauf hin, dass es lange Verzögerungen zwischen den Abrufe und den Abrufe gibt.
  - Es wird empfohlen, genügend Puffer zu haben, um die höchste Nachrichtenrate zu verarbeiten.
3. Für Pufferpools mit langlebigen Nachrichten, bei denen mehr Nachrichten vorhanden sind, als in den Pufferpool passen können:
  - (QPSTRIO + QPSTWIO) /Statistikintervall ist die E/A-Rate für Seitengruppen. Wenn dieser Wert hoch ist, sollten Sie in Betracht ziehen, mehrere Seitengruppen auf verschiedenen Datenträgern zu verwenden, damit die Ein-/Ausgabe parallel ausgeführt werden kann.
  - Während des Zeitraums, in dem die Nachrichten verarbeitet werden (z. B. wenn Nachrichten während des Tages in eine Warteschlange geschrieben und über Nacht verarbeitet werden), sollte die

Anzahl der gelesenen Ein-/Ausgaben (QPSTRIO) ungefähr die Gesamtzahl der geschriebenen Seiten (QPSTTPW) sein. Dies zeigt, dass für jede geschriebene Seite eine Seite gelesen wird.

Wenn QPSTRIO viel größer als QPSTTPW ist, zeigt dies, dass die Seiten mehrfach gelesen werden. Dies kann das Ergebnis der Anwendung sein, die MQGET von *MsgId* oder *CorrelId* verwendet, wenn die Warteschlange nicht indexiert ist, oder wenn Sie Nachrichten in der Warteschlange verwenden, die als Nächstes verwendet werden.

Mit den folgenden Aktionen kann dieses Problem behoben werden:

- a. Erhöhen Sie die Größe des Pufferpools, so dass genügend Seiten vorhanden sind, um die Warteschlange zusätzlich zu den geänderten Seiten zu halten.
- b. Verwenden Sie das Warteschlangenattribut INDXTYPE, mit dem eine Warteschlange von *MsgId* oder *CorrelId* indexiert werden kann, und die Notwendigkeit einer sequenziellen Suche in der Warteschlange wird nicht mehr benötigt.
- c. Ändern Sie das Design der Anwendung so, dass die Verwendung von MQGET mit *MsgId* oder *CorrelId* oder die Option zum Abrufen der nächsten mit der Option zum Durchsuchen entfernt wird.

**Anmerkung:** Anwendungen, die langlebige Nachrichten verwenden, verarbeiten normalerweise die erste verfügbare Nachricht und verwenden MQGET nicht mit *MsgId* oder *CorrelId*, und sie können nur die erste verfügbare Nachricht durchsuchen.

- d. Verschieben Sie die Seitengruppen in einen anderen Pufferpool, um die Konkurrenzsituation zwischen den Nachrichten von verschiedenen Anwendungen zu verringern.

#### Pufferpools verwalten

Um Ihre Pufferpools effizient zu verwalten, müssen Sie die Faktoren berücksichtigen, die sich auf die E/A-Operationen des Pufferpools und auch auf die Statistiken auswirken, die den Pufferpools zugeordnet sind.

Die folgenden Faktoren wirken sich auf die E/A-Operationen des Pufferpools aus.

- Wenn eine Seite, die die erforderlichen Daten enthält, nicht im Pufferpool gefunden wird, wird sie synchron zu einem verfügbaren Puffer auf der DASD-Seitengruppe gelesen.
- Immer wenn eine Seite aktualisiert wird, wird sie in eine interne Warteschlange von Seiten gestellt, die (möglicherweise) in die DASD-Einheit geschrieben werden sollen. Dies bedeutet, dass der von dieser Seite verwendete Puffer für die Verwendung durch eine andere Seite nicht verfügbar ist, bis der Puffer in die DASD-Einheit geschrieben wurde.
- Wenn die Anzahl der Seiten, die in die DASD-Einheit geschrieben werden, 85% der Gesamtzahl der Puffer im Pool überschreitet, wird ein asynchroner Schreibprozessor gestartet, um die Puffer auf die DASD-Einheit zu stellen.

Wenn die Anzahl der Puffer, die für Seitenaufforderungen zur Verfügung stehen, weniger als 15% der Gesamtzahl der Puffer im Pool beträgt, wird der asynchrone Schreibprozessor gestartet, um die Schreiboperationen für E/A-Operationen auszuführen.

Der Schreibprozessor stoppt, wenn die Anzahl der Seiten, die in die DASD-Einheit eingereicht wurden, auf 75% der Gesamtzahl der Puffer im Pool gefallen ist.

- Wenn die Anzahl der für das Schreiben in die DASD-Einheit in die Warteschlange gestellten Seiten 95% der Gesamtzahl der Puffer im Pool überschreitet, führen alle Aktualisierungen zu einem synchronen Schreiben der Seite in die DASD-Einheit.

Wenn die Anzahl der Puffer, die für Seitenaufforderungen zur Verfügung stehen, weniger als 5% der Gesamtzahl der Puffer im Pool beträgt, werden alle Aktualisierungen zu einem synchronen Schreibzugriff auf die Seite in die DASD-Einheit.

- Wenn die Anzahl der Puffer, die für Anforderungen zum Abrufen von Seiten verfügbar sind, immer null erreicht, wird eine Transaktion ausgesetzt, die diese Bedingung feststellt, bis der asynchrone Schreibprozessor beendet ist.

- Wenn eine Seite häufig aktualisiert wird, verbringt die Seite den Großteil ihrer Zeit in der Warteschlange der Seiten, die darauf warten, auf die DASD-Einheit geschrieben zu werden. Da diese Warteschlange mindestens kürzlich verwendet wurde, ist es möglich, dass eine häufig aktualisierte Seite, die in diese zuletzt verwendete Warteschlange gestellt wird, nie in die DASD-Einheit geschrieben wird. Aus diesem Grund wird zum Zeitpunkt der Aktualisierung, wenn die Seite gefunden wurde, dass sie für mindestens zwei Prüfpunkte auf die Schreiboperation in der DASD-Warteschlange gewartet hat, sie synchron in die DASD-Einheit geschrieben. Die Aktualisierung erfolgt an der Prüfpunktzeit und wird ausgesetzt, bis der asynchrone Schreibprozessor beendet ist.

Das Ziel dieses Algorithmus ist es, die Zeitseiten im Pufferpoolspeicher zu maximieren, während das System funktioniert, wenn die Belastung des Pufferpools durch die Systembelastung unter Stress ist.

### **Datensätze für Sperrenmanager**

Verwenden Sie dieses Thema als Referenz auf das Format der Datensätze des Sperrenmanagers.

Das Format des Statistikdatensatzes für den Sperrmanager wird im Assemblermakro `thlqual.SCSQMACS(CSQDQLST)` beschrieben.

Die Datensätze enthalten Daten zu den folgenden Informationen:

- Die Anzahl der Sperrenanforderungen und Sperrenfreigabebeanforderungen.
- Die Anzahl der Male, die eine Sperrenanforderung festgestellt hat, dass die angeforderte Sperre bereits angehalten wurde.

### **Datensätze für den Db2-Manager**

Verwenden Sie diesen Abschnitt als Referenz für das Format der Datensätze für den Db2-Manager.

Das Format der Statistikdatensätze für den Db2-Manager wird in der folgenden Tabelle und im Assemblermakro `'thlqual.SCSQMACS(CSQDQ5ST)'` sowie in der C-Headerdatei `'thlqual.SCSQC370(CSQDSMFC)'` beschrieben. Die Feldnamen in C sind alle in Kleinbuchstaben angegeben, z. B. `q5st`, `q5stid`.

Wenn der Warteschlangenmanager nicht als Mitglied einer Gruppe mit gemeinsamer Warteschlange gestartet wurde, werden in diesem Datensatz keine Daten aufgezeichnet.

Offset: Dez.	Offset: Hex	Typ	Len	Name	Beschreibung
0	0	Struktur	668	Q5ST	Statistikdaten für den Db2-Manager
0	0	Bitfolge	2	Q5STID	Steuerblock-ID
2	2	Integer	2	Q5STLL	Steuerblocklänge
4	4	Zeichen	4	Q5STEYEC	Steuerungsblock-Blickfang
8	8	Zeichen	660	Q5STZERO	QMST-Teil bei Gelegenheit gelöscht
8	8	Integer	4	ANZAHL TAS	Anzahl der Server-Tasks
12	C	Integer	4	ACTTASK	Anzahl der aktiven Servertasks
16	10	Integer	4	CONNCNT	Anzahl der Verbindungsanforderungen
20	14	Integer	4	DISCCNT	Anzahl der Verbindungsunterbrechungsanforderungen
24	18	Integer	4	DHIGMAX	Max. Anforderungswarteschlangenlänge
28	1C	Integer	4	ABNDCNT	Anzahl der Abbrüche von Db2-SRV-Tasks
32	20	Integer	4	ERFORDERT	Anzahl der erneut anfordernden Anforderungen

Tabelle 36. Db2-Statistikdatensatz (Q5ST) (Forts.)

Offset: Dez.	Offset: Hex	Typ	Len	Name	Beschreibung
36	24	Integer	4	DEADCNT	Anzahl der Deadlock-Zeitlimitüberschreitungen
40	28	Integer	4	DELECNT	Anzahl Löschanforderungen
44	2C	Integer	4	LISTCNT	Anzahl der Listenanforderungen
48	30	Integer	4	LESCNT	Anzahl der Leseanforderungen
52	34	Integer	4	UPDTCNT	Anzahl der Aktualisierungsanforderungen
56	38	Integer	4	SCHREIBEN	Anzahl der Schreibanforderungen
60	3C	Integer	4	SCSSEL	SCST (Shared-Channel-Status) wählt
64	40	Integer	4	SCSINS	SCST-Einfügungen
68	44	Integer	4	SCSUPD	SCST-Aktualisierungen
72	48	Integer	4	SCSDEL	SCST-Löschungen
76	4C	Integer	4	SSKSEL	SSKT (shared-sync-key) wählt
80	50	Integer	4	SSKINS	SSKT-Einfügungen
84	54	Integer	4	SSKDEL	SSKT-Löschungen
88	58	Integer	4	SCSBFTS	SCST-Anzahl der Puffer zu klein
92	5C	Integer	4	SCSMAXR	Maximale Anzahl der SCST-Zeilen bei Abfrage
96	60	Integer	4	* (2)	Reserved
104	68	Zeichen	8	DELETCUW	Kumulativer STCK-Unterschied-Thread-Löschen
112	70	Zeichen	8	DELETMXW	Maximale STCK-Differenz-Threadlösch
120	78	Zeichen	8	DELESCUW	Kumulativer STCK-Unterschied-SQL-Löschen
128	80	Zeichen	8	DELESMXW	Maximale STCK-Differenz-SQL-Löschen
136	88	Zeichen	8	LISTTCUW	Kumulative STCK-Differenz-Threadliste
144	90	Zeichen	8	LISTTMXW	Maximale STCK-Differenz-Threadliste
152	98	Zeichen	8	LISTSCUW	Kumulative STCK-Differenz-SQL-Liste
160	A0	Zeichen	8	LISTSMXW	Maximale STCK-Differenz-SQL-Liste
168	A8	Zeichen	8	READTCUW	Differenz der kumulativen STCK-Differenz-Thread-Lese-
176	B0	Zeichen	8	READTMXW	Maximale STCK-Differenz-Thread gelesen
184	B8	Zeichen	8	READSCUW	Kumulativer STCK-Unterschied-SQL-Lese-
192	C0	Zeichen	8	READSMXW	Maximale STCK-Differenz-SQL-Lese-
200	C8	Zeichen	8	UPDTTCUW	Kumulative STCK-Differenz-Threadaktualisierung
208	D0	Zeichen	8	UPDTTMXW	Maximale STCK-Differenz-Threadaktualisierung

Tabelle 36. Db2-Statistikdatensatz (Q5ST) (Forts.)

Offset: Dez.	Offset: Hex	Typ	Len	Name	Beschreibung
216	D8	Zeichen	8	UPDTSCUW	Kumulative STCK-Differenz-SQL-Aktualisierung
224	E0	Zeichen	8	UPDTSMXW	Maximale STCK-Differenz-SQL-Aktualisierung
232	E8	Zeichen	8	SCHREIBEN	Kumulativer STCK-Unterschied-Thread-Schreibzugriff
240	F0	Zeichen	8	WRITTMXW	Maximale STCK-Differenz-Thread-Schreibzugriff
248	F8	Zeichen	8	WRITSCUW	Kumulativer STCK-Unterschied-SQL-Schreibzugriff
256	100	Zeichen	8	WRITSMXW	Maximale STCK-Differenz-SQL-Schreibzugriff
264	108	Zeichen	8	SCSSTCUW	Differenz der kumulativen STCK-Differenz-Thread-Auswahl
272	110	Zeichen	8	SCSSTMXW	Maximale STCK-Differenz-Thread-Auswahl
280	118	Zeichen	8	SCSSSCUW	Kumulativer STCK-Unterschied-SQL-Auswahl
288	120	Zeichen	8	SCSSSMXW	Maximale STCK-Differenz-SQL-Auswahl
296	128	Zeichen	8	SCSITCUW	Kumulativer STCK-Unterschied-Threadeinfügung
304	130	Zeichen	8	SCSITMXW	Maximale STCK-Differenz-Threadeinfügung
312	138	Zeichen	8	SCSISCUW	Kumulativer STCK-Unterschied-SQL-Einfügung
320	140	Zeichen	8	SCSISMXW	Maximale STCK-Differenz-SQL-Einfügung
328	148	Zeichen	8	SCSUTCUW	Kumulative STCK-Differenz-Threadaktualisierung
336	150	Zeichen	8	SCSUTMXW	Maximale STCK-Differenz-Threadaktualisierung
344	158	Zeichen	8	SCSUSCUW	Kumulative STCK-Differenz-SQL-Aktualisierung
352	160	Zeichen	8	SCSUSMXW	Maximale STCK-Differenz-SQL-Aktualisierung
360	168	Zeichen	8	SCSDTCUW	Kumulativer STCK-Unterschied-Thread-Löschen
368	170	Zeichen	8	SCSDTMXW	Maximale STCK-Differenz-Threadlösch
376	178	Zeichen	8	SCSDSCUW	Kumulativer STCK-Unterschied-SQL-Löschen
384	180	Zeichen	8	SCSDSMXW	Maximale STCK-Differenz-SQL-Löschen
392	188	Zeichen	8	SSKSTCUW	Differenz der kumulativen STCK-Differenz-Thread-Auswahl
400	190	Zeichen	8	SSKSTMXW	Maximale STCK-Differenz-Thread-Auswahl
408	198	Zeichen	8	SSKSSCUW	Kumulativer STCK-Unterschied-SQL-Auswahl
416	1A0	Zeichen	8	SSKSSMXW	Maximale STCK-Differenz-SQL-Auswahl

Tabelle 36. Db2-Statistikdatensatz (Q5ST) (Forts.)

Offset: Dez.	Offset: Hex	Typ	Len	Name	Beschreibung
424	1A8	Zeichen	8	SSKITCUW	Kumulativer STCK-Unterschied-Threadeinfügung
432	1B0	Zeichen	8	SSKITMXW	Maximale STCK-Differenz-Threadeinfügung
440	1B8	Zeichen	8	SSKISCUW	Kumulativer STCK-Unterschied-SQL-Einfügung
448	1C0	Zeichen	8	SSKISMWX	Maximale STCK-Differenz-SQL-Einfügung
456	1C8	Zeichen	8	SSKDTCUW	Kumulativer STCK-Unterschied-Thread-Löschen
464	1D0	Zeichen	8	SSKDTMXW	Maximale STCK-Differenz-Threadlösch
472	1D8	Zeichen	8	SSKDSCUW	Kumulativer STCK-Unterschied-SQL-Löschen
480	1E0	Zeichen	8	SSKDSMXW	Maximale STCK-Differenz-SQL-Löschen
488	1E8	Integer	4	LMSSEL	Anzahl der Db2-BLOB-Leseanforderungen
492	1EC	Integer	4	LMSINS	Anzahl der Db2-BLOB-Einfügeanforderungen
496	1F0	Integer	4	LMSUPD	Anzahl der Db2-BLOB-Aktualisierungsanforderungen
500	1F4	Integer	4	LMSDEL	Anzahl der Db2-BLOB-Löschanforderungen
504	1F8	Integer	4	LMSLIS	Anzahl der Db2-BLOB-Auflistungsanforderungen
508	IFC	64-Bit-Inte	8	LMSSTCUW	Die gesamte abgelaufene Zeit für alle Thread-Lese-BLOB-Anforderungen
516	204	64-Bit-Inte	8	LMSSTMWX	Maximal abgelaufene Zeit für eine Thread-Lese-BLOB-Anforderung
524	20C	64-Bit-Inte	8	LMSSSCUW	Die gesamte abgelaufene Zeit für alle BLOB-Leseanforderungen für SQL.
532	214	64-Bit-Inte	8	LMSSSMWX	Maximal abgelaufene Zeit für eine SQL-Lese-BLOB-Anforderung
540	21C	64-Bit-Inte	8	LMSITCUW	Die gesamte abgelaufene Zeit für alle BLOB-Anforderungen des Threads.
548	224	64-Bit-Inte	8	LMSITMXW	Maximal abgelaufene Zeit für eine Thread-Insert-BLOB-Anforderung
556	22C	64-Bit-Inte	8	LMSISCUW	Die gesamte abgelaufene Zeit für alle BLOB-Anforderungen des Typs ' SQL einfügen
564	234	64-Bit-Inte	8	LMSISMWX	Maximal abgelaufene Zeit für eine BLOB-Anforderung für SQL-Einfügung
572	23C	64-Bit-Inte	8	LMSUTCUW	Die gesamte abgelaufene Zeit für alle BLOB-Anforderungen des Thread-Aktualisier
580	244	64-Bit-Inte	8	LMSUTMXW	Maximal abgelaufene Zeit für eine BLOB-Anforderung zum Aktualisieren von Threads

Tabelle 36. Db2-Statistikdatensatz (Q5ST) (Forts.)

Offset: Dez.	Offset: Hex	Typ	Len	Name	Beschreibung
588	24C	64-Bit-Inte	8	LMSUSCUW	Die gesamte abgelaufene Zeit für alle BLOB-Anforderungen für SQL-Aktualisierungen.
596	254	64-Bit-Inte	8	LMSUSMXW	Maximal abgelaufene Zeit für eine BLOB-Anforderung für SQL-Aktualisierung
604	25C	64-Bit-Inte	8	LMSDTCUW	Die gesamte abgelaufene Zeit für alle BLOB-Anforderungen zum Löschen von Threads
612	264	64-Bit-Inte	8	LMSDTMXW	Maximal abgelaufene Zeit für eine Threadlösch-BLOB-Anforderung
620	26C	64-Bit-Inte	8	LMSDSCUW	Die gesamte abgelaufene Zeit für alle BLOB-Anforderungen zum Löschen von SQL.
628	274	64-Bit-Inte	8	LMSDSMXW	Maximal abgelaufene Zeit für eine SQL-DELETE-BLOB-Anforderung
636	27C	64-Bit-Inte	8	LMSLTCUW	Die gesamte abgelaufene Zeit für alle BLOB-Anforderungen der Threadliste.
644	284	64-Bit-Inte	8	LMSLTMXW	Maximal abgelaufene Zeit für eine BLOB-Anforderung einer Threadliste
652	28C	64-Bit-Inte	8	LMSLSCUW	Die gesamte abgelaufene Zeit für alle BLOB-Anforderungen der SQL-Liste
660	294	64-Bit-Inte	8	LMSLSMXW	Maximal abgelaufene Zeit für eine BLOB-Anforderung der SQL-Liste

Die Daten enthalten Zähler für jeden Anforderungstyp, den der Db2-Ressourcenmanager unterstützt. Für diese Anforderungstypen werden die folgenden maximalen und kumulativen Antwortzeiten für die folgenden Anforderungen beibehalten:

- Die Zeit, die insgesamt im Db2-Ressourcenmanager verbraucht wurde (Thread-Zeit).
- Die Zeit, die für die Ausführung der RRSAF- und SQL-Teile der Anforderung aufgewendet wurde (eine Untergruppe der Thread-Zeit, die als SQL-Zeit bezeichnet wurde).

Informationen werden auch für folgende Informationen bereitgestellt:

- Die Anzahl der zugeordneten Server-Tasks.
- Die maximale Gesamtanforderungstiefe für jede der Servertasks.
- Die Anzahl der abnormalen Beendigung einer der Server-Task-Anforderungen.

Wenn die abnormale Beendigungszahl nicht null ist, wird eine requeue-Zahl angegeben, die die Anzahl der Anforderungen in der Warteschlange angibt, die infolge der abnormalen Beendigung in andere Server-Tasks eingereiht wurden.

Wenn die durchschnittliche Threadzeit wesentlich größer ist als die durchschnittliche SQL-Zeit, kann dies darauf hindeuten, dass Threadanforderungen übermäßig viel Zeit auf eine Servertask warten, um den SQL-Teil der Anforderung zu verarbeiten. Wenn dies der Fall ist, überprüfen Sie das Feld DHIGMAX. Wenn der Wert größer als eins ist, sollten Sie die im Parameter QSGDATA des Systemparametermakros CSQ6SYSP angegebene Anzahl der Db2-Server-Tasks erhöhen.

## Coupling Facility Manager-Datensätze

Verwenden Sie dieses Thema als Referenz auf das Format der Datensätze der Coupling Facility-Manager.

Das Format des Coupling Facility-Manager-Statistikdatensatzes wird in der folgenden Tabelle und in Assemblermakrothlqual beschrieben. SCSQMACS (CSQDQEST) und C-Headerdatei thlqual.SCSQC370 (CSQDSMFC). Die Feldnamen in C sind alle in Kleinbuchstaben angegeben, z. B. qest, qestid.

Wenn der Warteschlangenmanager nicht als Mitglied einer Gruppe mit gemeinsamer Warteschlange gestartet wurde, werden in diesem Datensatz keine Daten aufgezeichnet.

*Tabelle 37. Coupling Facility-Statistikdatensatz (QEST)*

Offset: Dez.	Offset: Hex	Typ	Len	Name	Beschreibung
0	0	Struktur	4104	QEST	CF-Manager-Statistik
0	0	Bitfolge	2	QESTID	Steuerblock-ID
2	2	Integer	2	QESTLL	Steuerblocklänge
4	4	Zeichen	4	QESTEYEC	Steuerungsblock-Blickfang
8	8	Zeichen	4096	QESTZERO	QEST-Teil bei Anlässen gelöscht
8	8	Zeichen	64	QESTSTUC (0:63)	Array (ein Eintrag pro Struktur)
8	8	Zeichen	12	QESTSTR	Name der Struktur
20	14	Integer	4	QESTSTRN	Strukturnummer
24	18	Integer	4	QESTCSEC	Anzahl der IXLLSTE-Aufrufe
28	1C	Integer	4	QESTCMEC	Anzahl der IXLLSTM-Aufrufe
32	20	Zeichen	8	QESTSSTC	Zeit für die Verwendung von IXLLSTE-Aufrufen
40	28	Zeichen	8	QESTMSTC	Zeit für die Verwendung von IXLLSTM-Aufrufen
48	30	Integer	4	QESTRSEC	Anzahl der IXLLSTE-Relaufwerke
52	34	Integer	4	QESTRMEC	Anzahl der IXLLSTM-Relaufwerke
56	38	Integer	4	QESTSFUL	Anzahl der Strukturen fulls
60	3C	Integer	4	QESTMNUS	Maximale Anzahl der Einträge im Gebrauch
64	40	Integer	4	QESTMLUS	Maximale Anzahl der zu verwendenden Elemente
68	44	Zeichen	4	*	Reserved
4104	1008	Zeichen	0	*	Ende des Steuerblocks

Die Daten enthalten Informationen für jede Coupling-Facility-Listenstruktur, einschließlich der Struktur CSQ\_ADMIN, die der WS-Manager während des Statistikintervalls verbinden konnte. Die Informationen für jede Struktur enthalten die folgenden Informationen:

- Die Anzahl der abgelaufenen und kumulativen abgelaufenen Zeiten für IXLLSTE- und IXLLSTM-Anforderungen.
- Die Häufigkeit, mit der eine Anforderung aufgrund einer Zeitlimitüberschreitung erneut versucht werden musste.

- Die Häufigkeit, mit der eine Bedingung 'Struktur voll' aufgetreten ist.

## **z/OS Themenmanagerdatensätze**

Verwenden Sie dieses Thema als Verweis auf das Format der Topic-Managerdatensätze.

Das Format des Topic-Manager-Statistikdatensatzes wird in der folgenden Tabelle und in Assemblermakro thlqual.SCSQMACS (CSQDQTST) und C-Headerdatei thlqual.SCSQC370 (CSQDSMFC) beschrieben. Die Feldnamen in C sind alle in Kleinbuchstaben angegeben, z. B. qtst, qtstid.

Offset: Dez.	Offset: Hex	Typ	Len	Name	Beschreibung
0	0	Struktur	96	QTST	Themenmanagerstatistik
0	0	Bitfolge	2	QTSTID	Steuerblock-ID
2	2	Integer	2	QTSTLL	Steuerblocklänge
4	4	Zeichen	4	TESTEYEC	Steuerungsblock-Blickfang
8	8	Zeichen	88	QTSTZERO	QTST-Teil bei Gelegenheit gelöscht
8	8	Integer	4	QTSTSTOT	Gesamtanzahl der Subskriptionsanforderungen
12	0C	Integer	4	QTSTSDUR	Permanente Subskriptionsanforderungen
16	10	Integer	4	QTSTSHIG (1: 3)	Array mit hoher Wasserzeichen (API, ADMIN, PROXY) für Subskriptionen
28	1C	Integer	4	QTSTSLOW (1: 3)	Untere Wasserzeichenbereich für Subskriptionen (API, ADMIN, PROXY)
40	28	Integer	4	QTSTSEXP	Abonnements abgelaufen
44	2C	Integer	4	QTSTTMSG	Summe der Nachrichten, die in die Unterwarteschlange gestellt
48	30	Integer	4	QTSTSPHW	Hochwasserzeichen für Einzelveröffentlichungsteilnehmer
52	34	Integer	4	QTSTPTOT (1: 3)	Gesamtanzahl der Veröffentlichungsanforderungen (API, ADMIN, PROXY)
64	40	Integer	4	QTSTPTHI	Gesamtveröffentlichungszeichen für Hochwasser
68	44	Integer	4	QTSTPTLO	Gesamtveröffentlichungswasserzeichen insgesamt
72	48	Integer	4	QTSTPNOS	Anzahl der Veröffentlichungen zu keinem Subskribenten
76	4C	Integer	4	*	Reserved

Offset: Dez.	Offset: Hex	Typ	Len	Name	Beschreibung
80	50	Bitfolge	8	QTSTETHW	Verfallzeit HW bei Veröffentlichung
88	58	Bitfolge	8	QTSTETTO	Gesamtsumme der Verlit-zungszeit bei Veröffentlichung

### z/OS **SMDS-Datendatensätze des Coupling Facility-Managers**

Verwenden Sie dieses Thema als Referenz auf das Format der SMDS-Datensätze (SMDS = Coupling Facility Manager-Gemeinsam genutzte Nachrichtendaten).

Das Format des SMDS-Statistikdatensatzes (SMDS = Coupling Facility Manager Shared Message Data Set) wird im Assemblermakro `th1qual.SCSQMACS(CSQDQESD)` und in der C-Headerdatei `th1qual.SCSQC370(CSQDSMFC)` beschrieben.

Die Statistikdaten stellen Informationen zur Auslastung der gemeinsamen Nachrichtengruppe, der E/A-Aktivität für die Gruppe der gemeinsam genutzten Nachrichtendatensätze und der SMDS-Pufferauslastung bereit.

Wenn der Warteschlangenmanager nicht als Mitglied einer Gruppe mit gemeinsamer Warteschlange gestartet wurde, werden in diesem Datensatz keine Daten aufgezeichnet.

### z/OS **Layout der Datensätze für den Kanalinitiator mit dem SMF-Typ 115**

In diesem Abschnitt wird das Layout von Datensätzen für Kanalinitiatorstatistikdaten (SMF-Typ 115, Subtype 231) beschrieben.

## Selbstdefinierendes Kapitel

Der selbstdefinierende Abschnitt für die Kanalinitiatorstatistik folgt dem SMF-Standardheader. Er ist im standardmäßigen Triplet-Format strukturiert. Das Format der Triplets wird in der Struktur `qwsx` in der Headerdatei `th1qual.SCSQC370(CSQDSMFC)` für die Programmiersprache C und im Assemblermakro `th1qual.SCSQMACS(CSQDQWSX)` beschrieben.

Unter [Tabelle 39 auf Seite 358](#) wird das Format des selbstdefinierenden Abschnitts gezeigt.

Offset: Dez.	Off- set: Hex	Typ	Länge	Name	Beschreibung
0	0	Integer	4	QWSX0PSO	Offset zwischen dem Start des SMF-Datensatzes und dem ersten Instrumentierungsstandardheader (QWHS)
4	4	Integer	2	QWSX0PSL	Länge von QWHS
6	6	Integer	2	QWSX0PSN	Anzahl der Instanzen von QWHS
8	8	Integer	4	QWSX0R1O	Offset zwischen dem Start des SMF-Datensatzes und dem ersten Block mit Steuerinformationen des Kanalinitiators (QCCT)
12	C	Integer	2	QWSX0R1L	Länge von QCCT
14	E	Integer	2	QWSX0R1N	Anzahl der Instanzen von QCCT

Tabelle 39. Struktur des selbstdefinierenden Abschnitts für die Kanalinitiatorstatistik (Forts.)

Offset: Dez.	Off- set: Hex	Typ	Länge	Name	Beschreibung
16	10	Integer	4	QWSX0R20	Offset zwischen dem Start des SMF-Datensatzes und dem ersten Dispatcher-Taskblock (QCT_DSP)
20	14	Integer	2	QWSX0R2L	Länge von QCT_DSP
22	16	Integer	2	QWSX0R2N	Anzahl der Instanzen von QCT_DSP
24	18	Integer	4	QWSX0R30	Offset zwischen dem Start des SMF-Datensatzes und dem ersten Adaptertaskblock (QCT_ADP)
28	1C	Integer	2	QWSX0R3L	Länge von QCT_ADP
30	1E	Integer	2	QWSX0R3N	Anzahl der Instanzen von QCT_ADP
32	20	Integer	4	QWSX0R40	Offset zwischen dem Start des SMF-Datensatzes und dem ersten SSL-Taskblock (QCT_SSL)
36	24	Integer	2	QWSX0R4L	Länge von QCT_SSL
38	26	Integer	2	QWSX0R4N	Anzahl der Instanzen von QCT_SSL
40	28	Integer	4	QWSX0R50	Offset zwischen dem Start des SMF-Datensatzes und dem ersten DNS-Taskblock (QCT_DNS)
44	2C	Integer	2	QWSX0R5L	Länge von QCT_DNS
46	2E	Integer	2	QWSX0R5N	Anzahl der Instanzen von QCT_DNS

In der Regel enthält ein Datensatz alle Daten. Wenn eine große Anzahl von Dispatchern, Adaptern oder SSL-Tasks vorhanden ist, werden die Daten auf mehrere Datensätze aufgeteilt.

In diesem Fall kann die Anzahl der Instanzen eines bestimmten Tasktyps null sein und Informationen zu einer Gruppe von Tasks können sich auf mehrere Datensätze verteilen. Der Block mit den Steuerinformationen zum Kanalinitiator (QCCT) ist nur im ersten Datensatz vorhanden. Die Daten können beispielsweise folgendermaßen auf zwei SMF-Datensätze aufgeteilt sein:

Tabelle 40. Beispieldaten

Zähler	Erster Datensatz	Letzter Datensatz
QWHS	1	1
QCCT	1	0
QCT_DSP	50	5
QCT_ADP	0	10
QCT_SSL	0	3
QCT_DNS	0	1

Dieses Beispiel zeigt, dass während des SMF-Intervalls 55 TCBs des Dispatchers vorhanden waren.

## Instrumentierungsstandardheader (QWHS)

Das Format von QWHS wird in der Struktur `qwhs` in der Headerdatei `thlqual.SCSQC370(CSQDSMFC)` für die Programmiersprache C und im Assemblermakro `thlqual.SCSQMACS(CSQDQWHS)` beschrieben. Darin sind die folgenden Schlüsselfelder enthalten, die für die Datensätze mit dem Kanalinitiator des SMF-Typs 115 relevant sind:

*Tabelle 41. Schlüsselfelder in QWHS*

Name	Länge	Beschreibung
QWHSNDA	1 Byte	Anzahl der selbstdefinierenden Abschnitte
QWHSSSID	4 Byte	Subsystemname
QWHS SMFC	1 Bit	Gibt an, ob mehrere SMF-Datensätze mit Informationen für dieses Intervall vorhanden sind. Wenn dieses Bit aktiviert ist, werden Informationen für dieses Intervall in weiteren SMF-Datensätzen fortgesetzt. Wenn dieses Bit nicht mehr vorhanden ist, ist dies der letzte oder einzige Datensatz. Mit der Subsystem-ID in QWHSSSID und der Startzeit des SMF-Intervalls in QWHSTIME können mehrere Datensätze für das gleiche Intervall gruppiert werden.
QWHSTIME	8 Byte	Ortszeit beim Start des Intervalls im STCK-Format
QWHS DURN	8 Byte	Dauer vom Start bis zum Ende des Intervalls im STCK-Format
QWHSSTCK	8 Byte	Ende des Intervalls in UTC im STCK-Format

### **Kanalinitiatorstatistikdatensätze**

Verwenden Sie dieses Thema als Referenz für Kanalinitiatorstatistikdatensätze.

Das Format des Kanalinitiatorstatistikdatensatzes enthält zwei Teile:

- Der erste Teil ist der Steuerinformationsblock des Kanalinitiators, der im Assemblermakro `thlqual.SCSQMACS(CSQDQCCT)` beschrieben wird. Weitere Informationen finden Sie unter „[Steuerinformationsblock für Kanalinitiator](#)“ auf Seite 361.
- Der zweite Teil ist der Taskblock des Kanalinitiators, der im Assemblermakro `thlqual.SCSQMACS(CSQDQCTA)` beschrieben wird.

Der Taskblock des Kanalinitiators enthält Informationen zu den vier Tasktypen innerhalb des CHINIT. Weitere Informationen finden Sie unter:

- „[Dispatcher-Tasks](#)“ auf Seite 362
- „[Adaptertasks](#)“ auf Seite 363
- „[Task "Domain Name Server \(DNS\)“](#)“ auf Seite 364
- „[SSL-Tasks](#)“ auf Seite 365

Zu jeder Task gehören:

- Die abgelaufene Zeit, die die Task im Intervall für die Verarbeitung von Anforderungen benötigt hat (*qctel tm*)
- Die von der Task im Intervall verbrauchte CPU-Zeit, die sich aus der CPU-Nutzung während der Verarbeitung von Anfragen und der CPU-Nutzung zwischen Anfragen zusammensetzt (*qctcptm*)
- Gesamtwartezeit dieser Task im Intervall (*qctwttm*)
- Die Anzahl der Anforderungen im Intervall (*qctreqn*)

Sie können diese Informationen verwenden, um zu sehen, wie ausgelastet die Task war, und ermitteln, ob Sie weitere Tasks auf der Basis der Analyse hinzufügen müssen.

Bei TLS- und DNS-Tasks sind die Dauer der längsten Anforderung (*qctlgdu*, *qctlsdu*) und die Uhrzeit, zu der diese (*qctlgdm*, *qctlsdm*) aufgetreten sind, ebenfalls enthalten.

Diese können nützlich sein, um zu ermitteln, wann Kanalanforderungen lange Zeit in Anspruch genommen haben. Beispielsweise wird eine DNS-Suchanforderung an einen Server außerhalb Ihres Unternehmens und nicht in Millisekunden, sondern in Sekunden ausgeführt.

Der Wert für die CPU-Zeit (*qctcptm*) umfasst die gesamte von der Task verbrauchte CPU-Zeit, sowohl bei der Bearbeitung von Anfragen als auch zwischen den Bearbeitungsanfragen. Der Wert für die abgelaufene Zeit (*qcteltm*) umfasst nur die Zeit während der Verarbeitung von Anfragen. Dies bedeutet, dass die CPU-Zeit größer sein kann als die abgelaufene Zeit.

Die Beispielabrechnungsdaten in den folgenden Tasks wurde mit dem IBM MQ-SupportPac MP1B formatiert.

Beide Teile werden auch in der Headerdatei `th1qua1.SCSQC370` (CSQDSMFC) für die Programmiersprache C beschrieben. Beachten Sie, dass die Feldnamen in C alle in Kleinbuchstaben angegeben sind, z. B. *qcct*, *qct\_adp*.

### Steuerinformationsblock für Kanalinitiator

Verwenden Sie dieses Thema als Referenz für den Steuerinformationsblock des Kanalinitiators.

Der Steuerinformationsblock des Kanalinitiators enthält Basisinformationen zu diesem CHINIT, einschließlich:

- CHINIT-Jobname (*qcctjobn*)
- QSG-Name, wenn er sich in einer Gruppe mit gemeinsamer Warteschlange (*qcctqsgn*) befindet
- Spitzenwert für die Anzahl der aktuellen Kanäle (*qcctnocc*)
- Spitzenwert für die Anzahl der aktiven Kanäle (*qcctnoac*)
- MAXCHL-Maximal zulässige Stromkanäle (*qcctmxcc*)
- ACTCHL-maximal zulässige aktive Kanäle (*qcctmxac*)
- TCPCHL-maximal zulässige TCP/IP-Kanäle (*qcctmxtpt*)
- LU62CHL-maximal zulässige LU62-Kanäle (*qcctmxlu*)
- Von CHINIT in der erweiterten privaten Region (*qcctstus*) verwendeter Speicher. Diese Informationen werden auch über die Nachricht CSQX004I im CHINIT-Jobprotokoll bereitgestellt.

Das Format des Steuerinformationsblocks des Kanalinitiators wird in der Struktur *qcct* in der Headerdatei `th1qua1.SCSQC370` (CSQDSMFC) für die Programmiersprache C und im Assemblermakro `th1qua1.SCSQMACS` (CSQDQCCS) beschrieben.

Sie können diese Informationen verwenden, um festzustellen, ob sich die Anzahl der aktiven Kanäle dem konfigurierten Maximalwert nähert. Beachten Sie, dass die Anzahl der aktuellen und aktiven Kanäle die Werte sind, wenn der Datensatz erstellt wurde. Zwischen den beiden Intervallen könnte es also mehr als die Anzahl der aktiven Kanäle geben.

## Kanalinformationen aus SMF-Daten

Im Folgenden finden Sie ein Beispiel für Kanalinformationen aus SMF-Daten:

```
MVCA,MQPV,2014/03/18,13:00:00,VRM:800,
From 2014/03/18,12:45:00.015222 to 2014/03/18,13:00:00.083630 duration 900.068408 seconds
Peak number used of current channels..... 1
Peak number used of active channels ..... 1
MAXCHL. Max allowed current channels.....9999
ACTCHL. Max allowed active channels.....9999
TCPCHL. Max allowed TCP/IP channels.....9999
LU62CHL. Max allowed LU62 channels..... 200
Storage used by Chinit..... 436MB
```

Sie können die Speichernutzung überwachen und prüfen, ob der Wert nach oben trendet. Wenn sich die insgesamt verwendete Speichermenge auf den gesamten verfügbaren Speicher nähert, wird möglicher-

weise kein Speicher mehr verfügbar sein, und möglicherweise können viele weitere Kanäle nicht mehr unterstützt werden.

Wenn die Anzahl der aktiven aktuellen Kanäle auf die maximale Anzahl Kanäle tendiert, müssen Sie unter Umständen die maximale Anzahl an Kanälen erhöhen.

## Dispatcher-Tasks

Dieser Abschnitt enthält Beispieldaten für die Statistik zu Dispatcher-Tasks und Informationen zum Interpretieren der Daten.

Das Format des Dispatcher-Taskblocks wird in der Struktur `qct_dsp` in der Headerdatei `th1qua1.SCSQC370(CSQDSMFC)` für die Programmiersprache C und im Assemblermakro `th1qua1.SCSQMACS(CSQDQCTA)` beschrieben.

## Beispieldaten

Task	Type	Requests	Busy %	CPU used, Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	DISP	26587	0.4	0.592463	0.1	22	127
1	DISP	26963	0.3	0.588092	0.1	22	112
2	DISP	864329	2.7	2.545668	0.3	3	28
3	DISP	26875	0.4	0.590825	0.1	22	120
4	DISP	26874	0.4	0.603285	0.1	22	123
Summ	DISP	971628	0.8	4.920332	0.1	5	38

Die Beispieldaten zeigen, dass es fünf Dispatcher gab. Ein Kanal ist beim Start einem Dispatcher zugeordnet. Der Kanalinitiator versucht, den Arbeitsaufwand beim Zuordnen eines Kanal zu einem Dispatcher auf alle Dispatcher zu verteilen. In diesem Beispiel wird gezeigt, dass ein Dispatcher mehr Anforderungen verarbeitet als andere Dispatcher. Dies ist normal, da einige Kanäle möglicherweise gestoppt werden, so dass der Dispatcher weniger Kanäle verarbeitet, und manche Kanäle können busier sein als andere.

- Es wurden 4,9 Sekunden CPU von den Dispatchern verwendet.
- Die durchschnittliche Anforderung belegte 5 Mikrosekunden CPU und benötigte 38 Mikrosekunden Zeit.
- Ein Dispatcher wird verwendet, um Daten über ein Kommunikationsnetz zu senden und zu empfangen, und dies ist in der Regel nicht von externen Ereignissen abhängig. Die durchschnittliche abgelaufene Zeit sollte daher nahe der durchschnittlichen CPU-Zeit liegen, die verwendet wird. Der Wert für die CPU-Zeit (*qctcptm*) umfasst die gesamte von der Task verbrauchte CPU-Zeit, sowohl bei der Verarbeitung von Anfragen als auch zwischen den Bearbeitungsanfragen.

Der Wert für die abgelaufene Zeit (*qcteltm*) umfasst nur die Zeit während der Verarbeitung von Anfragen. Dies bedeutet, dass die CPU-Zeit größer sein kann als die abgelaufene Zeit. Wenn der CHINIT aufgrund von CPU-Mangel verzögert wird, ist das Verhältnis zwischen der durchschnittlichen abgelaufenen Zeit und der durchschnittlichen CPU-Zeit viel größer, als wenn der CHINIT nicht wegen der CPU verzögert wird.

- Die durchschnittliche CPU pro Anforderung ist vom Nachrichtenverkehr abhängig. Große Nachrichten verwenden beispielsweise mehr CPU als kleinere Nachrichten.

Die Felder werden wie folgt berechnet:

- Dauer: `qwhs.qwhsdurn`
- Anforderungen: `qctreqn`
- % ausgelastet: `qcteltm` und Dauer
- Verwendte CPU: `qctcptm`
- CPU%: `qctcptm` und Dauer
- Durchschnittliche CPU: `qctcptm` und `qctreqn`
- Average ET: `qcteltm` und `qctreqn`

In der Regel sollte die Anzahl der Dispatcher die Anzahl der Prozessoren in der logischen Partition (LPAR) kleiner oder gleich der Anzahl der Prozessoren sein. Wenn Sie mehr Dispatcher als Prozessoren in der

LPAR haben, können sie für CPU-Ressourcen konkurrieren. Weitere Informationen zum Optimieren des Systems finden Sie unter [SupportPac MP16](#).

Kanäle haben eine Affinität zu einem Dispatcher, so dass Sie möglicherweise feststellen, dass einige Dispatcher viele weitere Anforderungen verarbeiten, als ein anderer Dispatcher.

Sie können den Befehl `ALTER QMGR CHIDISPS ()` verwenden, um die Anzahl der verwendeten Dispatcher zu ändern. Alle Änderungen werden beim nächsten Start des Kanalinitiators wirksam.

## **Adaptertasks**

Dieses Thema enthält Beispieldaten für die Statistik zu Adaptertasks und Informationen zum Interpretieren der Daten.

Das Format des Adaptertaskblocks wird in der Struktur `qct_adp` in der Headerdatei der Programmiersprache C `thlqual.SCSQC370(CSQDSMFC)` und im Assemblermakro `thlqual.SCSQMACS(CSQDQCTA)` beschrieben.

## Beispieldaten

Task	Type	Requests	Busy %	CPU used, Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	ADAP	470297	10.2	41.290670	4.6	88	194
1	ADAP	13907	0.6	1.589428	0.2	114	365
2	ADAP	2517	0.2	0.185325	0.0	74	746
3	ADAP	1095	0.1	0.085774	0.0	78	907
4	ADAP	535	0.1	0.040743	0.0	76	947
5	ADAP	220	0.0	0.016228	0.0	74	1175
6	ADAP	82	0.0	0.005521	0.0	67	1786
7	ADAP	80	0.0	0.004248	0.0	53	1160
Summ	ADAP	488733	1.4	43.217938	0.6	88	205

Die Felder werden wie folgt berechnet:

- Dauer: *qwhs.qwhsdurn*
- Anfragen: *qctreqn*
- Belegung %: *qcteltm* und Dauer
- Verwendete CPU: *qctcptm*
- CPU %: *qctcptm* und Dauer
- Durchschnittliche CPU: *qctcptm* und *qctreqn* durchschnittlich
- ET: *qcteltm* und *qctreqn*

Dieses Beispiel zeigt, dass acht Adaptertasks ausgeführt wurden.

### Adapternummer 0

- Die Mehrzahl der Anforderungen wurde verarbeitet (470297 von 488733).
- War belegt 10,2% des Intervalls
- Verwendete 41,3 Sekunden der CPU

### Insgesamt

Die durchschnittliche CPU pro Anforderung betrug 88 Mikrosekunden CPU und dauerte 205 Mikrosekunden.

Die Adapter verarbeiten IBM MQ-Anforderungen. Einige dieser Anforderungen können z. B. auf die Protokoll-E/A während einer Festschreibung warten, sodass die durchschnittliche abgelaufene Zeit pro Anforderung nur wenig Bedeutung hat.

Der Wert für die CPU-Zeit (*qctcptm*) umfasst die gesamte von der Task verbrauchte CPU-Zeit, sowohl bei der Verarbeitung von Anfragen als auch zwischen den Bearbeitungsanfragen. Der Wert für die abgelaufene Zeit (*qcteltm*) umfasst nur die Zeit während der Verarbeitung von Anfragen. Dies bedeutet, dass die CPU-Zeit größer sein kann als die abgelaufene Zeit.

Bei einer IBM MQ-Anforderung wird der erste freie Adapter verwendet.

- Wenn mindestens ein Adapter nur wenig genutzt (weniger als 1%) belegt ist, haben Sie genügend Adapter.
- Wenn mindestens ein Adapter nicht verwendet wurde, sind genügend Adapter definiert.
- Wenn alle Adapter verwendet wurden, müssen Sie möglicherweise mehr Adapter zuordnen.
- Wenn alle Adapter verwendet wurden und sie alle für den Großteil des Intervalls ausgelastet waren, müssen Sie mehr Adapter zuordnen.

Sie können den Befehl ALTER QMGR CHIADAPS () verwenden, um die Anzahl der verwendeten Adapter zu ändern. Alle Änderungen werden beim nächsten Start des Kanalinitiators wirksam.



**Achtung:** Wenn es zu viele Adapter gibt, die auf eine kleine Gruppe von Warteschlangen einwirken, kann es zu einer Konkurrenzsituation innerhalb des Warteschlangenmanagers kommen.

## Zugehörige Verweise

[ALTER QMGR](#)

### Task "Domain Name Server (DNS)"

Dieser Abschnitt enthält Beispieldaten für die DNS-Taskstatistik und Informationen zum Interpretieren der Daten.

Das Format des DNS-Taskblocks wird in der Struktur `qct_dns` in der Headerdatei `thlqual.SCSQC370 (CSQDSMFC)` für die Programmiersprache C und im Assemblermakro `thlqual.SCSQMACS (CSQDQCTA)` beschrieben.

## Beispieldaten

```
Task, Type, Requests, Busy %, CPU used, CPU %, "avg CPU", "avg ET", longest,
date,           time
      0,  DNS,    14002,    0.0, 0.122578, 0.0,      9,      11,      463, 2014/03/18,
12:56:33.987671
Summ,  DNS,    14002,    0.0, 0.122578, 0.0,      9,      11,      463, 2014/03/18,
12:56:33.987671
```

Der Kanalinitiator verwendet eine einzelne DNS-Task. Das Beispiel zeigt, dass die Task 14002 Anforderungen verarbeitet hat und die Anforderung durchschnittlich 9 Mikrosekunden CPU-Zeit benötigte und 11 Mikrosekunden der abgelaufenen Zeit benötigt hat.

Die längste DNS-Anforderung hat die abgelaufene Zeit von 463 Mikrosekunden übernommen, und dies ist um 12:56:33 Ortszeit aufgetreten.

Die Felder werden wie folgt berechnet:

- Dauer: `qwhs.qwhsdurn`
- Anfragen: `qctreqn`
- Belegung %: `qcteltm` und Dauer
- Verwendete CPU: `qctcptm`
- CPU %: `qctcptm` und Dauer
- Durchschnittliche CPU: `qctcptm` und `qctreqn` durchschnittlich
- Durchschnittliche ET: `qcteltm` und `qctreqn`
- Längste: `qctlgdu`
- Längste bei: `qctlgtm`

Die DNS-Task kann aus Ihrem Unternehmen gehen, um die IP-Adresse zu suchen, die einem Namen zugeordnet ist. Wenn die durchschnittliche abgelaufene Zeit erheblich mehr ist als die durchschnittliche CPU-Zeit, die Sie verwendet haben, können Sie möglicherweise einige lange Anforderungen haben.

Wenn der Wert für die längste Anforderungszeit nicht akzeptabel ist, sollten Sie mit Ihrem Netzteam zusammenarbeiten, um zu ermitteln, warum Sie lange Anforderungen haben. Es kann sein, dass Sie in Ihren Verbindungen einen ungültigen Namen haben.

Wenn die DNS-Task für 25% der Dauer ausgelastet ist, sollten Sie die Ursache weiter untersuchen.

Der Wert für die CPU-Zeit (*qctcptm*) umfasst die gesamte von der Task verbrauchte CPU-Zeit, sowohl bei der Verarbeitung von Anfragen als auch zwischen den Bearbeitungsanfragen. Der Wert für die abgelaufene Zeit (*qcteltm*) umfasst nur die Zeit während der Verarbeitung von Anfragen. Dies bedeutet, dass die CPU-Zeit größer sein kann als die abgelaufene Zeit.

**Anmerkung:** Es gibt Anforderungen an die DNS-Task, die keine DNS-Suchoperationen sind. Daher kann es sein, dass die Anzahl der Anforderungen größer als null ist-aber keine längsten Anforderungsinformationen.

## SSL-Tasks

Dieser Abschnitt enthält Beispieldaten für die SSL-Taskstatistik und Informationen zum Interpretieren der Daten.

Das Format des SSL-Taskblocks wird in der Struktur *qct\_ssl* in der Headerdatei `thlqual.SCSQC370 (CSQDSMFC)` für die Programmiersprache C und im Assemblermakro `thlqual.SCSQMACS (CSQDQCTA)` beschrieben.

### Beispieldaten

Task, date,	Type,	Requests, time	Busy %,	CPU used, Seconds,	CPU %,	"avg CPU", uSeconds,	"avg ET", uSeconds,	longest, uSeconds,	
0, 12:46:40.237697	SSL,	3112,	1.2,	0.248538,	0.3,	80,	362,	8864,	2014/03/18,
1, 12:46:18.938022	SSL,	3070,	1.2,	0.245433,	0.3,	80,	359,	4714,	2014/03/18,
2, 12:46:35.358145	SSL,	3170,	1.2,	0.255557,	0.3,	81,	362,	7273,	2014/03/18,
3, 12:46:44.514045	SSL,	3060,	1.2,	0.246542,	0.3,	81,	365,	13164,	2014/03/18,
4, 12:46:22.134123	SSL,	3120,	1.3,	0.251927,	0.3,	81,	373,	22438,	2014/03/18,
Summ, 12:46:22.134123	SSL,	15532,	1.2,	1.247998,	0.3,	80,	364,	22438,	2014/03/18,

Diese Beispieldaten zeigen, dass die durchschnittliche Anforderung 364 Mikrosekunden dauerte. Die längste Anforderung war für SSL Task 4, hat 22.438 Mikrosekunden und ist um 12 :46:22.134123 Ortszeit aufgetreten.

Die Felder werden wie folgt berechnet:

- Dauer: *qwhs.qwhsdurn*
- Anfragen: *qctreqn*
- Belegung %: *qcteltm* und Dauer
- Verwendete CPU: *qctcptm*
- CPU %: *qctcptm* und Dauer
- Durchschnittliche CPU: *qctcptm* und *qctreqn* durchschnittlich
- Durchschnittliche ET: *qcteltm* und *qctreqn*
- Längste: *qctlsdu* längste bei: *qctlstm*

Ein Laufkanal ist einer SSL-Task zugeordnet, ähnlich wie ein Kanal einem Dispatcher zugeordnet ist. Die SSL-Tasks können die Verschlüsselungskoprozessoren verwenden, die der logischen Partition zur Verfügung stehen. Die abgelaufene Zeit kann also die Zeit umfassen, die für einen Koprozessor aufgewendet wurde. Sie sollten die durchschnittliche abgelaufene Zeit während des gesamten Tages überwachen. Wenn diese Zeit während der Spitzenauslastung erheblich zunimmt, sollten Sie sich an die zuständigen z/OS-Systemprogrammierer wenden, da Ihre Koprozessoren möglicherweise überlastet sind.

Wenn die SSL-Tasks für einen signifikanten Teil des Intervalls ausgelastet sind, kann die Anzahl der SSL-Tasks, die die Anzahl der SSL-Tasks erhöhen, erhöht werden. Wenn die SSL-Tasks auf externe Ressourcen wie einen Koprozessor warten, hat das Erhöhen der Anzahl der SSL-Tasks nur geringe Auswirkungen.

Sie können den Befehl `ALTER QMGR SSLTASKS ()` verwenden, um die Anzahl der verwendeten SSL-Tasks zu ändern. Alle Änderungen werden beim nächsten Start des Kanalinitiators wirksam.

Der Wert für die CPU-Zeit (*qctcptom*) umfasst die gesamte von der Task verbrauchte CPU-Zeit, sowohl bei der Verarbeitung von Anfragen als auch zwischen den Bearbeitungsanfragen. Der Wert für die abgelaufene Zeit (*qcteltm*) umfasst nur die Zeit während der Verarbeitung von Anfragen. Dies bedeutet, dass die CPU-Zeit größer sein kann als die abgelaufene Zeit.

### Zugehörige Verweise

[ALTER QMGR](#)

## **IBM MQ-Abrechnungsdaten interpretieren**

IBM MQ-Abrechnungsdaten werden in SMF-Datensätze des Typs 116 geschrieben. Verwenden Sie dieses Thema als Verweis auf die verschiedenen Typen von Abrechnungsdatensätzen.

IBM MQ-Abrechnungsdaten können für die folgenden Subtypen erfasst werden:

**0**

Abrechnungsdatensätze des Nachrichtenmanagers (wie viel Prozessorzeit wurde für die Verarbeitung von IBM MQ-API-Aufrufen verbraucht und die Anzahl der MQPUT- und MQGET-Aufrufe). Diese Informationen werden erzeugt, wenn eine angegebene Task die Verbindung zu IBM MQ trennt. Die Informationen in diesen Datensatz können also einen Zeitraum von mehreren Stunden umfassen.

**1**

Abrechnungsdaten für jede Task, auf Thread- und Warteschlangenebene.

**2**

Zusätzliche Abrechnungsdaten auf Warteschlangenebene (wenn die Task mehr Warteschlangen verwendet hat, als in den Subtyp 1-Datensatz passen könnte).

**10**

Abrechnungsdaten für Kanäle.

**Anmerkung:** Abrechnungsdaten für bestimmte Kanäle können durch das Kanalattribut `STATCHL` und das Attribut `STATACLS`-Warteschlangenmanager aktiviert oder unterdrückt werden.

Dabei ist Folgendes zu beachten:

- Subtyp 0 wird mit Traceklasse erstellt (1)
- Die Subtypen 1 und 2 werden mit der Traceklasse (3) erstellt.
- Subtyp 10 wird mit Accounting-Trace-Klasse (4) erstellt

Untertyp

## **Layout eines SMF-Datensatzes des Typs 116**

Verwenden Sie dieses Thema als Verweis auf das Format eines SMF-Schriftartdatensatzes.

Das Standardlayout für SMF-Datensätze umfasst drei Teile:

### **SMF-Header**

Stellt das Format, die Identifikation und die Uhrzeit- und Datumsinformationen zum Datensatz selbst bereit.

### **Selbstdefinierendes Kapitel**

Definiert die Position und Größe der einzelnen Datensätze innerhalb des SMF-Datensatzes.

### **Datensätze**

Die tatsächlichen Daten von IBM MQ, die Sie analysieren möchten.

Weitere Informationen zu SMF-Satzformaten finden Sie unter [z/OS MVS System Management Facilities \(SMF\)](#).

## SMF-Header

Tabelle 42 auf Seite 367 zeigt das Format des SMF-Datensatzheaders (SM116).

Tabelle 42. Beschreibung des SMF-Satzkennsatzes						
Off-set: Dez.	Offset: Hex	Art	Len	Ihren Na-men	Beschreibung	Beispiel
0	0	Struktur	28	SM116	SMF-Datensatz-Header.	
0	0	Integer	2	SM116LEN	SMF-Satzlänge.	01A4
2	2		2		Reserviert.	
4	4	Integer	1	SM116FLG	Systemanzeiger.	5E
5	5	Integer	1	SM116RTY	Satztyp. Der SMF-Datensatztyp für IBM MQ-Abrechnungsdatensätze lautet immer 116 (X'74').	74
6	6	Integer	4	SM116TME	Zeit, wenn ein SMF-Datensatz verschoben wurde.	00356124
10	A	Integer	4	SM116DTE	Datum, an dem der SMF-Datensatz verschoben wurde	0100223F
14	E	Zeichen	4	SM116SID	z/OS-Subsystem-ID. Definiert das z/OS-Subsystem, auf dem die Datensätze erfasst wurden.	D4E5F4F1 (MV41)
18	12	Zeichen	4	SM116SSI	IBM MQ-Subsystem-ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM116STF	Satzsubtyp.	0000
24	18	Zeichen	3	SM116REL	IBM MQ-Version.	F9F1F0 (910)
27	1B		1		Reserviert.	
28	1C	Zeichen	0	SM116END	Ende des SMF-Headers und Beginn des selbstdefinierenden Abschnitts.	

**Anmerkung:** Die (hexadezimalen) Werte in der rechten Spalte beziehen sich auf [Abbildung 22 auf Seite 369](#).

### Selbstdefinierende Abschnitte

Ein selbstdefinierter Abschnitt eines SMF-Satzes gibt Auskunft darüber, wo ein Abrechnungsdatensatz zu finden ist, wie lange er ist und wie oft dieser Datensatztyp wiederholt wird (mit unterschiedlichen Werten). Die sich selbst definierenden Abschnitte folgen dem Header in einem festen Offset vom Anfang des SMF-Datensatzes.

Jeder selbstdefinierende Abschnitt verweist auf rechnungsbezogene Daten. In [Tabelle 43 auf Seite 368](#) sind die Offsets vom Anfang des SMF-Datensatzheaders zusammengefasst.

Tabelle 43. Offsets für selbstdefinierende Abschnitte

Satzsubtyp (SMF116STF)	Quelle der Abrechnungs- daten	Offset des selbstdefinierenden Schnittabschnitts		Siehe ...
		Dez	Hex	
Alle	Allgemeiner Header	28	X'1C '	„Allgemeiner IBM MQ- SMF-Header“ auf Sei- te 370
0	Nachrichtenmanager	44	X'2C '	„Nachrichtenmanager- datensätze“ auf Seite 372
1	Thread-ID-Datensatz	36	X'24 '	„Datensätze auf Thread-Ebene und auf Warteschlangen- ebene“ auf Seite 373
1	Thread-Level-Abrechnung	44	X'2C '	„Datensätze auf Thread-Ebene und auf Warteschlangen- ebene“ auf Seite 373
1	Abrechnung auf Warte- schlangenebene	52	X'34 '	„Datensätze auf Thread-Ebene und auf Warteschlangen- ebene“ auf Seite 373. Dieser Abschnitt ist nur vorhanden, wenn das Feld WTASWQCT in der taskbezogenen Informationsstruktur (WTAS) ungleich null ist.
2	Thread-ID-Datensatz	36	X'24 '	„Datensätze auf Thread-Ebene und auf Warteschlangen- ebene“ auf Seite 373
2	Abrechnung auf Warte- schlangenebene	44	X'2C '	„Datensätze auf Thread-Ebene und auf Warteschlangen- ebene“ auf Seite 373
10	Kanalabrechnung			„Datensätze für Ka- nalabrechnungsdaten“ auf Seite 376

**Anmerkung:** Andere selbstdefinierende Abschnitte beziehen sich nur auf Daten für die Verwendung von IBM.

Jeder selbstdefinierende Abschnitt ist zwei Vollwörter lang und hat das folgende Format:

```
sssssssl1111nnnn
```

Dabei gilt:

**SSSSSSS**

Vollwort, das den Offset vom Start des SMF-Datensatzes enthält.

||||

Halfword, das die Länge dieses Datensatzes enthält.

nnnn

Halfword gibt die Anzahl der Datensätze in diesem SMF-Datensatz an.

Abbildung 22 auf Seite 369 zeigt ein Beispiel für einen Teil eines SMF-Datensatzes des Typs 116. Die Zahlen in der linken Spalte stellen die relative Position im Hexadezimalformat ab dem Anfang des Datensatzes dar. Jede Zeile entspricht 16 Byte an Daten, wobei jedes Byte zwei Hexadezimalzeichen ist, z. B. 0C. Die Zeichen in der rechten Spalte stellen die druckbaren Zeichen für jedes Byte dar. Nicht druckbare Zeichen werden durch einen Punkt (.) angezeigt.

In diesem Beispiel sind alternative Felder im SMF-Header unterstrichen, damit sie leichter erkennbar sind; Tabelle 42 auf Seite 367 hilft bei ihrer Identifizierung. Der selbstdefinierende Abschnitt für einen der Abrechnungsdatensätze des Nachrichtenmanagers (an der relativen Position in Tabelle 43 auf Seite 368) ist in **Fettdruck** dargestellt.

```
000000 01A40000 5E740035 61240100 223FD4E5 *...;.../....MV*
000000 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600....*
000000 00700001 00000054 00B00001 00000104 *.....*
000000 00300001 00000000 00000000 00000000 *.....*
000000 00000000 00000000 00000000 00000000 *.....*
```

Abbildung 22. Teil eines SMF-Satzes 116 mit Header- und selbstdefinierenden Abschnitten

Der selbstdefinierende Abschnitt für den Typ der Nachrichten-Manager-Abrechnungsdaten befindet sich bei der relativen Position X'2C ' vom Anfang des SMF-Datensatzes und enthält die folgenden Informationen:

- Die relative Position der Abrechnungsdaten des Nachrichtenmanagers liegt bei X'00000104 ' Byte vom Anfang des SMF-Datensatzes.
- Dieser Nachrichtenmanagerdatensatz ist X'0030 ' Byte lang.
- Es gibt einen Datensatz (X'0001 ').

**Anmerkung:** Verwenden Sie in den selbstdefinierenden Abschnitten immer Offsets, um die Abrechnungsdatensätze zu lokalisieren.

## Typ 116 SMF-Datensätze verarbeiten

Verwenden Sie dieses Thema als Referenz auf das Format des Abrechnungssatzes des Verarbeitungstyps.

Alle Abrechnungsdaten, die Sie von SMF erfassen, müssen verarbeitet werden, um nützliche Informationen zu extrahieren. Überprüfen Sie beim Verarbeiten der Daten, dass die Datensätze aus IBM MQ stammen und es sich dabei um die von Ihnen erwarteten Datensätze handelt.

Validieren Sie den Wert der folgenden Felder:

- SM116RTY, SMF-Satznummer = X'74 ' (116)
- SM116STF, der Satzsubtyp, muss 0000, 0001, 0002 oder 0010 sein.

Das Lesen aus den aktiven SMF-Dateigruppen (oder SMF-Protokolldatenströmen) wird nicht unterstützt. Sie müssen das SMF-Programm IFASMFDP (bzw. bei Verwendung von Protokolldatenströmen das SMF-Programm IFASMF DL) verwenden, um einen Speicherauszug von SMF-Datensätzen in einer sequenziellen Datei zu erstellen, damit sie verarbeitet werden können. Weitere Informationen finden Sie im Abschnitt „Systemverwaltungsfunktion verwenden“ auf Seite 335.

Einzelheiten zu den Strukturen und Feldern finden Sie im IBM MQ-SupportPac [MP1B](#).

Es gibt ein C-Beispielprogramm namens CSQ4SMFD, das den Inhalt von SMF-Datensätzen des Typs 115 und 116 aus der sequenziellen Datei ausgibt. Das Programm wird als Quelle in thlqual.SCSQC37S und im ausführbaren Format in thlqual.SCSQLOAD. bereitgestellt. Die Beispiel-JCL wird in thlqual.SCSQPROC (CSQ4SMFJ) bereitgestellt.

Sie müssen die SMFIN-DD-Karte mit dem Namen der SMF-Datei aktualisieren. Zeigen Sie mit dem z/OS-Befehl '/D SMF' den Namen des Datasets an und aktualisieren Sie die DUMPOUT-DD-Karte mit dem Namen des Ausgabedatensatzes.

Außerdem müssen Sie die erforderlichen START- und END-Zeiten angeben.

Das folgende JCL-Beispiel extrahiert SMF-Datensätze aus SMF-Dateien und erstellt einen Speicherauszug dieser Datensätze in der Datei SMFOUT:

```
//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=
//SMFIN DD DSN=xxxxxx.MANA,DISP=SHR
//SMFOUT DD DSN=xxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
INDD(SMFIN,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

Das folgende JCL-Beispiel extrahiert SMF-Datensätze aus dem in LSNAME benannten SMF-Protokoll-Datastrom und erstellt einen Speicherauszug dieser Datensätze in der Datei SMFOUT:

```
//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=*
//SMFOUT DD DSN=xxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
LSNAME(IFASMF.MQ,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

### Allgemeiner IBM MQ-SMF-Header

Verwenden Sie diesen Abschnitt als Verweis auf den allgemeinen Abrechnungsdatensatz für den SMF-Headertyp von IBM MQ.

Das Format dieses Datensatzes wird in Tabelle 44 auf Seite 370 in den Assemblermakros thlqual.SCSQMACS (CSQDQWHS) und thlqual.SCSQMACS (CSQDQWHC) und in der C-Headerdatei thlqual.SCSQC370 (CSQDSMFC) beschrieben. Die Feldnamen in C sind alle in Kleinbuchstaben, z. B. qwhs, qwhsnsda.

Die QWHS-Daten enthalten den Namen des Subsystems. Bei Subtyp 1-Datensätzen wird auch angezeigt, ob die Abrechnungssätze auf Warteschlangenebene vorhanden sind. Wenn das QWHSNSDA-Feld 3 oder weniger ist, gibt es nicht, und der entsprechende selbstdefinierende Abschnitt (bei der relativen Position X'34 ') ist nicht festgelegt.

Die QWHC-Daten enthalten Informationen über den Benutzer (z. B. die Benutzer-ID (QWHCAID) und den Typ der Anwendung (QWHCATYP)). Der Abschnitt QWHC wird nur für Datensätze des Typs 0 ausgeführt. Die entsprechenden Informationen sind im Thread-Identifikationsdatensatz für die Datensätze des Subtyps 1 und 2 enthalten.

Tabelle 44. Struktur des gemeinsamen IBM MQ-SMF-Kopfsatzes QWHS					
Offset: Dez.	Offset: Hex	Art	Länge	Ihren Namen	Beschreibung
0	0	Struktur	128	QWHS	
0	0		6		Reserved
6	6	Zeichen	1	QWHSNSDA	Anzahl der selbstdefinierenden Abschnitte in den SMF-Datensätzen
7	7		5		Reserved

Tabelle 44. Struktur des gemeinsamen IBM MQ-SMF-Kopfsatzes QWHS (Forts.)

Offset: Dez.	Off- set: Hex	Art	Länge	Ihren Namen	Beschreibung
12	C	Zeichen	4	QWHSSSID	Subsystemname
16	10		24		Reserved
40	28	Zeichen	8	QWHCAID	Benutzer-ID, die dem z/OS-Job zugeordnet ist
48	30	Zeichen	12	QWHCCV	Threadquerverweis
60	3C	Zeichen	8	QWHCCN	Verbindungsname
68	44		8		Reserved
76	4C	Zeichen	8	QWHCOPID	Benutzer-ID, die der Transaktion zugeordnet ist
84	54	Integer	4	QWHCATYP	Typ des Verbindungssystems (1 = CICS, 2 = Batch oder TSO, 3 = IMS-Steuerregion, 4 = IMS-MPP oder -BMP, 5 = Befehlsserver, 6 = Kanalinitiator, 7 = RRS-Batch)
88	58	Zeichen	22	QWHCTOKN	Abrechnungstoken, das auf die z/OS-Abrechnungsdaten für den Benutzer gesetzt ist
110	6E	Zeichen	16	QWHCNID	Netzwerk-ID
126	7E		2		Reserved

### **Leistungsdaten von CICS und IBM MQ kombinieren**

Dieser Abschnitt enthält Referenzinformationen zur Kombination von Leistungsdaten aus IBM MQ und CICS.

Der allgemeine Abschnitt QWHCTOKN des Abrechnungsdatensatzes für den IBM MQ-SMF-Headertyp wird zur Korrelation von SMF-Datensätzen mit dem CICS-Typ 110 mit SMF-Datensätzen mit dem IBM MQ-Typ 116 verwendet.

CICS generiert für jede CICS-Task ein Token für eine Arbeitseinheit von LU6.2. Das Token wird verwendet, um ein Abrechnungstoken zu generieren, das in QWHCTOKN in der Korrelationsheader von Subtyp-Null-Datensätzen geschrieben wird.

Die Details werden auch in den Abschnitt WTIDACCT in den Sätzen des Subtyps 1 und 2 geschrieben. Das Abrechnungstoken ermöglicht eine Korrelation zwischen den Leistungsdaten von CICS und IBM MQ für eine Transaktion.

### **Threadquerverweisdaten**

Verwenden Sie dieses Thema als Referenz auf das Format des Abrechnungssatzes für den Querverweistyp des Threads.

Die Interpretation der Daten im Feld 'Querverweis' (QWHCCV) des Threads ist unterschiedlich. Dies hängt davon ab, wie sich die Daten beziehen:

- CICS-Verbindungen (QWHCATYP=1) - siehe [Tabelle 45 auf Seite 372](#)
- IMS-Verbindungen (QWHCATYP=3 oder 4) - siehe [Tabelle 46 auf Seite 372](#)
- Stapelverbindungen (QWHCATYP= 2 oder 7)-Dieses Feld besteht aus binären Nullen
- Sonstige-keine aussagekräftigen Daten

Offset: Dez.	Offset: Hex	Typ	Länge	Beschreibung
48	30	Zeichen	4	CICS-Threadnummer
52	34	Zeichen	4	CICS-Transaktionsname.
56	38	Integer	4	CICS-Tasknummer.

Einige Einträge enthalten Leerzeichen. Diese beziehen sich auf die Task und nicht auf eine bestimmte Transaktion.

Offset: Dez.	Offset: Hex	Typ	Länge	Beschreibung
48	30	Zeichen	4	Regions-ID der IMS-Partitionsspezifikationsstabelle (PST)
52	34	Zeichen	8	Name des IMS-Programmspezifikationsblocks (PSB).

### Nachrichtenmanagerdatensätze

Verwenden Sie dieses Thema als Referenz auf das Format der Accounting-Datensätze des Nachrichtenmanagers.

Der Nachrichtenmanager ist eine Komponente von IBM MQ, mit der alle API-Anforderungen verarbeitet werden. Das Format der Accounting-Datensätze des Nachrichtenmanagers wird im Assemblermakro `thlqual.SCSQMACS (CSQDQMAC)` beschrieben.

Die QMAC-Daten geben Auskunft über die Prozessorzeit, die für die Verarbeitung von IBM MQ-Aufrufen benötigt wird, sowie die Anzahl der MQPUT- und MQGET-Anforderungen für Nachrichten mit unterschiedlicher Größe.

**Anmerkung:** Eine einzelne IMS-Anwendung kann zwei SMF-Datensätze schreiben. Addieren Sie in diesem Fall die Zahlen aus beiden Datensätzen, um die korrekte Summe für die IMS-Anwendung bereitzustellen

### Sätze mit Nullprozessorzeit

Es werden zuweilen Datensätze erstellt, die im Feld QMACCPUT keine Prozessorzeit enthalten. Diese Datensätze treten auf, wenn die Ausführung von Tasks mit langer Laufzeit für IBM MQ entweder beendet ist oder zur Ausgabe von Abrechnungsdatensätzen aufgefordert wird, indem der Abrechnungstrace gestoppt wird. Solche Tasks sind im CICS-Adapter und im Kanalinitiator vorhanden (für verteilte Steuerung von Warteschlangen). Die Anzahl dieser Tasks mit einer Prozessorzeit von null hängt davon ab, wie viel Aktivität im System vorhanden ist:

- Für den CICS-Adapter kann dies zu bis zu neun Datensätzen mit einer Prozessorzeit von null führen.
- Für den Kanalinitiator kann die Anzahl der Datensätze mit der Prozessorzeit null bis zu der Summe von `Adapters + Dispatchers + 6`, wie in den WS-Manager-Attributen definiert, sein.

Diese Datensätze spiegeln den Umfang der Arbeit wider, die unter der Task ausgeführt wurde, und kann ignoriert werden.

### Beispielabrechnungssatz des Subtyps 'Null'

Verwenden Sie dieses Thema als Verweis auf das Format der Abrechnungssätze des Typs 0, der untergeordnete Typen enthält.

Abbildung 23 auf Seite 373 zeigt einen SMF-Datensatz vom Typ 116, Subtyp Null. In dieser Abbildung sind der SMF-Datensatz-Header und der QMAC-Abrechnungsdatensatz unterstrichen. Die selbstdefinierenden Abschnitte sind in Fettschrift.

```

000000 01A40000 5E740035 61240100 223FD4E5 *...;.../.....MV*
000010 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600....*
000020 00700001 00000054 00B00001 00000104 *.....*
000030 00300001 00000000 00000000 00000000 *.....*
000040 00000000 00000000 00000000 00000000 *.....*
000050 00000000 B478AB43 9C6C2280 B478AB47 *.....%.....*
000060 9DB47E02 00000000 04C0F631 00000001 *.=.....}6.....*
000070 9880E72D 00000000 014D9540 00000000 *..X.....(. ....*
000080 08480C80 00000010 40404040 40404040 *.....*
000090 00000000 00000000 00000051 00000000 *.....*
0000A0 00000000 00000000 00000000 00000000 *.....*
0000B0 00000000 00000000 00000000 00000000 *.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 00000000 00000000 00000000 *.....*
0000E0 00000000 00000000 00000000 00000000 *.....*
0000F0 00000000 00000000 00000000 00000000 *.....*
000100 00000000 D4140030 D8D4C1C3 00000000 *...M...QMAC...*
000110 689C738D 00000050 00000000 00000050 *.....&.....&*
000120 0000000A 00000000 00000000 00000000 *.....*
000130 00000000 0024011A 00030710 02DAACF0 *.....0*

```

Abbildung 23. Beispiel für SMF-Typ 116, Subtyp Null

### z/OS Datensätze auf Thread-Ebene und auf Warteschlangenebene

Verwenden Sie dieses Thema als Referenz auf das Format der Abrechnungssätze auf Thread-Ebene und auf Warteschlangenebene.

Abrechnungssätze auf Threadebene werden mithilfe von IBM MQ für jede Task erfasst. Für jede Task wird ein Abrechnungsdatensatz auf Thread-Ebene in den SMF geschrieben, wenn die Task abgeschlossen ist. Bei einer Task mit langer Laufzeit werden auch Daten in das Statistikintervall geschrieben, das durch den Parameter STATIME des Systemparametermakros CSQ6SYSP (oder durch das System SMF-Statistik-Broadcast) festgelegt wurde, sofern die Task die vorherigen statistischen Daten erfasst hat. Darüber hinaus werden Abrechnungsdaten zu jeder Warteschlange erfasst, die die Task öffnet. Ein Abrechnungssatz auf Warteschlangenebene wird für jede Warteschlange geschrieben, die die Task seit dem letzten Schreiben auf Thread-Level-Abrechnungssatz verwendet hat.

Abrechnungssätze auf Thread- und Warteschlangenebene werden erstellt, wenn Sie die Klasse 3 angeben, wenn Sie den Accounting-Trace starten. Verwenden Sie beispielsweise folgenden Befehl:

```
START TRACE(ACCTG) DEST(SMF) CLASS(3)
```

Die Abrechnungsdaten auf Threadebene werden in einen SMF-Typ 116, Subtyp 1, geschrieben und folgen auf die Datensätze auf Warteschlangenebene. Wenn die Task viele Warteschlangen geöffnet hat, werden weitere Warteschlangeninformationen in einen oder mehrere SMF-Datensätze des Typs 116 des Subtyps 2 geschrieben. Ein Thread-ID-Steuerblock ist in jedem Datensatz des Subtyps 1 und 2 enthalten, damit Sie die einzelnen Datensätze mit der richtigen Task verknüpfen können. Gewöhnlich beträgt die maximale Anzahl der Datensätze auf Warteschlangenebene in jedem SMF-Datensatz etwa 45.

Das Format des Abrechnungssatzes auf Threadebene wird im Assemblermakro thlqual.SCSQMACS (CSQDWTAS) beschrieben. Das Format des Abrechnungssatzes auf Warteschlangenebene wird im Assemblermakro thlqual.SCSQMACS (CSQDWQ) beschrieben. Das Format des Thread-ID-Datensatzes wird im Assemblermakro thlqual.SCSQMACS (CSQDWTID) beschrieben. Alle diese Datensätze werden auch in der C-Headerdatei thlqual.SCSQC370 (CSQDSMFC) beschrieben. Die Feldnamen in C sind alle in Kleinbuchstaben angegeben, z. B. 'wtas', 'wtasshex'.

### z/OS Bedeutung der Kanalnamen

Verwenden Sie dieses Thema als Verweis auf die Bedeutung von Kanalnamen.

Der Kanalname in der WTID wird wie im folgenden Beispiel gezeigt erstellt. In diesem Beispiel ist ein Senderkanal vom WS-Manager QM1 zum Warteschlangenmanager QM2 vorhanden.

Die Bedeutung von Kanalnamen wird in der folgenden Tabelle beschrieben.

Tabelle 47. Bedeutung von Kanalnamen		
Feldname	Bedeutung	Beispiel
Für WS-Manager QM1 verfügt der Senderkanal über die folgenden Felder:		
WTIDCCN	Der Jobname.	QM1CHIN
WTIDCHL	Der Kanalname	QM1.QM2
WTIDCHLC	Dies wird im CONNAME des Kanals definiert.	WINMVS2B(2162)
Für WS-Manager QM2 verfügt der Empfängerkanal über die folgenden Felder:		
WTIDCCN	Der Jobname.	QM2CHIN
WTIDCHL	Der Kanalname	QM1.QM2
WTIDCHLC	Wo der Kanal herkam	9.20.101.14

**z/OS** *Beispieldatensätze für den Subtyp 1 und den Subtyp 2*

Verwenden Sie dieses Thema als Referenz auf das Format der Abrechnungssätze des Subtyps 1 und des Subtyps 2.

Abbildung 24 auf Seite 374 und Abbildung 25 auf Seite 375 zeigen Beispiele für SMF-Datensätze vom Typ 116, Subtyp 1 und Subtyp 2. Diese beiden Abrechnungssätze wurden für einen Stapeljob erstellt, der 80 Warteschlangen geöffnet hat. Da viele Warteschlangen geöffnet wurden, war ein Subtyp 2-Datensatz erforderlich, der alle erzeugten Informationen enthält.

```

000000  703C0000 5E74002D 983B0100 229FD4E5  *...;.....MV*
000010  F4F1D4D8 F0F70001 F6F0F000 00006FCC  *41MQ07..600..?.*
000020  00700001 0000003C 00000001 0000010C  *.....}*
000030  02C00001 000003CC 02400030 F70000D0  *..{.....7..}*
000040  E6E3C9C4 00000000 00000000 00000040  *WTID..... *
.
.
000100  00000000 00000000 7F4A4BB8 F70102C0  *....."...7..{*
000110  E6E3C1E2 B4802373 0BF07885 7F4AE718  *WTAS.....0..".X.*

```

Abbildung 24. Beispiel für SMF-Typ 116, Subtyp 1

Der erste selbstdefinierende Abschnitt beginnt bei X'24 'und ist im Beispiel **fett** ; X'0000003C' ist der Offset zum WTID-Datensatz, X'00D0 'ist die Länge des WTID-Satzes und X'0001' die Anzahl der WTID-Sätze.

Der zweite selbstdefinierende Abschnitt beginnt bei X'2C 'und ist in *kursiv* ; X'0000010C' ist der Offset zum WTAS-Datensatz, X'02C0 'ist die Länge des WTAS-Datensatzes und X'0001' die Anzahl der WTAS-Datensätze.

Der dritte selbstdefinierende Abschnitt beginnt bei X'34 'und ist im Beispiel **fett** ; X'000003CC' ist der Offset zum ersten WQST-Datensatz, X'0240 'ist die Länge des WQST-Datensatzes und X'0030' die Anzahl der WQST-Datensätze.

Abbildung 25 auf Seite 375 zeigt ein Beispiel für einen SMF-Datensatz des Typs 116, Subtyp 2.

```

000000 49740000 5E74002D 983B0100 229FD4E5 *....;.....MV*
000010 F4F1D4D8 F0F70002 F6F0F000 00004904 *41MQ07..600....*
000020 00700001 00000034 00000001 00000104 *.....}.....*
000030 02400020 F70000D0 E6E3C9C4 00000002 *..7..}WTID....*
.
.
.
000100 7F4A4BB8 F7020240 E6D8E2E3 00000001 *"...7.. WQST....*

```

Abbildung 25. Beispiel für SMF-Typ 116, Subtyp 2

Der erste selbstdefinierende Abschnitt beginnt bei X'24 'und ist im Beispiel **fett** ; X'00000034' ist der Offset zum WTID-Datensatz, X'00D0 'ist die Länge des WTID-Satzes und X'0001' die Anzahl der WTID-Sätze.

Der zweite selbstdefinierende Abschnitt beginnt bei X'2C 'und ist in *kursiv* ; X'00000104' ist der Offset zum ersten WQST-Datensatz, X'0240 'ist die Länge des WQST-Satzes und X'0020' die Anzahl der WQST-Datensätze.

Abbildung 26 auf Seite 375 zeigt ein Beispiel für einen SMF-Datensatz des Typs 116, Subtyp 1, bei dem keine Warteschlangen geöffnet wurden und es folglich keine selbst definierten Abschnitte für WQST-Datensätze gibt.

```

000000          5E740039 4E9B0104 344FD4E5 * .....|MV*
000010 F4F1D4D8 F0F70001 F6F0F000 000003DC *41MQ07..600....*
000020 00800001 00000034 00000001 00000104 *.....}.....*
000030 02D80001 F70000D0 E6E3C9C4 00000002 *..Q..7..WTID....*
000040 C1F8C5C1 C4C5D740 C1F8C5C1 C4C54040 *A8EADEP A8EADE *
000050 40404040 40404040 00000000 00000000 * .....*
000060 40404040 40404040 4040          *          *

```

Abbildung 26. Beispiel SMF-Typ 116, Subtyp 1-Datensatz ohne WQST-Datensätze

Der erste selbstdefinierende Abschnitt beginnt bei X'24 'und ist im Beispiel **fett** ; X'00000034' ist der Offset zum WTID-Datensatz, X'00D0 'ist die Länge des WTID-Satzes und X'0001' die Anzahl der WTID-Sätze.

Der zweite selbstdefinierende Abschnitt beginnt bei X'2C 'und ist in *kursiv* ; X'0000010C' ist der Offset zum WTAS-Datensatz, X'02D8 'ist die Länge des WTAS-Datensatzes und X'0001' die Anzahl der WTAS-Datensätze.

Es gibt keinen selbstdefinierenden Abschnitt, der einen WQST-Datensatz beschreibt, was dem dritten selbstdefinierenden Abschnitt in [Abbildung 24](#) auf Seite 374 entspricht.

## Layout der Datensätze für den Kanalinitiator mit dem SMF-Typ 116

In diesem Abschnitt wird das Layout der Kanalabrechnungsdaten (SMF-Typ 116, Subtype 10) beschrieben.

### Selbstdefinierendes Kapitel

Der selbstdefinierende Abschnitt für die Kanalabrechnungsdaten folgt dem Standard-SMF-Header. Er ist im standardmäßigen Triplet-Format strukturiert. Das Format der Triplets wird in der Struktur qws5 in der Headerdatei der Programmiersprache C `thlqual.SCSQC370(CSQDSMFC)` und im Assemblermakro `thlqual.SCSQMACS(CSQDQWS5)` beschrieben.

Unter [Tabelle 48](#) auf Seite 376 wird das Format des selbstdefinierenden Abschnitts gezeigt.

Tabelle 48. Struktur des selbstdefinierenden Abschnitts für die Kanalabrechnung

Offset: Dez.	Off- set: Hex	Typ	Länge	Name	Beschreibung
0	0	Integer	4	QWS50PSO	Offset zwischen dem Start des SMF-Datensatzes und dem ersten Instrumentierungsstandardheader (QWHS)
4	4	Integer	2	QWS50PSL	Länge von QWHS
6	6	Integer	2	QWS50PSN	Anzahl der Instanzen von QWHS
8	8	Integer	4	QWS50R1O	Offset zwischen dem Start des SMF-Datensatzes und dem ersten Datensatz mit Kanalabrechnungsdaten (QCST)
12	C	Integer	2	QWS50R1L	Länge von QCST
14	E	Integer	2	QWS50R1N	Anzahl der Instanzen von QCST

### Instrumentierungsstandardheader (QWHS)

Das Format von QWHS wird in der Struktur `qwhs` in der Headerdatei der Programmiersprache C `thlqual.SCSQC370 (CSQDSMFC)` und im Assemblermakro `thlqual.SCSQMACS (CSQDQWHS)` beschrieben. Sie enthält die folgenden Schlüsselfelder, die für Datensätze des Kanalinitiators SMF 116 relevant sind:

Tabelle 49. Schlüsselfelder in QWHS

Name	Länge	Beschreibung
QWHSNDA	1 Byte	Anzahl der selbstdefinierenden Abschnitte
QWHSSSID	4 Byte	Subsystemname
QWHS SMFC	1 Bit	Gibt an, ob mehrere SMF-Datensätze mit Informationen für dieses Intervall vorhanden sind. Wenn dieses Bit aktiviert ist, werden Informationen für dieses Intervall in weiteren SMF-Datensätzen fortgesetzt. Wenn dieses Bit nicht mehr vorhanden ist, ist dies der letzte oder einzige Datensatz.
QWHSTIME	8 Byte	Ortszeit beim Start des Intervalls im STCK-Format
QWHS DURN	8 Byte	Dauer vom Start bis zum Ende des Intervalls im STCK-Format
QWHSSTCK	8 Byte	Ende des Intervalls in UTC im STCK-Format

### Datensätze für Kanalabrechnungsdaten

Verwenden Sie dieses Thema als Referenz für Datensätze der Kanalabrechnungsdaten.

Das Format des Datensatzes für die Kanalabrechnung ist Assemblermakro `thlqual.SCSQMACS (CSQDQCST)` beschrieben. Das Format ist auch in der Headerdatei `thlqual.SCSQC370 (CSQDSMFC)` für die Programmiersprache C beschrieben. Beachten Sie, dass die Feldnamen in C alle in Kleinbuchstaben angegeben sind, z. B. `qcst`.

Die Kanalabrechnungsdaten enthalten Informationen zum Status und zur Statistik der einzelnen Kanal-exemplarinstanzen. Hierzu gehören:

- Durchschnittliche Netzzeit (`qcstntav`)
- Durchschnittliche Zeit beim Beenden (`qcstetav`)

- Grenzwert für Kanalstapeldaten (*qcstcbd*)
- Kanalbatchintervall (*qcstcbit*)
- Kanalstapelgröße (*qcstcbz*)
- Kanalzuteilernummer (*qcstdspn*)
- Kanaldisposition (*qcstchdp*)
- Kanalname (*qcstchnm*)
- Kanalstatus (*qcstchst*)
- Kanalstart-Zeit (*qcststrt*)
- Statuserfasste Kanalstatus (*qcstcltm*)
- Kanalstoppzeit (*qcstludt*)
- Kanaltyp (*qcstchty*)
- Allgemeiner Name (CN) von SSLCERTI (*qcstslcn*)
- Komprimierungsrate (*qcstcpa*)
- Verbindungsname (*qcstcnm*)
- Aktuelle gemeinsame Dialoge (*qcstcscv*)
- DNS-Auflösungszeit (*qcstdnrt*)
- Effektiver Wert des Parameters STATCHL (*qcststcl*)
- Uhrzeit der letzten Nachricht (*qcstlmst*)
- Maximale Netzzeit (*qcstntmx*)
- Maximale Zeit für Exit (*qcstetmx*)
- Minimale Netzzeit (*qcstntmn*)
- Mindestzeit beim Verlassen (*qcstetmn*)
- Name des fernen Warteschlangenmanagers oder der fernen Anwendung (*qcstrqmn*)
- Anzahl der Batches (*qcstbatc*)
- Anzahl der Byte für Nachrichtendaten (*qcstnbyt*)
- Anzahl der Byte für persistente Nachrichtendaten (*qcstnpby*)
- Anzahl der Byte, die für Nachrichtendaten und Steuerinformationen empfangen wurden (*qcstbyrc*)
- Anzahl der gesendeten Byte für Nachrichtendaten und Steuerinformationen (*qcstbyst*)
- Anzahl der vollen Stapelverarbeitungen (*qcstfuba*)
- Anzahl der Nachrichten oder Anzahl der MQI-Aufrufe (*qcstnmsg*)
- Anzahl persistenter Nachrichten (*qcstnpgm*)
- Anzahl der Einrückungsversuche (*qcstptrc*)
- Anzahl der leerer Übertragungswarteschlange (*qcstqetc*)
- Anzahl empfangener Übertragungspuffer ( **qcstbfrc** )
- Anzahl der gesendeten Übertragungspuffer (*qcstbfst*)
- Seriennummer von SSLPEER (*qcstslsn*)
- SSL CipherSpec (null bedeutet, dass TLS nicht verwendet wird) (*qcstslcs*)
- Das Datum und die Zeit der maximalen Netzzeit (*qcstntdt*)
- Das Datum und die Zeit der maximalen Zeit beim Beenden (*qcstetdt*)

Beachten Sie, dass diese beiden Felder für das Kanalabrechnungsfeld *qcstetmn* (Mindestzeit beim Verlassen) und *qcstntmn* (minimale Netzzeit) bei Nichtverwendung mit dem Hexadezimalwert 8FFFFFFF initialisiert werden.

Sie können diese Informationen verwenden, um den Durchsatz eines Kanals anzuzeigen, wenn sich die tatsächlichen Batches dem Grenzwert, der Latenzzeit des Netzes, Informationen zum fernen Ende, der Leistung des Benutzerexits usw. nähern.

Hier finden Sie ein Beispiel für die Kanalabrechnungsdaten, die mit dem IBM MQ-SupportPac [MP1B](#) formatiert wurden.

Die verfügbaren Felder basieren auf dem Befehl zum Anzeigen des Kanalstatus (DIS CHS) und der Kanalstatistik von IBM MQ auf Plattformen mit Ausnahme von z/OS, wobei einige zusätzliche Felder enthalten sind.

The data and time of the start and end of the record in local time, and the duration

```
SMF interval start      2014/03/26,02:30:00
SMF interval end       2014/03/26,02:45:00
SMF interval duration   899.997759 seconds
```

Information about the channel

```
Connection name      9.20.4.159
Channel disp         PRIVATE
Channel type         RECEIVER
Channel status       CLOSING
Channel STATCHL     HIGH
```

```
Start date & time      2014/03/26,02:44:58
Channel status collect time 2014/03/26,02:45:00
Last status changed   1900/01/01,00:00:00
Last msg time         2014/03/26,02:44:59
```

```
Batch size            50
Messages/batch        3.3
Number of messages    1,102
Number of persistent messages 1,102
Number of batches     335
Number of full batches 0
Number of partial batches 335
Buffers sent          337
Buffers received      1,272
Message data          5,038,344  4 MB
Persistent message data 5,038,344  4 MB
Non persistent message data 0 0 B
Total bytes sent      9,852  9 KB
Total bytes received  5,043,520  4 MB
Bytes received/Batch  15,055  14 KB
Bytes sent/Batch      29 29 B
Batches/Second       1
Bytes received/message 4,576  4 KB
Bytes sent/message    8 8 B
Bytes received/second 28,019  27 KB/sec
Bytes sent/second     54 54 B/sec
Compression rate      0
```

The name of the queue manager at the remote end of the connection

```
Remote qmgr/app      MQPH
Put retry count      0
```

## IBM MQ-Netz optimieren

Verwenden Sie die Optimierungstipps in diesem Abschnitt, um die Leistung Ihres Warteschlangenmanagements zu verbessern.

### Tuning Client- und Serververbindungskanäle

Die Standardeinstellungen für die Client- und Serververbindungskanäle wurden in IBM WebSphere MQ 7.0 geändert, damit gemeinsam genutzte Dialoge verwendet werden können. In IBM MQ 8.0 wurden dann Leistungsverbesserungen für verteilte Server eingeführt. Um die neuen Funktionen zu nutzen, die neben gemeinsam genutzten Dialogen eingeführt wurden, ohne dass die Leistung auf dem verteilten Server beeinträchtigt wird, setzen Sie **SHARECNV** auf Ihren Serververbindungskanälen von IBM MQ 8.0 oder höher auf 1.

Ab IBM WebSphere MQ 7.0 wird jeder Kanal standardmäßig für die Ausführung von bis zu 10 Clientdialogen pro Kanalinstanz definiert. Vor IBM WebSphere MQ 7.0 wurde jeder Dialog einer anderen Kanalinstanz zugeordnet. Die in IBM WebSphere MQ 7.0 hinzugefügten Erweiterungen enthalten außerdem die folgenden Funktionen:

- Bidirektionale Überwachungssignale
- Administratorstop-quiet
- Vorauslesen
- Asynchrone-konsumieren durch Clientanwendungen

Bei einigen Konfigurationen bringt die Verwendung von gemeinsamen Dialogen erhebliche Vorteile mit sich. Bei verteilten Servern ist die Verarbeitung von Nachrichten auf Kanälen, die die Standardkonfiguration von 10 gemeinsamen Datenaustauschvorgängen verwenden, jedoch im Durchschnitt 15% langsamer als auf Kanälen, die keine gemeinsamen Dialoge verwenden. In einer MQI-Kanalinstanz, die Dialoge gemeinsam verwendet, werden alle Dialoge in einem Socket vom selben Thread empfangen. Wenn die Dialoge, die einen Socket gemeinsam nutzen, alle ausgelastet sind, sind die Dialoge miteinander verbunden, um den empfangenden Thread zu verwenden. Der Konflikt verursacht Verzögerungen, und in dieser Situation ist die Verwendung einer kleineren Anzahl von gemeinsamen Gesprächen besser.

Sie verwenden den Parameter **SHARECNV**, um die maximale Anzahl an Dialogen anzugeben, die über eine bestimmte TCP/IP-Clientkanalinstanz gemeinsam genutzt werden. Einzelheiten zu den möglichen Werten und zu den neuen Funktionen, die in IBM WebSphere MQ 7.0 hinzugefügt wurden, finden Sie unter [MQI-Client: Standardverhalten von Clientverbindungen und Serververbindungen](#). Wenn Sie keine gemeinsamen Dialoge benötigen, gibt es zwei Einstellungen, die die beste Leistung in IBM MQ 8.0 oder höher bieten:

- In: **SHARECNV(1)**. Verwenden Sie diese Einstellung wann immer möglich. Sie beseitigt einen Konflikt bei der Verwendung des empfangenden Threads und Ihre Clientanwendungen können die Vorteile der neuen Funktionen nutzen, die in IBM WebSphere MQ 7.0 hinzugefügt wurden. Für diese Einstellung wurde die Leistung der verteilten Server in IBM MQ 8.0 oder höher deutlich verbessert. Die Leistungssteigerungen gelten für Clientanwendungen von IBM MQ 8.0 oder höher, die nicht vorlesungssynchrone Wartezeitaufrufe absetzen, z. B. C-Client-MQGET-Wartezeitaufrufe. Wenn diese Clientanwendungen verbunden sind, verwendet der verteilte Server weniger Threads und weniger Speicher, und der Durchsatz wird erhöht.
- In: **SHARECNV(0)**. Die Kanalinstanz verhält sich genau so, als ob es sich um einen IBM WebSphere MQ 6.0-Server- oder Clientkanalinstanz handelt, und Sie erhalten nicht die zusätzlichen Funktionen wie bidirektionale Überwachungssignale, die verfügbar sind, wenn Sie **SHARECNV** auf 1 oder höher setzen. Verwenden Sie den Wert 0 nur, wenn Clientanwendungen vorhanden sind, die nicht ordnungsgemäß ausgeführt werden, wenn Sie **SHARECNV** auf 1 oder höher setzen.

#### **Anmerkungen:**

- Wenn mit einem Server Clients verbunden sind, die Dialoge über einen Socket gemeinsam nutzen, und Sie die Einstellung für gemeinsame Dialoge von **SHARECNV(10)** auf **SHARECNV(1)** verringern, hat dies folgende Auswirkungen:
  - Höhere Socketverwendung auf dem Server.
  - Höhere Kanalinstanzen auf dem Server.

In diesem Fall können Sie auch die Einstellungen für **MaxChannels** und **MaxActiveChannels** erhöhen.

- Wenn Anwendungen mit einer nicht wiedereintrittsfähigen Bibliothek verbunden oder kompiliert werden, wird der Wert [CURSHCNV\(0\)](#) ausgehandelt, auch wenn ein höherer Wert in [CLNTCONN](#) und [SVRCONN](#) festgelegt ist.

Aus Gründen der Konsistenz mit früheren Releases wurde der SVRCONN-Standardkanal nicht aktualisiert. Daher müssen Sie **SHARECNV** explizit auf 1 oder 0 setzen.

#### **Zugehörige Konzepte**

[MQI-Client: Standardverhalten von Clientverbindung und Serververbindung](#)

## Verteilte Publish/Subscribe-Netze optimieren

Verwenden Sie die Optimierungstipps in diesem Abschnitt, um die Leistung der verteilten Publish/Subscribe-Cluster und -Hierarchien von IBM MQ zu verbessern.

### Zugehörige Konzepte

„Cluster überwachen“ auf Seite 323

In einem Cluster können Sie Anwendungsnachrichten, Steuernachrichten und Protokolle überwachen. Es gibt spezielle Überwachungsaspekte, wenn die Clusterauslastung zwischen zwei oder mehr Instanzen einer Warteschlange verteilt wird.

## Direkte Publish/Subscribe-Clusterleistung

In direkt weitergeleiteten Publish/Subscribe-Clustern werden Informationen wie z. B. Clusterthemen und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dieser Prozess kann eine erhebliche zusätzliche Belastung für das System verursachen. Um die Auswirkungen des Cluster-Managements auf die Leistung zu reduzieren, können Sie Aktualisierungen zu Zeiten hoher Systemauslastung ausführen, eine viel kleinere Untergruppe von Warteschlangenmanagern definieren, die an Publish/Subscribe beteiligt sind, und dass ein "überlappender" Cluster oder ein Wechsel zur Verwendung des Topic-Host-Routing erfolgt.

In einem Publish/Subscribe-Cluster sind zwei Workloadquellen für einen Warteschlangenmanager vorhanden:

- Nachrichten für Anwendungsprogramme direkt verarbeiten.
- Handhabung von Nachrichten und Kanälen, die für die Verwaltung des Clusters benötigt werden.

In einem typischen Punkt-zu-Punkt-Cluster ist die Auslastung des Clustersystems weitgehend auf die Informationen beschränkt, die von den Mitgliedern des Clusters nach Bedarf explizit angefordert werden. Daher können Sie in einem anderen Punkt als einem sehr großen Punkt-zu-Punkt-Cluster, z. B. einer, der Tausende von Warteschlangenmanagern enthält, den Leistungseffekt der Verwaltung des Clusters weitgehend absehen. In einem direkt weitergeleiteten Publish/Subscribe-Cluster werden jedoch Informationen wie z. B. Clusterthemen, Warteschlangenmanager-Mitgliedschaften und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dies kann zu einer erheblichen zusätzlichen Belastung des Systems führen. Daher müssen Sie die Auswirkungen der Clusterverwaltung auf die Leistung des Warteschlangenmanagers in Betracht ziehen, und zwar sowohl in der Ablaufsteuerung als auch in der Größe.

## Leistungsmerkmale von Direkt-Routing-Clustern

Vergleichen Sie einen Punkt-zu-Punkt-Cluster mit einem direkt weitergeleiteten Publish/Subscribe-Cluster in Bezug auf die Kernverwaltungstasks.

Zunächst ein Punkt-zu-Punkt-Cluster:

1. Wenn eine neue Clusterwarteschlange definiert ist, werden die Zielinformationen an die vollständigen WS-Manager-Repositorys übertragen und nur an andere Cluster-Member gesendet, wenn sie zum ersten auf eine Clusterwarteschlange verweisen (z. B., wenn eine Anwendung versucht, sie zu öffnen). Diese Informationen werden dann lokal vom WS-Manager zwischengespeichert, um die Notwendigkeit zu entfernen, die Informationen bei jedem Zugriff auf die Warteschlange fern abzurufen.
2. Das Hinzufügen eines WS-Managers zu einem Cluster wirkt sich nicht direkt auf die Auslastung anderer WS-Manager aus. Die Informationen zum neuen Warteschlangenmanager werden in die vollständigen Repositorys verschoben, aber die Kanäle für den neuen Warteschlangenmanager von anderen WS-Managern im Cluster werden nur erstellt und gestartet, wenn der Datenverkehr zum oder vom neuen WS-Manager fließt.

Zusammengefasst bezieht sich die Last auf einen Warteschlangenmanager in einem Punkt-zu-Punkt-Cluster auf den Nachrichtendatenverkehr, der für Anwendungsprogramme verarbeitet wird, und hängt nicht direkt mit der Größe des Clusters zusammen.

Zweitens, ein Direct-Routing-Publish/Subscribe-Cluster:

1. Wenn ein neues Clusterthema definiert wird, werden die Informationen an die vollständigen WS-Manager-Repositorys und von dort aus direkt an alle Member des Clusters übertragen, wodurch die Kanäle für jedes Member des Clusters aus den vollständigen Repositorys gestartet werden, wenn dies noch nicht begonnen hat. Wenn es sich um das erste direkte Clusterthema handelt, wird jedes WS-Manager-Member Informationen zu allen anderen WS-Manager-Membern im Cluster gesendet.
2. Wenn eine Subskription für ein Clusterthema in einer neuen Themenzeichenfolge erstellt wird, werden die Informationen direkt von diesem Warteschlangenmanager an alle anderen Member des Clusters übertragen, wodurch Kanäle für jedes Member des Clusters von diesem Warteschlangenmanager aus gestartet werden, falls dies noch nicht begonnen hat.
3. Wenn ein neuer Warteschlangenmanager einem vorhandenen Cluster beiträgt, werden Informationen zu allen Clusterthemen (und allen WS-Manager-Membern, wenn ein direktes Clusterthema definiert ist) von den vollständigen WS-Managern des Repositorys an den neuen WS-Manager übertragen. Der neue Warteschlangenmanager synchronisiert dann die Kenntnisse aller Subskriptionen für Clusterthemen im Cluster mit allen Membern des Clusters.

Zusammengefasst wächst die Clustermanagementbelastung an jedem WS-Manager in einem direkt weitergeleiteten Publish/Subscribe-Cluster mit der Anzahl an Warteschlangenmanagern, Clusterthemen und Änderungen an Subskriptionen für verschiedene Themenzeichenfolgen im Cluster, unabhängig von der lokalen Verwendung dieser Clusterthemen in jedem Warteschlangenmanager.

In einem großen Cluster oder einer, in der die Geschwindigkeit der Subskriptionen hoch ist, kann diese Ebene der Clusterverwaltung für alle Warteschlangenmanager einen erheblichen Systemaufwand darstellen.

## Reduzierung der Auswirkungen von Direct-Routing-Publish/Subscribe auf die Leistung

Wenn Sie die Auswirkungen der Clusterverwaltung auf die Leistung eines Direct-routed-Publish/Subscribe-Clusters verringern möchten, sollten Sie die folgenden Optionen berücksichtigen:

- Führen Sie Cluster-, Topic- und Subskriptionsaktualisierungen zu Off-Peak-Zeiten des Tages aus.
- Definieren Sie eine viel kleinere Untergruppe von Warteschlangenmanagern, die an Publish/Subscribe beteiligt sind, und stellen Sie sicher, dass ein "überlappender" Cluster vorhanden ist. Dieser Cluster ist dann der Cluster, in dem die Clusterthemen definiert sind. Obwohl einige WS-Manager jetzt in zwei Clustern vorhanden sind, wird die Gesamtwirkung von Publish/Subscribe reduziert:
  - Die Größe des Publish/Subscribe-Clusters ist kleiner.
  - Warteschlangenmanager, die nicht im Publish/Subscribe-Cluster enthalten sind, sind durch den Datenverkehr im Cluster-Management wesentlich weniger betroffen.

Wenn die vorherigen Optionen Ihre Leistungsprobleme nicht ausreichend lösen, sollten Sie stattdessen einen Publish/Subscribe-Cluster mit dem *Topic-Host* verwenden. Ein detaillierter Vergleich der direkten Routing- und Themenhostweiterleitung in Publish/Subscribe-Clustern finden Sie unter [Publish/Subscribe-Cluster entwerfen](#).

### Zugehörige Konzepte

Thema Host für Publish/Subscribe-Clusterleistung bereitstellen

Ein Publish/Subscribe-Cluster im Themenhost gibt Ihnen präzise Kontrolle darüber, welche WS-Manager die einzelnen Themen hosten. Diese Themenhosts werden zu den *Routing*-Warteschlangenmanagern für diese Verzweigung der Themenstruktur. Darüber hinaus müssen Warteschlangenmanager ohne Subskriptionen oder Publisher keine Verbindung zu den Themenhosts herstellen. Diese Konfiguration kann die Anzahl der Verbindungen zwischen WS-Managern im Cluster und die Menge der Informationen, die zwischen den Warteschlangenmanagern übergeben werden, erheblich reduzieren.

Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

## Subskriptionsleistung in Publish/Subscribe-Netzen

Verteiltes Publish/Subscribe in IBM MQ funktioniert durch die Weitergabe von Kenntnissen zur Position, an der Subskriptionen für verschiedene Themenzeichenfolgen im Warteschlangenmanagernetz erstellt wurden. Auf diese Weise kann der Warteschlangenmanager, auf dem eine Nachricht veröffentlicht wird, angeben, welche anderen WS-Manager eine Kopie der veröffentlichten Nachricht benötigen, damit sie mit ihren Subskriptionen übereinstimmen.

## Thema Host für Publish/Subscribe-Clusterleistung bereitstellen

Ein Publish/Subscribe-Cluster im Themenhost gibt Ihnen präzise Kontrolle darüber, welche WS-Manager die einzelnen Themen hosten. Diese Themenhosts werden zu den *Routing* -Warteschlangenmanagern für diese Verzweigung der Themenstruktur. Darüber hinaus müssen Warteschlangenmanager ohne Subskriptionen oder Publisher keine Verbindung zu den Themenhosts herstellen. Diese Konfiguration kann die Anzahl der Verbindungen zwischen WS-Managern im Cluster und die Menge der Informationen, die zwischen den Warteschlangenmanagern übergeben werden, erheblich reduzieren.

Der Publish/Subscribe-Cluster eines Topic-Hosts verhält sich wie folgt:

- Die Themen werden manuell auf einzelnen WS-Managern des *Topic-Hosts* im Cluster definiert.
- Wenn eine Subskription für einen Cluster-WS-Manager ausgeführt wird, werden Proxy-Subskriptionen nur auf den Themenhosts erstellt.
- Wenn eine Anwendung Informationen zu einem Thema veröffentlicht, leitet der empfangende WS-Manager die Veröffentlichung an einen Warteschlangenmanager weiter, der das Thema hostet. Der Topic-Host sendet die Veröffentlichung dann an alle Warteschlangenmanager im Cluster, die gültige Subskriptionen für das Thema besitzen.

Eine detailliertere Einführung in das Thema Host-Routing finden Sie unter [Topic host routing in clusters](#).

Bei vielen Konfigurationen ist das Thema-Host-Routing eine besser geeignete Topologie als *direktes Routing*, da es die folgenden Vorteile bietet:

- Verbesserte Skalierbarkeit größerer Cluster. Nur die TOPICHOST-Warteschlangenmanager müssen mit allen anderen Warteschlangenmanagern im Cluster eine Verbindung herstellen können. Deshalb gibt es weniger Kanäle zwischen den Warteschlangenmanagern und zwischen den Warteschlangenmanagern weniger Datenverkehr in Verbindung mit der Publish/Subscribe-Verwaltung, als dies beim direkten Routing der Fall ist. Wenn sich Subskriptionen auf einem Warteschlangenmanager ändern, müssen nur die Topic-Host-Warteschlangenmanager informiert werden.
- Größere Kontrollmöglichkeiten bei der physischen Konfiguration. Beim direkten Routing setzen alle Warteschlangenmanager sämtliche Rollen voraus und müssen daher gleichermaßen befähigt sein. Beim TOPICHOST-Routing wählen Sie explizit die TOPICHOST-Warteschlangenmanager aus. Damit können Sie sicherstellen, dass diese Warteschlangenmanager auf entsprechend leistungsfähigen Systemen aktiv sind, während die anderen Warteschlangenmanager auf weniger leistungsfähigen Systemen installiert werden können.

Beim TOPICHOST-Routing gelten für Ihr System jedoch bestimmte Einschränkungen:

- Die Systemkonfiguration und -wartung muss sorgfältiger geplant werden als dies beim DIRECT-Routing erforderlich ist. Sie müssen in der Themenstruktur die Punkte festlegen, die zu Clustern zusammengeslossen werden sollen; ebenso müssen Sie festlegen, wo im Cluster sich die Themendefinitionen befinden.
- Wenn ein neues Thema mit TOPICHOST-Routing definiert wird, werden die Informationen wie bei Themen, für die DIRECT-Routing definiert ist, an die Warteschlangenmanager mit vollständigem Repository und von dort direkt an alle Clustermitglieder übertragen. Dadurch werden von den vollständigen Repositories aus Kanäle zu jedem Clustermitglied gestartet (sofern dies noch nicht geschehen ist).
- Veröffentlichungen werden immer von einem Warteschlangenmanager, bei dem es sich nicht um einen Themenhost handelt, an einen Warteschlangenmanager, der als Themenhost dient, gesendet; dies ist auch dann der Fall, wenn im Cluster keine Subskriptionen vorhanden sind. Wenn daher davon ausgegangen werden kann, dass Subskriptionen vorliegen, oder wenn der Aufwand für die globale Konnektivität und für globale Informationen höher ist als der eventuell zusätzliche Datenverkehr beim Übertragen von Veröffentlichungen, sollten Sie Themen-Routing verwenden.

- Nachrichten, die auf Warteschlangenmanagern veröffentlicht werden, die keine Hosts sind, gehen nicht direkt an den Warteschlangen, der die Subskription hostet, sondern werden immer über einen TOPICHOST-Warteschlangenmanager weitergeleitet. Dadurch erhöhen sich der Gesamtaufwand für den Cluster und die Latenzzeit bei der Nachrichtenübertragung, wodurch sich die Leistung verschlechtert.

**Anmerkung:** Für bestimmte Konfigurationen können Sie diese Einschränkung sinnvoll entfernen, wie im Abschnitt [Topic-Host-Routing mit zentralisierten Publishern oder Subskribenten](#) beschrieben.

- Die Verwendung nur eines Warteschlangenmanagers als Themenhost stellt einen SPoF (Single Point of Failure) für alle Nachrichten dar, die zu einem Thema veröffentlicht werden. Durch eine Definition mehrerer Themenhosts wird ein solcher SPoF ausgeschlossen. Bei Verwendung mehrerer Hosts ändert sich allerdings die Reihenfolge, in der veröffentlichte Nachrichten für Subskriptionen empfangen werden.
- TOPICHOST-Warteschlangenmanager erzeugen ein zusätzliches Nachrichtenvolumen, weil sie Veröffentlichungsdatenverkehr von mehreren Warteschlangenmanagern verarbeiten müssen. Dieses Volumen kann verringert werden, indem mehrere TOPICHOSTs für ein einzelnes Thema verwendet werden (wobei die Reihenfolge der Nachrichten nicht beibehalten wird) oder indem verschiedene Warteschlangenmanager als Hosts für weitergeleitete Themen für verschiedene Zweige der Themenstruktur verwendet werden.

## Thema Host-Routing mit zentralisierten Publishern oder Subskribenten

Wenn Sie den zusätzlichen "Hop" entfernen möchten, wenn Veröffentlichungen immer über einen Topic-Host-Warteschlangenmanager an Subskriptionen weitergeleitet werden, konfigurieren Sie die Publisher oder die Subskriptionen auf demselben Warteschlangenmanager, auf dem sich das Thema befindet. Dieser Ansatz bringt maximale Leistungsvorteile in den folgenden beiden Fällen:

- Topics mit vielen Publishern und wenigen Abonnements. In diesem Fall hosten die Subskriptionen auf dem Topic-Host-Warteschlangenmanager.
- Topics mit wenigen Publishern und vielen Abonnements. In diesem Fall hosten die Publisher auf dem Topic-Host-Warteschlangenmanager.

Die folgende Abbildung zeigt einen Topic-Host-Warteschlangenmanager, der auch die Subskriptionen enthält. Dieser Ansatz entfernt den zusätzlichen "Hop" zwischen dem Bereitsteller und dem Subskribenten und reduziert die unnötige gemeinsame Nutzung von Subskriptionswissen über alle Member des Clusters hinweg:

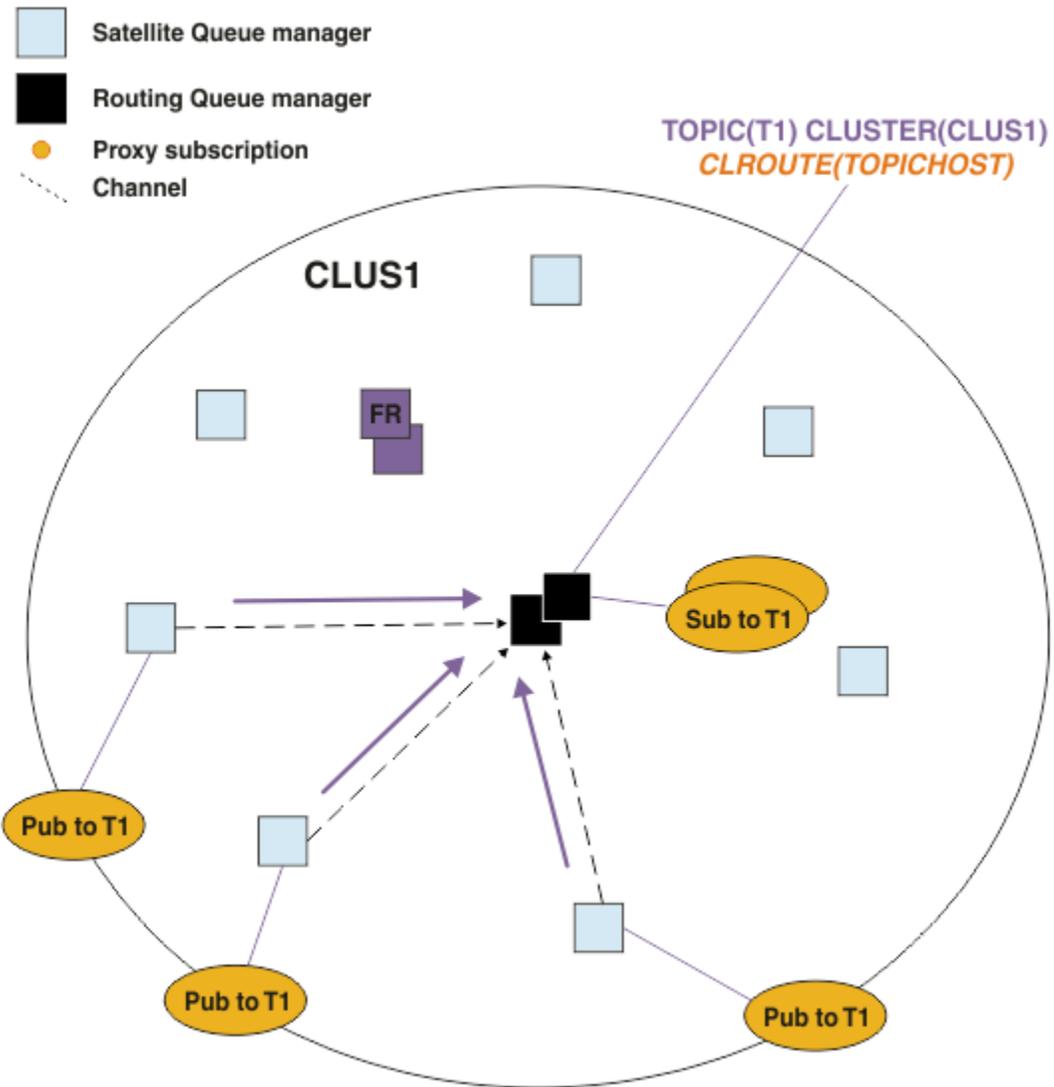


Abbildung 27. Hosting-Subskriptionen für einen Topic-Host-Warteschlangenmanager

Die folgende Abbildung zeigt einen Topic-Host-Warteschlangenmanager, der auch die Publizier enthält. Dieser Ansatz entfernt den zusätzlichen "Hop" zwischen dem Bereitsteller und dem Subskribenten und reduziert die unnötige gemeinsame Nutzung von Subskriptionswissen über alle Member des Clusters hinweg:

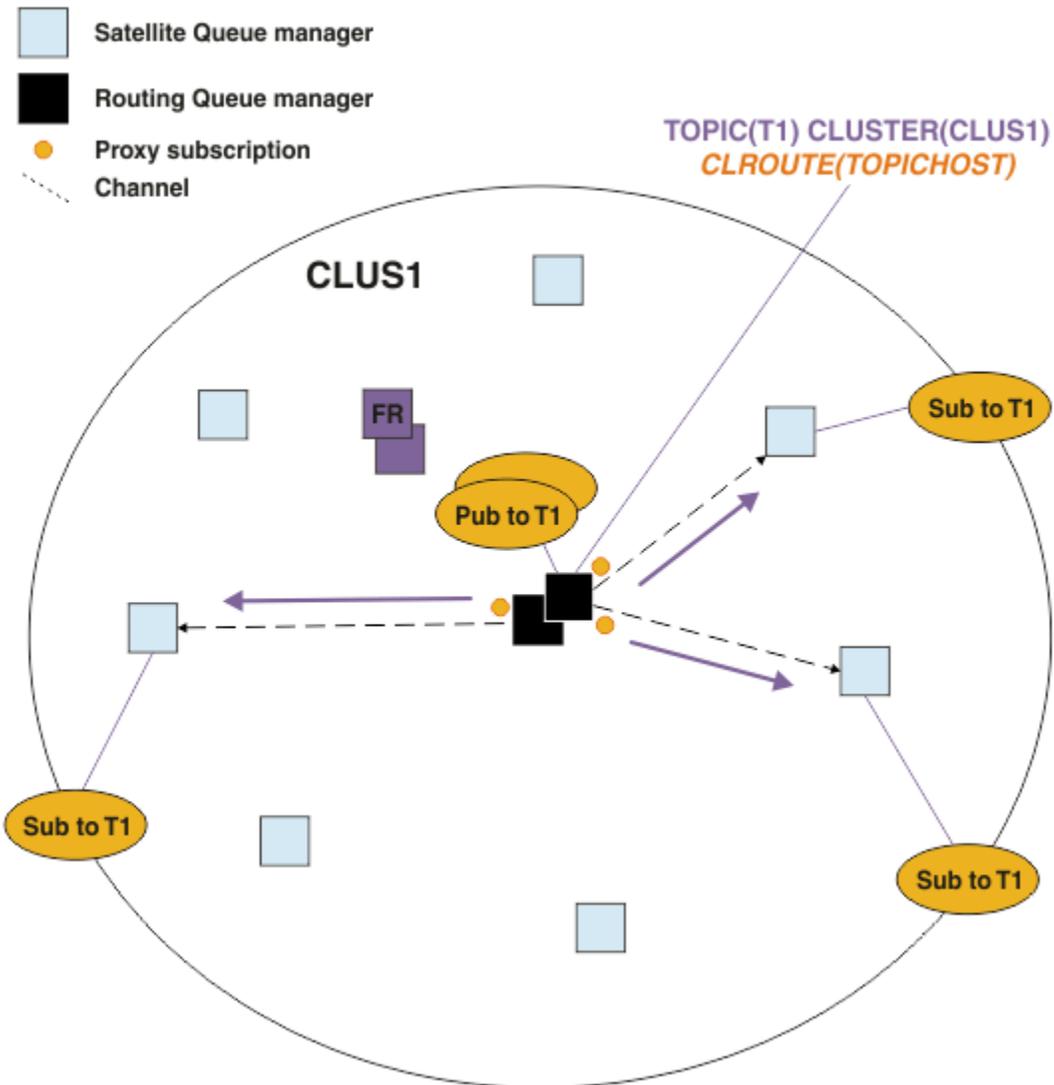


Abbildung 28. Hosting-Veröffentlichungen zu einem Topic-Host-Warteschlangenmanager

### Zugehörige Konzepte

#### Direkte Publish/Subscribe-Clusterleistung

In direkt weitergeleiteten Publish/Subscribe-Clustern werden Informationen wie z. B. Clusterthemen und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dieser Prozess kann eine erhebliche zusätzliche Belastung für das System verursachen. Um die Auswirkungen des Cluster-Managements auf die Leistung zu reduzieren, können Sie Aktualisierungen zu Zeiten hoher Systemauslastung ausführen, eine viel kleinere Untergruppe von Warteschlangenmanagern definieren, die an Publish/Subscribe beteiligt sind, und dass ein "überlappendes" Cluster oder ein Wechsel zur Verwendung des Topic-Host-Routing erfolgt.

#### Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

#### Subskriptionsleistung in Publish/Subscribe-Netzen

Verteiltes Publish/Subscribe in IBM MQ funktioniert durch die Weitergabe von Kenntnissen zur Position, an der Subskriptionen für verschiedene Themenzeihenfolgen im Warteschlangenmanagernetz erstellt wurden. Auf diese Weise kann der Warteschlangenmanager, auf dem eine Nachricht veröffentlicht wird,

angeben, welche anderen WS-Manager eine Kopie der veröffentlichten Nachricht benötigen, damit sie mit ihren Subskriptionen übereinstimmen.

## Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

In einer Punkt-zu-Punkt-Messaging-Topologie ist die Beziehung zwischen Nachrichtenkonsumenten und Nachrichtenproduzenten leicht verständlich. Sie können Schätzungen von Nachrichtenproduktion und -verbrauch, Warteschlange nach Warteschlange, Kanal nach Kanal abrufen. Wenn ein Mangel an Ausgewogenheit besteht, werden die Engpässe schnell erkannt und dann behoben.

Es ist schwieriger zu prüfen, ob Publisher und Subskribenten in einer Publish/Subscribe-Topologie ausgeglichen sind. Beginnen Sie bei jeder Subskription, und arbeiten Sie mit den WS-Managern, die Publisher für das Thema verwenden, zurück. Berechnen Sie die Anzahl der Veröffentlichungen, die von jedem WS-Manager zu jedem Subskribenten fließen.

Jede Veröffentlichung, die mit einer Subskription in einem fernen Warteschlangenmanager (basierend auf Proxy-Subskriptionen) übereinstimmt, wird in eine Übertragungswarteschlange gestellt. Wenn mehrere ferne WS-Manager über Proxy-Subskriptionen für diese Veröffentlichung verfügen, werden mehrere Kopien der Nachricht in eine Übertragungswarteschlange gestellt, die jeweils für einen anderen Senderkanal bestimmt sind.

In einem Publish/Subscribe-Cluster werden diese Veröffentlichungen in der Warteschlange `SYSTEM.INTER.QMGR.PUBS` auf den fernen Warteschlangenmanagern, die die Subskriptionen enthalten, als Ziel verwendet. In einer Hierarchie richtet sich jede Veröffentlichung an die Warteschlange `SYSTEM.BROKER.DEFAULT.STREAM` oder an alle anderen Datenstromwarteschlangen, die in der Datei `SYSTEM.QPUBSUB.QUEUE.NAMELIST` auf den fernen Warteschlangenmanagern aufgeführt sind. Jeder WS-Manager verarbeitet Nachrichten, die in dieser Warteschlange ankommen, und stellt sie den richtigen Subskriptionen auf diesem Warteschlangenmanager zu.

Überwachen Sie daher die Auslastung an den folgenden Stellen, an denen Engpässe auftreten können:

- Überwachen Sie die Auslastung in den einzelnen Subskriptionswarteschlangen.
  - Dieser Engpass impliziert, dass die subskribierende Anwendung die Veröffentlichungen nicht so schnell konsumiert, wie sie veröffentlicht werden.
- Überwachen Sie die Auslastung in der Warteschlange `SYSTEM.INTER.QMGR.PUBS` oder in den Datenstromwarteschlangen.
  - Dieser Engpass impliziert, dass der Warteschlangenmanager Veröffentlichungen von einem oder mehreren fernen Warteschlangenmanagern schneller empfängt, als er sie an die lokalen Subskriptionen verteilen kann.
  - Wenn Sie in einem Topic-Host-Warteschlangenmanager bei der Verwendung von Topic-Host-Routing in einem Cluster sehen, sollten Sie zusätzliche Warteschlangenmanager-Topic-Hosts verwenden, die es ermöglichen, die Workload der Veröffentlichung auf diese zu balancieren. Dies wirkt sich jedoch auf die Nachrichtenreihenfolge in allen Veröffentlichungen aus. Weitere Informationen finden Sie unter [Topic-Host-Routing unter Verwendung mehrerer Topic-Hosts für ein einzelnes Thema](#).
- Überwachen Sie die Auslastung auf den Kanälen zwischen dem Veröffentlichungswarteschlangenmanager und den subskribierenden WS-Managern, die von den Übertragungswarteschlangen des Veröffentlichungswarteschlangenmanagers gespeist werden.
  - Dieser Engpass impliziert, dass entweder ein oder mehrere Kanäle nicht aktiv sind, oder Nachrichten werden schneller in den lokalen WS-Manager veröffentlicht, als die Kanäle sie an den fernen Warteschlangenmanager übergeben können.
  - Wenn Sie einen Publish/Subscribe-Cluster verwenden, können Sie zusätzliche Clusterempfängerkanäle auf dem Zielwarteschlangenmanager definieren. Auf diese Weise kann die Auslastung der Veröffentlichungen auf diese Weise ausgeglichen werden. Dies wirkt sich jedoch auf die Nachrichten-

reihenfolge in Veröffentlichungen aus. Stellen Sie außerdem in Betracht, eine Konfiguration mit mehreren Clusterübertragungswarteschlangen zu verschieben, da dadurch die Leistung unter bestimmten Umständen verbessert werden kann.

- Wenn die Veröffentlichungsanwendung eine Publish/Subscribe-Schnittstelle in der Warteschlange verwendet, überwachen Sie die Auslastung (a) in die Warteschlange `SYSTEM.BROKER.DEFAULT.STREAM` und alle anderen Datenstromwarteschlangen, die in der Warteschlange `SYSTEM.QPUBSUB.QUEUE.NAMELIST` aufgeführt sind; und (b) die Warteschlange `SYSTEM.BROKER.DEFAULT.SUBPOINT` und alle anderen in der Datei `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST` aufgelisteten Unterpunktwarteschlangen.
  - Dieser Engpass impliziert, dass Nachrichten von lokalen Veröffentlichungsanwendungen schneller gestellt werden, als der lokale WS-Manager die Nachrichten verarbeiten kann.

## Zugehörige Konzepte

### Direkte Publish/Subscribe-Clusterleistung

In direkt weitergeleiteten Publish/Subscribe-Clustern werden Informationen wie z. B. Clusterthemen und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dieser Prozess kann eine erhebliche zusätzliche Belastung für das System verursachen. Um die Auswirkungen des Cluster-Managements auf die Leistung zu reduzieren, können Sie Aktualisierungen zu Zeiten hoher Systemauslastung ausführen, eine viel kleinere Untergruppe von Warteschlangenmanagern definieren, die an Publish/Subscribe beteiligt sind, und dass ein "überlappendes" Cluster oder ein Wechsel zur Verwendung des Topic-Host-Routing erfolgt.

### Thema Host für Publish/Subscribe-Clusterleistung bereitstellen

Ein Publish/Subscribe-Cluster im Themenhost gibt Ihnen präzise Kontrolle darüber, welche WS-Manager die einzelnen Themen hosten. Diese Themenhosts werden zu den *Routing*-Warteschlangenmanagern für diese Verzweigung der Themenstruktur. Darüber hinaus müssen Warteschlangenmanager ohne Subskriptionen oder Publisher keine Verbindung zu den Themenhosts herstellen. Diese Konfiguration kann die Anzahl der Verbindungen zwischen WS-Managern im Cluster und die Menge der Informationen, die zwischen den Warteschlangenmanagern übergeben werden, erheblich reduzieren.

### Subskriptionsleistung in Publish/Subscribe-Netzen

Verteiltes Publish/Subscribe in IBM MQ funktioniert durch die Weitergabe von Kenntnissen zur Position, an der Subskriptionen für verschiedene Themenzeichenfolgen im Warteschlangenmanagernetz erstellt wurden. Auf diese Weise kann der Warteschlangenmanager, auf dem eine Nachricht veröffentlicht wird, angeben, welche anderen WS-Manager eine Kopie der veröffentlichten Nachricht benötigen, damit sie mit ihren Subskriptionen übereinstimmen.

### „Cluster überwachen“ auf Seite 323

In einem Cluster können Sie Anwendungsnachrichten, Steuernachrichten und Protokolle überwachen. Es gibt spezielle Überwachungsaspekte, wenn die Clusterauslastung zwischen zwei oder mehr Instanzen einer Warteschlange verteilt wird.

## Subskriptionsleistung in Publish/Subscribe-Netzen

Verteiltes Publish/Subscribe in IBM MQ funktioniert durch die Weitergabe von Kenntnissen zur Position, an der Subskriptionen für verschiedene Themenzeichenfolgen im Warteschlangenmanagernetz erstellt wurden. Auf diese Weise kann der Warteschlangenmanager, auf dem eine Nachricht veröffentlicht wird, angeben, welche anderen WS-Manager eine Kopie der veröffentlichten Nachricht benötigen, damit sie mit ihren Subskriptionen übereinstimmen.

Dieser Ansatz minimiert das Senden von veröffentlichten Nachrichten an Warteschlangenmanager, auf denen keine übereinstimmenden Subskriptionen vorhanden sind. Die Weitergabe des Subskriptionswissens kann jedoch zu einem erheblichen Systemaufwand werden, wenn die Anzahl der Themenzeichenfolgen, die subskribiert werden, hoch ist und sich durch häufige Subskriptionserstellung und -löschung ständig ändert.

Sie können die Leistung beeinflussen, indem Sie anpassen, wie Veröffentlichungen und Subskriptionen in Ihrem Publish/Subscribe-Netz ausgeführt werden. Wenn Ihr Netzdatenverkehr nur wenige Veröffentlichungen enthält und die Erstellung, Löschung oder Änderung der Schnellschreibung beendet ist,

können Sie die Subskription von Subskriptionsinformationen an alle Warteschlangenmanager stoppen und alle Veröffentlichungen an alle WS-Manager im Netz weiterleiten. Sie können auch den Fluss von Proxy-Subskriptionen und Veröffentlichungen für ein bestimmtes Thema zwischen verbundenen Warteschlangenmanagern einschränken, den Fluss von Proxy-Subskriptionen, die Platzhalterzeichen enthalten, beschränken und die Anzahl und die transiente Art von Themenzeichenfolgen reduzieren.

## **Weitergabe von Einzelsubskriptionen und *überall* veröffentlichen**

*Überall veröffentlichen* ist eine Alternative zur Weitergabe einzelner Abonnements. Bei der individuellen Weitergabe werden nur Veröffentlichungen, die über eine übereinstimmende Subskription in einem Warteschlangenmanager verfügen, an diesen Warteschlangenmanager weitergeleitet. Mit *überall veröffentlichen* werden alle Veröffentlichungen an alle WS-Manager im Netz weitergeleitet. Die empfangenden WS-Manager stellen dann die Veröffentlichungen zur Verfügung, die mit lokalen Subskriptionen übereinstimmen.

### **Weitergabe von Einzelabonn**

Dieser Mechanismus führt dazu, dass der Datenverkehr zwischen den Veröffentlichungsdaten des Warteschlangenmanagers am wenigsten liegt, da nur die Veröffentlichungen gesendet werden, die mit Subskriptionen für einen Warteschlangenmanager übereinstimmen.

Es gilt jedoch auch Folgendes:

- Für jede einzelne Themenzeichenfolge, die subskribiert ist, wird eine Proxy-Subskription an andere WS-Manager in der Publish/Subscribe-Topologie gesendet. Die Gruppe der Warteschlangenmanager richtet sich nach dem verwendeten Routing-Modell, wie im Abschnitt Verteiltes Publish/Subscribe-Netz planen beschrieben.
  - Dieser Messaging-Systemaufwand kann von Bedeutung sein, wenn viele Tausende von Subskriptionen erstellt oder gelöscht werden (z. B. wenn alle nicht permanenten Subskriptionen nach einem Neustart eines Warteschlangenmanagers erneut erstellt werden) oder wenn sich die Subskriptionsgruppe schnell ändert und jeweils eine andere Themenzeichenfolge verwendet wird.
  - Die Anzahl der WS-Manager, an die die Proxy-Subskription weitergegeben wird, wirkt sich auch auf den Umfang des Systemaufwands aus.
- Proxy-Subskriptionen werden mit asynchronem Messaging an andere WS-Manager geleitet. Dies hat die folgende Wirkung:
  - Es gibt eine Verzögerung zwischen der Erstellung einer Subskription und der Erstellung, Bereitstellung und Verarbeitung der Proxy-Subskription durch die anderen Warteschlangenmanager.
  - Nachrichten, die in diesen Warteschlangenmanagern in diesem Intervall veröffentlicht werden, werden nicht an die ferne Subskription zugestellt.

### **Überall veröffentlichen**

Bei diesem Mechanismus gibt es keinen Systemaufwand für die Proxy-Subskription pro Themenzeichenfolge auf dem System. Dies bedeutet, dass die Erstellung, Löschung oder Änderung der Schnellschubskription nicht zu einer erhöhten Netzauslastung und -verarbeitung führt.

Es besteht auch keine Verzögerung zwischen der Erstellung einer Subskription und den Veröffentlichungen, die an einen Warteschlangenmanager fließen, da alle Veröffentlichungen an alle Warteschlangenmanager flossen. Daher gibt es kein Fenster, in dem die Veröffentlichungen nicht an neu erstellte ferne Subskriptionen zugestellt werden.

Es gilt jedoch auch Folgendes:

- Wenn Sie alle Veröffentlichungen an alle Warteschlangenmanager in der Publish/Subscribe-Topologie senden, kann es zu einem übermäßigen Datenaustausch im Netz kommen, in dem Veröffentlichungen nicht über übereinstimmende Subskriptionen auf den einzelnen Warteschlangenmanagern verfügen.
  - Je größer die Anzahl der WS-Manager in der Topologie ist, um so größer ist der Systemaufwand.

Sie sollten die Verwendung des Mechanismus *publish überall* in Betracht ziehen, wenn Sie erwarten, dass eine Veröffentlichung von einem erheblichen Teil Ihrer Warteschlangenmanager subskribiert wird, oder wenn die Proxy-Subskriptionsüberschrift wegen der Häufigkeit von Subskriptionsänderungen zu hoch

sind. Sie sollten die individuelle Proxy-Abonnementweiterleitung verwenden, wenn Sie einen erhöhten Messaging-Datenverkehr erleben, wenn die Veröffentlichungen an alle Warteschlangenmanager gesendet werden, und nicht an die Warteschlangenmanager mit übereinstimmenden Subskriptionen.

Sie können das Verhalten *publish überall* auf jeder Ebene in der Themenstruktur festlegen. Um *überall veröffentlichen* zu aktivieren, setzen Sie den Parameter **PROXYSUB** für ein übergeordnetes Themenobjekt auf ERZWINGEN. Dies führt zu einer einzelnen Proxy-Subskription mit Platzhalterzeichen, die alle Topics unter diesem Themenobjekt in der Themenstruktur abgleicht. Wenn es für ein Clusterthemenobjekt festgelegt wird, wird das Attribut **PROXYSUB (FORCE)** an jeden Warteschlangenmanager im Netz weitergegeben, nicht nur an den Warteschlangenmanager, auf dem das Thema definiert wurde.

**Anmerkung:** Bei Verwendung in einer Hierarchie legen Sie **PROXYSUB (FORCE)** individuell auf jedem Warteschlangenmanager fest, sodass der Topologiemechanismus natürlich die Anzahl der Kanäle begrenzt. Wenn jedoch in einem Cluster verwendet wird, können viele zusätzliche Kanäle gestartet werden:

- In einem Topic-Host-Routing-Cluster werden die Kanäle von jedem WS-Manager zu jedem Topic-Host-Warteschlangenmanager gestartet.
- In einem direkt weitergeleiteten Cluster werden Kanäle von jedem WS-Manager zu jedem anderen Warteschlangenmanager gestartet.

Der Systemaufwand für das Starten vieler Kanäle ist in einem direkt weitergeleiteten Cluster am stärksten ausgeprägt und kann zu Leistungsproblemen führen. Siehe [„Direkte Publish/Subscribe-Clusterleistung“](#) auf Seite 380.

## Andere Möglichkeiten, den Fluss von Proxy-Subskriptionen und Veröffentlichungen zwischen verbundenen Warteschlangenmanagern zu beschränken

### Themenzeichenfolgen konsolidieren

Die Verwendung vieler unterschiedlicher, transienter Themenzeichenfolgen führt zu einem gewissen Verwaltungsaufwand für jeden Warteschlangenmanager in dem System, an dem Publisher oder Subskriptionen angehängt sind. Sie sollten die Verwendung von Themenzeichenfolgen regelmäßig bewerten, um festzustellen, ob sie konsolidiert werden können. Durch die Verringerung der Anzahl und der transienten Merkmale von Themenzeichenfolgen und damit der Publisher und Subskriptionen für diese werden die Auswirkungen auf das System reduziert.

### Schränken Sie die Veröffentlichungs- und Subskription

Für ein bestimmtes Thema können Sie die Einstellungen von [Veröffentlichungsbereich](#) und [Subskriptionsbereich](#) verwenden, damit Veröffentlichungen und Subskriptionen für den Warteschlangenmanager, für den sie definiert sind, lokal bleiben.

### Blocksubskriptionen für Themen mit Platzhalterzeichen

Sie können den Ablauf von Proxy-Subskriptionen, die Platzhalterzeichen enthalten, beschränken, indem Sie das Attribut **Topic PLATZHALTER** auf BLOCK setzen. Weitere Informationen finden Sie unter [Platzhalterzeichen in Proxy-Subskriptionen](#).

Siehe auch [„Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen“](#) auf Seite 386

## Proxy-Subskriptionsdatenverkehr in Clustern überwachen

Wenn Sie die Auslastung des Systems durch den Proxy-Subskriptionsdatenverkehr berücksichtigen, überwachen Sie zusätzlich zu den in [„Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen“](#) auf Seite 386 aufgeführten Warteschlangen auch die folgenden Clusterwarteschlangen:

- Die Warteschlange SYSTEM.INTER.QMGR.FANREQ auf dem Subskribentenwarteschlangenmanager.
- Die Warteschlange SYSTEM.INTER.QMGR.CONTROL auf allen anderen Warteschlangenmanagern im Cluster.

Jeder signifikante Nachrichtenrückstand in diesen Warteschlangen impliziert, dass entweder die Rate der Subskriptionsänderung für das System zu hoch ist oder dass ein Warteschlangenmanager im Cluster nicht ordnungsgemäß funktioniert. Wenn Sie vermuten, dass das Problem bei einem bestimmten WS-Manager liegt, überprüfen Sie, ob die Publish/Subscribe-Unterstützung für diesen Warteschlangenmanager nicht inaktiviert ist. Siehe **PSMODE** in [ALTER QMGR \(Warteschlangenmanager\)](#).

## Zugehörige Konzepte

### Direkte Publish/Subscribe-Clusterleistung

In direkt weitergeleiteten Publish/Subscribe-Clustern werden Informationen wie z. B. Clusterthemen und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dieser Prozess kann eine erhebliche zusätzliche Belastung für das System verursachen. Um die Auswirkungen des Cluster-Managements auf die Leistung zu reduzieren, können Sie Aktualisierungen zu Zeiten hoher Systemauslastung ausführen, eine viel kleinere Untergruppe von Warteschlangenmanagern definieren, die an Publish/Subscribe beteiligt sind, und dass ein "überlappender" Cluster oder ein Wechsel zur Verwendung des Topic-Host-Routing erfolgt.

### Thema Host für Publish/Subscribe-Clusterleistung bereitstellen

Ein Publish/Subscribe-Cluster im Themenhost gibt Ihnen präzise Kontrolle darüber, welche WS-Manager die einzelnen Themen hosten. Diese Themenhosts werden zu den *Routing*-Warteschlangenmanagern für diese Verzweigung der Themenstruktur. Darüber hinaus müssen Warteschlangenmanager ohne Subskriptionen oder Publisher keine Verbindung zu den Themenhosts herstellen. Diese Konfiguration kann die Anzahl der Verbindungen zwischen WS-Managern im Cluster und die Menge der Informationen, die zwischen den Warteschlangenmanagern übergeben werden, erheblich reduzieren.

### Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

### Proxy-Subskriptionen in einem Publish/Subscribe-Netz

## Reduzieren der Anzahl unerwünschter Themen in der Themenstruktur

Die Leistung eines Publish/Subscribe-Systems wird verbessert, indem die Anzahl unerwünschter Themen in der Themenstruktur reduziert wird. Was ist ein unerwünschtes Thema und wie können Sie sie entfernen?

Sie können eine große Anzahl von Themen erstellen, ohne die Leistung zu beeinträchtigen. Einige Methoden zum Verwenden von Publish/Subscribe führen jedoch dazu, dass Themenstrukturen kontinuierlich erweitert werden. Eine außergewöhnlich große Anzahl von Themen wird einmal erstellt und nie wieder verwendet. Die wachsende Anzahl von Themen kann zu einem Leistungsproblem werden.

Wie können Sie Designs vermeiden, die zu einer großen und wachsenden Zahl unerwünschter Themen führen? Was können Sie tun, um dem WS-Manager zu helfen, unerwünschte Themen aus der Themenstruktur zu entfernen?

Der WS-Manager erkennt ein unerwünschtes Thema, da es seit 30 Minuten ungenutzt ist. Der WS-Manager entfernt nicht verwendete Themen aus der Themenstruktur für Sie. Die Dauer von 30 Minuten kann geändert werden, indem das Warteschlangenmanagerattribut **TREELIFE** geändert wird. Sie können dem Warteschlangenmanager helfen, unerwünschte Themen zu entfernen, indem Sie sicherstellen, dass der WS-Manager nicht mehr verwendet wird. Der Abschnitt „Was ist ein nicht verwendetes Thema?“ auf Seite [390](#) erläutert, was ein nicht verwendetes Thema ist.

Ein Programmierer, der jede Anwendung entwirft und insbesondere eine lange laufende Anwendung entwirft, berücksichtigt seine Ressourcennutzung: Wie viel Ressourcen benötigt das Programm, gibt es irgendwelche unbegrenzten Anforderungen, und irgendwelche Ressourcenlecks? Topics sind eine Ressource, die Publish/Subscribe-Programme verwenden. Scrüönisieren Sie die Verwendung von Themen wie alle anderen Ressourcen, die ein Programm verwendet.

## Was ist ein nicht verwendetes Thema?

Bevor Sie definieren, was ein nicht verwendetes Thema ist, was zählt eigentlich zu einem Thema?

Wenn eine Themenzeichenfolge, wie z. B. USA/Alabama/Auburn, in ein Thema konvertiert wird, wird das Thema zur Themenbaumstruktur hinzugefügt. Weitere Themenknoten und die entsprechenden The-

men werden in der Baumstruktur erstellt, falls erforderlich. Die Themenzeichenfolge USA/Alabama/Auburn wird in eine Baumstruktur mit drei Themen konvertiert.

- USA
- USA/Alabama
- USA/Alabama/Auburn

Um alle Themen in der Themenstruktur anzuzeigen, verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('#') TYPE(TOPIC)`.

Ein nicht verwendetes Thema in der Themenstruktur weist die folgenden Eigenschaften auf.

#### **Sie ist einem Themenobjekt nicht zugeordnet.**

Ein Verwaltungsthemenobjekt verfügt über eine Themenzeichenfolge, die sie einem Thema zuordnet. Falls beim Definieren des Themenobjekts Alabama das Thema USA/Alabama, dem es zugeordnet werden soll, nicht vorhanden ist, wird das Thema aus der Themenzeichenfolge erstellt. Wenn das Thema vorhanden ist, werden das Themenobjekt und das Thema unter Verwendung der Themenzeichenfolge verknüpft.

#### **Es hat keine ständige Veröffentlichung**

Ein Thema mit einer ständigen Veröffentlichung entsteht, wenn ein Publisher eine Nachricht mit der Option `MQPMO_RETAIN` in ein Thema einreicht.

Verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama') RETAINED`, um zu überprüfen, ob USA/Alabama über eine ständige Veröffentlichung verfügt. Die Antwort lautet YES oder NO.

Mit dem **runmqsc** -Befehl `CLEAR TOPICSTR('USA/Alabama') CLTRTYPE(RETAINED)` können Sie eine ständige Veröffentlichung aus USA/Alabama entfernen.

#### **Es hat keine untergeordneten Themen**

USA/Alabama/Auburn ist ein Thema ohne untergeordnete Themen. USA/Alabama/Auburn ist das direkt untergeordnete Thema von USA/Alabama.

Zeigen Sie die direkten untergeordneten Elemente von USA/Alabama mit dem **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama+')` an.

#### **Es sind keine aktiven Publisher für den Knoten vorhanden.**

Ein aktiver Publisher für einen Knoten ist eine Anwendung, die das Thema für die Ausgabe geöffnet hat.

Eine Anwendung öffnet beispielsweise das Themenobjekt mit dem Namen **Alabama** mit den Optionen zum Öffnen `MQOO_OUTPUT`.

Verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACT-CONN`, um aktive Publisher für USA/Alabama und alle untergeordneten Elemente anzuzeigen.

#### **Es sind keine aktiven Subskribenten für den Knoten vorhanden.**

Ein aktiver Subskribent kann entweder eine permanente Subskription oder eine Anwendung sein, die eine Subskription für ein Thema bei `MQSUB` registriert und nicht geschlossen hat.

Verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACT-CONN`, um aktive Subskriptionen für USA/Alabama anzuzeigen.

Verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACT-CONN`, um aktive Subskriptionen für USA/Alabama und alle untergeordneten Elemente anzuzeigen.

### **Anzahl der Themen in einer Themenstruktur verwalten**

Zusammenfassend gibt es eine Reihe von Möglichkeiten, die Anzahl der Themen in einer Themenstruktur zu verwalten.

### TPCOUNT anzeigen

Verwenden Sie den **runmqsc** -Befehl `DISPLAY PUBSUB ALL` regelmäßig, um die Eigenschaft **TPCOUNT** anzuzeigen. Dies ist die Anzahl der Themenknoten in der Themenstruktur. Wenn die Zahl größer wird, kann dies darauf hinweisen, dass ein kürzerer **TREELIFE** erforderlich ist, oder dass eine Neugestaltung der Themen selbst erforderlich ist.

### TREELIFE ändern

Ein nicht verwendetes Thema hat standardmäßig eine Lebensdauer von 30 Minuten. Sie können die Lebensdauer eines nicht verwendeten Themas kleiner machen.

Der Befehl **runmqsc**, `ALTER QMGR TREELIFE(900)`, reduziert beispielsweise die Lebensdauer eines nicht verwendeten Themas von 30 Minuten auf 15 Minuten.

### Ausnahmsweise Neustart des WS-Managers

Wenn der Warteschlangenmanager erneut gestartet wird, wird die Themenstruktur von Themenobjekten, Knoten mit ständigen Veröffentlichungen und permanenten Subskriptionen erneut initialisiert. Themen, die durch den Betrieb von Veröffentlichungs- und Subskribentenprogrammen erstellt wurden, werden eliminiert.

Als letztes Mittel, wenn das Wachstum in unerwünschten Themen die Ursache von Leistungsproblemen in der Vergangenheit ist, starten Sie den Warteschlangenmanager erneut.

### Zugehörige Konzepte

[Themenstrukturen](#)

V 9.1.4

MQ Adv.

Linux

MQ Adv. VUE

## Aspera gateway kann die Leistung in Netzen mit hoher Latenz verbessern

IBM Aspera fasp.io Gateway stellt einen schnellen TCP/IP-Tunnel bereit, der den Netzdurchsatz für IBM MQ erheblich erhöhen kann.

Mit dem Aspera gateway kann die Leistung von Warteschlangenmanagerkanälen verbessert werden. Es ist besonders effektiv, wenn das Netz eine längere Latenzzeit hat oder dazu neigt, Pakete zu verlieren, und wird normalerweise dazu verwendet, die Verbindung zwischen Warteschlangenmanagern in verschiedenen Rechenzentren zu beschleunigen.

Bei ohnehin schnellen Netzen, in denen keine Paketverluste auftreten, wird die Netzleistung durch das Aspera gateway jedoch eher beeinträchtigt. Sie sollten die Netzleistung daher unbedingt vor und nach der Definition einer Aspera gatewayverbindung überprüfen.

Ein Warteschlangenmanager, der auf einer beliebigen berechtigten CD -Plattform ausgeführt wird, kann über eine Aspera gatewayeine Verbindung herstellen. Das Gateway selbst wird unter Red Hat® oder Ubuntu Linux bereitgestellt.

Weitere Informationen finden Sie im Abschnitt [Aspera gateway-Verbindung unter Linux definieren](#).

## Bemerkungen

---

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

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieser Dokumentation ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe  
IBM Europe, Middle East and Africa  
Tour Descartes  
2, avenue Gambetta  
92066 Paris La Défense  
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Lizenzierung von geistigem Eigentum

IBM Japan, Ltd.

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in dieser Veröffentlichung werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East and Africa  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Die in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Um diese so realistisch wie möglich zu gestalten, enthalten sie auch Namen von Personen, Firmen, Marken und Produkten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musterprogramme, die in Quellensprache geschrieben sind. Sie dürfen diese Musterprogramme kostenlos (d. h. ohne Zahlung an IBM) kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

## Informationen zu Programmierschnittstellen

---

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

Dieses Handbuch enthält Informationen über vorgesehene Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zu schreiben, um die Services von WebSphere MQ zu erhalten.

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

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

## Marken

---

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