

7.5

Überwachung und Leistung für IBM Web-Sphere MQ

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 297 gelesen werden.

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

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

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

Inhaltsverzeichnis

Überwachung und Leistung.....	5
Ereignisüberwachung.....	5
Instrumentierungsereignisse.....	6
Durchsatzereignisse.....	21
Konfigurationsereignisse.....	38
Befehlsereignisse.....	42
Ereignisse der Protokollfunktion.....	44
Beispielprogramm zur Überwachung von Instrumentierungsereignissen.....	50
Nachrichtenüberwachung.....	57
Aktivitäten und Operationen.....	57
Nachrichtenroutetechniken.....	59
Aktivitätenaufzeichnung.....	61
Trace-Route-Messaging.....	66
IBM WebSphere MQ -Anwendung für die Routenanzeige.....	81
Aktivitätsberichtsreferenz.....	100
Trace-Route-Nachrichtenreferenz.....	127
Trace-Route-Antwortnachrichtenreferenz.....	138
Abrechnungs-und Statistiknachrichten.....	140
Abrechnungsnachrichten.....	141
Statistiknachrichten.....	145
Abrechnungs-und Statistikdaten anzeigen.....	150
Nachrichtenreferenz für Abrechnung und Statistik.....	156
Anwendungsaktivitätstrace.....	206
Traceinformationen zur Anwendungsaktivität erfassen.....	206
Beispielprogramm 'amqsact'.....	215
Trace-Nachrichtenreferenz für Anwendungsaktivitäten.....	217
Real-time Monitoring.....	282
Attribute zur Steuerung der Echtzeitüberwachung.....	283
Warteschlangen-und Kanalüberwachungsdaten anzeigen.....	285
Warteschlangen überwachen.....	286
Kanäle überwachen.....	289
Windows-Leistungsmonitor.....	296
Bemerkungen.....	297
Informationen zu Programmierschnittstellen.....	298
Marken.....	299

Überwachung und Leistung

In IBM WebSphere MQ stehen eine Reihe von Überwachungsverfahren zur Verfügung, um Statistikdaten und andere spezifische Informationen zur Ausführung Ihres Warteschlangenmanagernetzes abzurufen. Verwenden Sie die Überwachungsinformationen und Anleitungen in diesem Abschnitt, um die Leistung Ihres Warteschlangenmanagernetzes zu verbessern.

Je nach Größe und Komplexität Ihres WS-Manager-Netzes können Sie eine Reihe von Informationen aus der Überwachung Ihres WS-Manager-Netzes abrufen. 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.

Zugehörige Tasks

[Konfiguration](#)

[WebSphere MQ verwalten](#)

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 WebSphere 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

[Ereignisnachrichtenreferenz](#)

„Ereignistypen“ auf Seite 8

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

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

Abbildung 1 auf Seite 7 veranschaulicht das Konzept der Instrumentierungsereignisse.

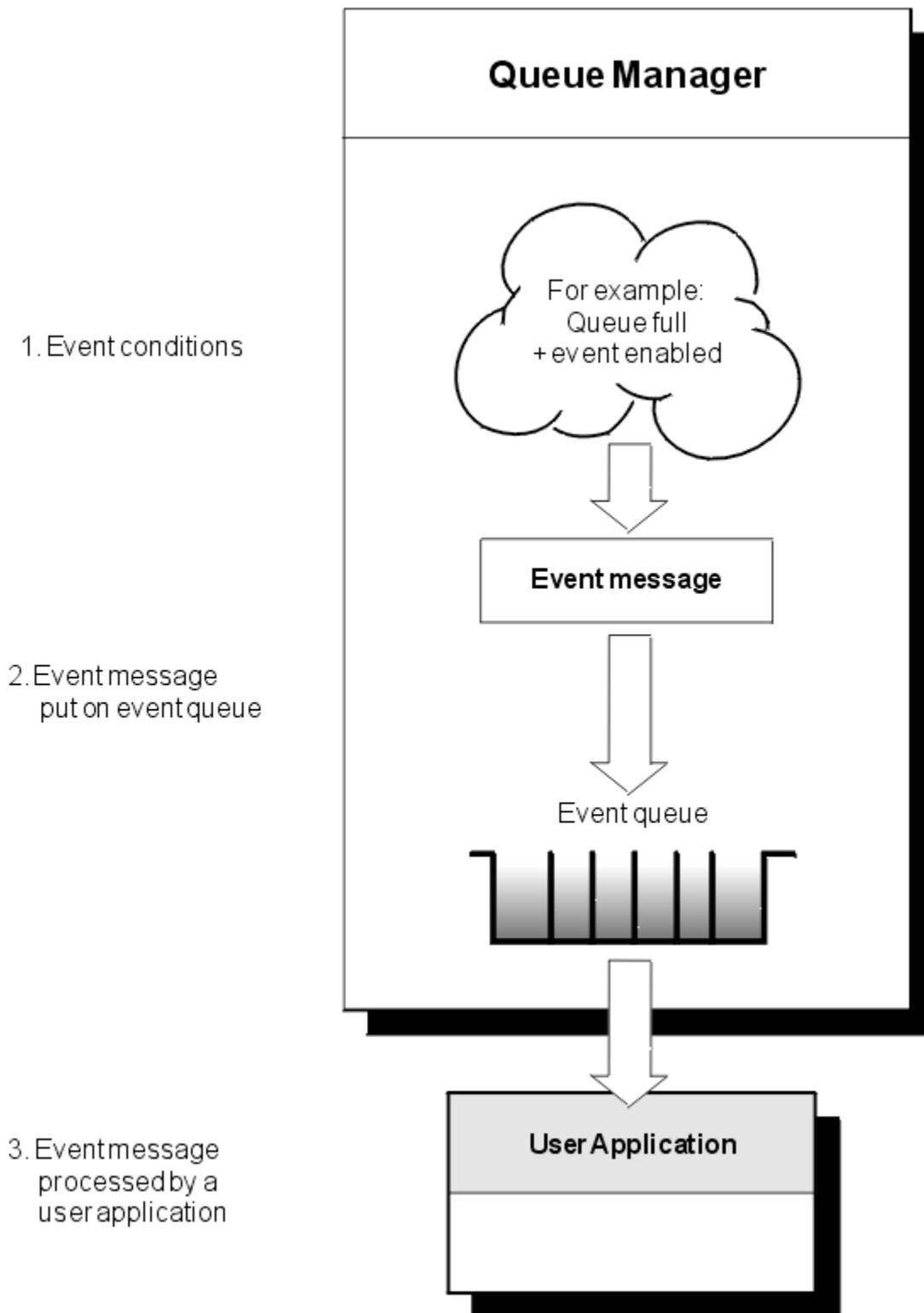


Abbildung 1. Informationen zu Instrumentierungs

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 von einem z/OS -Warteschlangenmanager überwachen, stellen Sie sicher, dass Sie EBCDIC in ASCII konvertieren.

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 unter IBM WebSphere MQ für IBM i, Windows, UNIX and Linux® -Systeme 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 Seite 50

Auf dieser Seite können Sie ein C-Beispielprogramm zur Überwachung von Instrumentierungsereignissen anzeigen.

Ereignistypen

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

IBM WebSphere MQ -Instrumentierungsereignisse haben die folgenden Typen:

- 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, auf vielen verschiedenen Knoten und für mehrere IBM WebSphere MQ -Anwendungen überwachen. Sie können insbesondere alle Knoten in Ihrem System von einem einzigen Knoten aus überwachen (für die Knoten, die IBM WebSphere MQ -Ereignisse unterstützen), wie in [Abbildung 2 auf Seite 10](#) dargestellt.

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

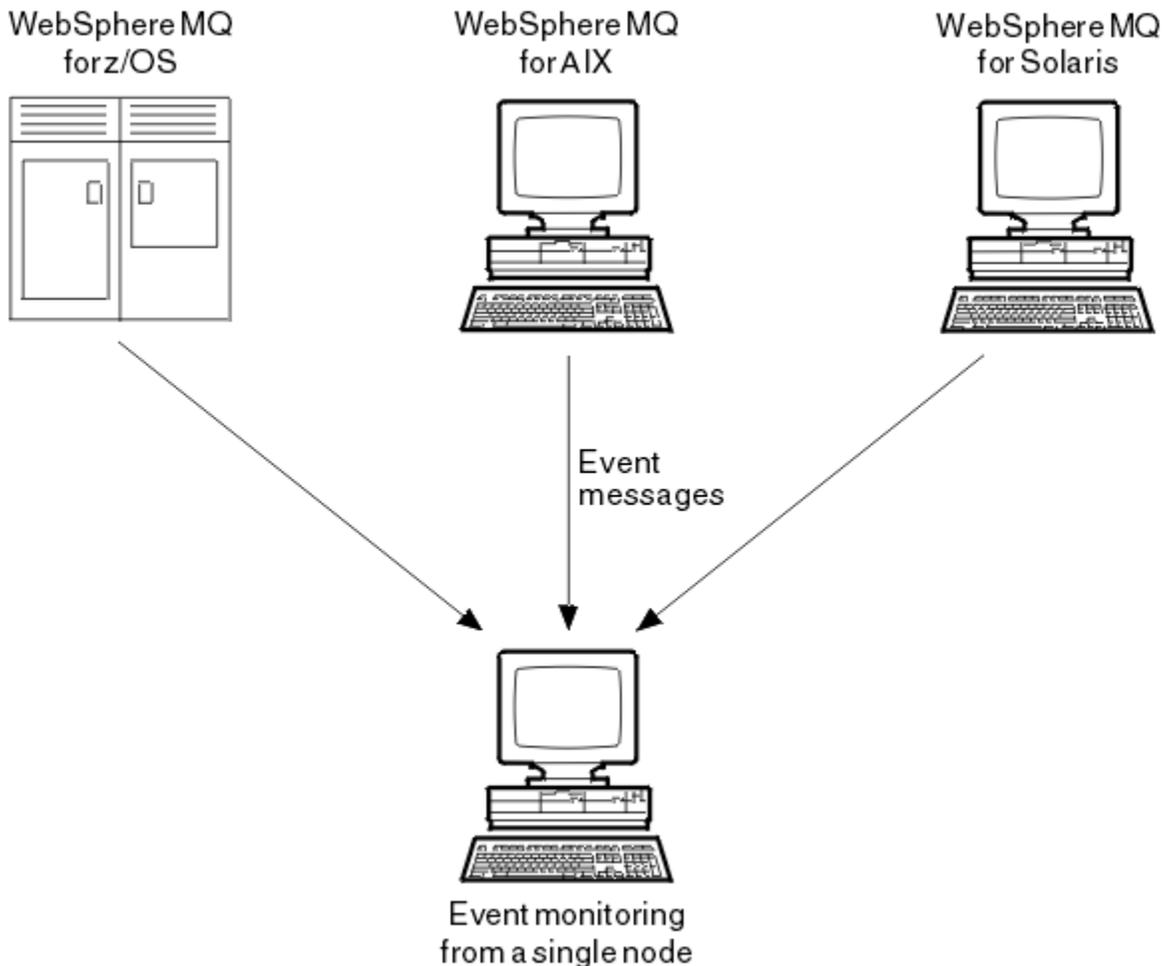


Abbildung 2. Warteschlangenmanager auf verschiedenen Plattformen überwachen, auf einem einzelnen Knoten

Instrumentierungsereignisse ermöglichen auch Anwendungen, die als Agenten für andere Verwaltungenetzwerke fungieren, z. B. Tivoli NetView für z/OS, Berichte zu überwachen und die entsprechenden Alerts 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 sind Bedingungen, die ein Warteschlangenmanagerereignis auslösen 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.

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

Berechtigung (nur auf Windows-und UNIX -Systemen)

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 Windowsund UNIX -Systemen 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

Lokale Ereignisse weisen darauf hin, dass eine Anwendung (oder der Warteschlangenmanager) nicht in der Lage war, auf eine lokale Warteschlange oder ein anderes lokales Objekt zuzugreifen. Eine Anwendung kann beispielsweise versuchen, auf ein Objekt zuzugreifen, das nicht definiert wurde. Die lokale Ereignisnachricht kann die folgenden Ereignisdaten enthalten:

- Fehler im Typ der Aliasbasiswarteschlange
- Unbekannte Aliasbasiswarteschlange
- Unbekannter Objektname

Fern

Ferne Ereignisse geben an, dass eine Anwendung oder der Warteschlangenmanager nicht auf eine ferne Warteschlange eines anderen Warteschlangenmanagers zugreifen kann. Die zu verwendende Übertragungswarteschlange ist beispielsweise möglicherweise nicht korrekt definiert. 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 Brückenergebnisse

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 Brückenergebnisse 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

SSL-Ereignisse

Das einzige Secure Sockets Layer-Ereignis (SSL oder TLS) ist das Ereignis "Channel SSL Error". Dieses Ereignis wird zurückgegeben, wenn ein Kanal, der SSL oder TLS verwendet, keine SSL-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 Warteschlangewartung'

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.

Konfigurationsereignisse

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

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

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

Es gibt vier Typen von Konfigurationsereignisereignis:

Objekt ereignisse erstellen

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

Objekt ereignisse ändern

Änderungsobjekt ereignisse werden generiert, wenn ein Objekt geändert wird. Die Ereignisnachricht enthält die folgenden Ereignisdaten: [Objekt ändern](#) .

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: [Objekt aktualisieren](#) .

Befehlsereignisse

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

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

Die Ereignisnachrichten für Befehlsereignisse werden in die Warteschlange SYSTEM.ADMIN.COM-MAND.EVENT gestellt.

Befehlsereignisse enthalten die folgenden Ereignisdaten: [Befehl](#)

Ereignisse der Protokollfunktion

Protokollierungsereignisse werden dokumentiert, wenn ein Warteschlangenmanager, der die lineare Protokollierung verwendet, mit dem Schreiben von Protokolldatensätzen in einen neuen Protokollspeicherbereich beginnt.

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: [Protokollfunktion](#)

Zusammenfassung der Ereignisnachrichtendaten

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

Ereignistyp	Sehen Sie sich diese Themen
Berechtigungseignisse	Nicht berechtigt (Typ 1)
	Nicht berechtigt (Typ 2)
	Nicht berechtigt (Typ 3)
	Nicht berechtigt (Typ 4)
	Nicht berechtigt (Typ 5)
	Nicht berechtigt (Typ 6)

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

Ereignistyp	Sehen Sie sich diese Themen
Ferne Ereignisse	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
SSL-Ereignisse	Kanal-SSL-Fehler
Start- und Stoppereignisse	Warteschlangenmanager aktiv
	Warteschlangenmanager nicht aktiv

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 WebSphere MQ -Scriptbefehle (MQSC).
- Die entsprechenden IBM WebSphere MQ -PCF-Befehle
- IBM WebSphere 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.

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 Warteschlangenmanagerereignissen den MQSC-Befehl ALTER QMGR und geben Sie das entsprechende WS-Manager-Attribut an. Tabelle 1 auf Seite 17 In ist zusammengefasst, wie Warteschlangenmanagerereignisse aktiviert werden. Wenn Sie ein WS-Manager-Ereignis inaktivieren möchten, setzen Sie den entsprechenden Parameter auf DISABLED .

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

Kanal-und Brückenergebnisse 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 .

Um Kanalereignisse zu aktivieren oder zu inaktivieren, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie dabei das entsprechende WS-Manager-Attribut an. Tabelle 2 auf Seite 17 fasst zusammen, wie Sie Kanal-und Brückenergebnisse aktivieren. Wenn Sie ein WS-Manager-Ereignis inaktivieren möchten, setzen Sie den entsprechenden Parameter auf DISABLED .

Ereignis	ALTER QMGR, Parameter
Kanal	CHLEV (ENABLED)
Nur bei Kanalfehlern	CHLEV (EXCEPTION)
IMS BRIDGE	BRIDGEEV (ENABLED)
SSL	SSLEV (ENABLED)
Automatische Kanaldefinition	CHADEV (ENABLED)

Wenn CHLEV auf eine Ausnahmeregung 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 Warteschlangewartung'

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 beliebiges 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 als lokale Warteschlangen, Aliaswarteschlangen oder als lokale Definitionen ferner Warteschlangen definieren. 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.

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 erforder-

derlich 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 WebSphere MQ -Nachrichten besteht eine Ereignisnachricht aus zwei Teilen: einem Nachrichtendeskriptor und den Nachrichtendaten.

[Bedingungen für ein Auslöserereignis](#)

Zugehörige Verweise

[QSGDisp \(MQLONG\)](#)

Format von Ereignisnachrichten

Ereignisnachrichten enthalten Informationen zu einem Ereignis und dessen Ursache. Wie andere WebSphere MQ -Nachrichten besteht 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 103](#)

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 107](#)

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

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

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 \(PCF-Header\)](#)

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](#)

Beispiele, die die Verwendung von Leistungsereignissen veranschaulichen, setzen voraus, dass Sie Warteschlangenattribute mit den entsprechenden IBM WebSphere MQ -Befehlen (MQSC) festlegen. Unter können Sie Warteschlangenattribute auch über die 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 21](#) 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.

Parameter	Beschreibung
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 eingereichten Nachrichten (die Anzahl der MQPUT-Aufrufe an die Warteschlange), da die Statistikdaten zum letzten Mal zurückgesetzt wurden.

Tabelle 3. Performance-Ereignisstatistik (Forts.)	
Parameter	Beschreibung
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.

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](#)

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 *nicht* in gemeinsam genutzten Warteschlangen 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.

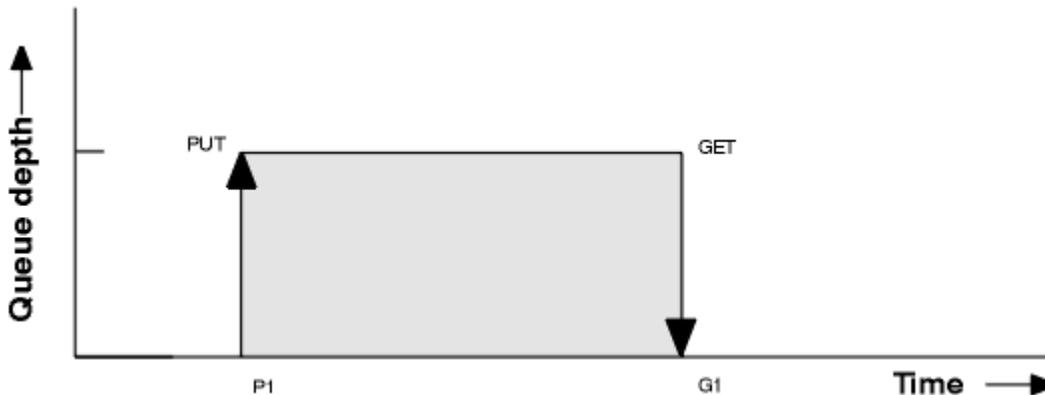


Abbildung 3. Informationen zum Warteschlangenserviceintervall

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)

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 Warteschlangenserviceintervall OK Keine Ereignisse des Warteschlangenserviceinterv	QSVCIHV (HIGH) QSVCIHV (OK) QSVCIHV (NONE)
Serviceintervall	QSVCIHV (<i>tt</i>) , wobei <i>tt</i> der Service ist Intervallzeit in Millisekunden.

Führen Sie die folgenden Schritte aus, um Warteschlangenserviceintervallereignisse zu aktivieren

Vorgehensweise

1. Setzen Sie das Warteschlangenmanager-Attribut PERFMHV auf ENABLED.
Leistungsereignisse werden auf dem Warteschlangenmanager aktiviert.
2. Setzen Sie das Steuerattribut QSVCIHV für ein Warteschlangenserviceintervall-Hoch- oder OK-Ereignis in der Warteschlange nach Bedarf.

3. Legen Sie das Attribut QSVCIINT 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') QSVCIINT(10000) QSVCIIEV(HIGH)
```

Beispiele für Warteschlangenserviceintervall

Verwenden Sie diese Beispiele, um die Informationen zu verstehen, die Sie aus Warteschlangenserviceintervallereignissen abrufen können.

Die drei Beispiele zeigen zunehmend komplexere Darstellungen der Verwendung von Warteschlangenserviceintervallereignissen.

Die Abbildungen, die den Beispielen beigelegt sind, 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.

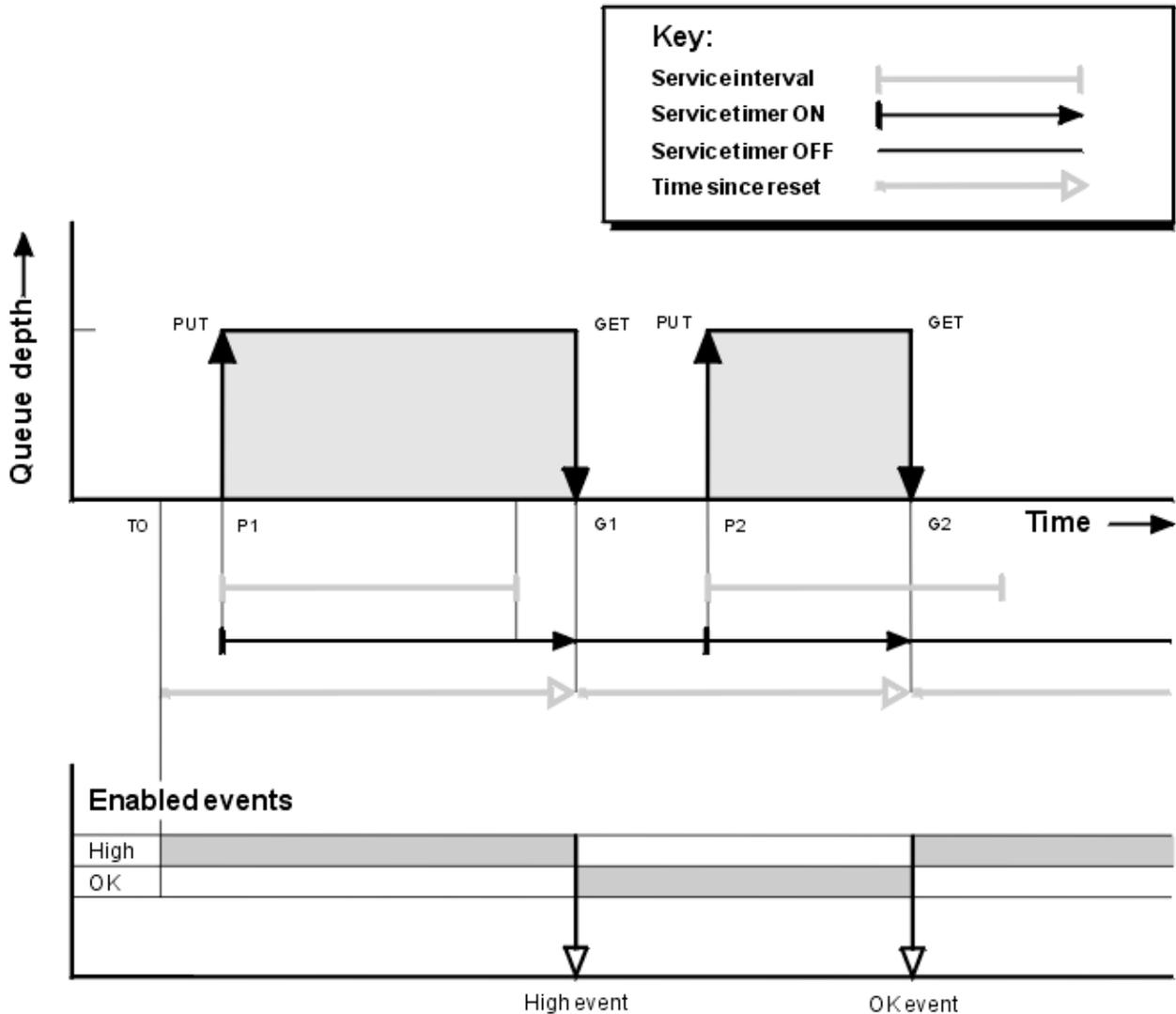


Abbildung 4. Ereignisse des Warteschlangenserviceintervalls-Beispiel 1

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 'Warteschlangenserviceintervall 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 28](#) sind die Ereignisstatistikdaten für dieses Beispiel zusammengefasst.

<i>Tabelle 5. Ereignisstatistikzusammenfassung für Beispiel 1</i>		
	Ereignis 1	Ereignis 2
Zeitpunkt des Ereignisses	T (G1)	T (G2)
Typ des Ereignisses	Hoch	OK
Zurücksetzungszeit (TimeS)	T (G1)-T (0)	T (G2)-T (G1)
HighQDepth	1	1
MsgEnqCount	1	1
MsgDeqCount	1	1

Der mittlere Teil von [Abbildung 4 auf Seite 27](#) 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.

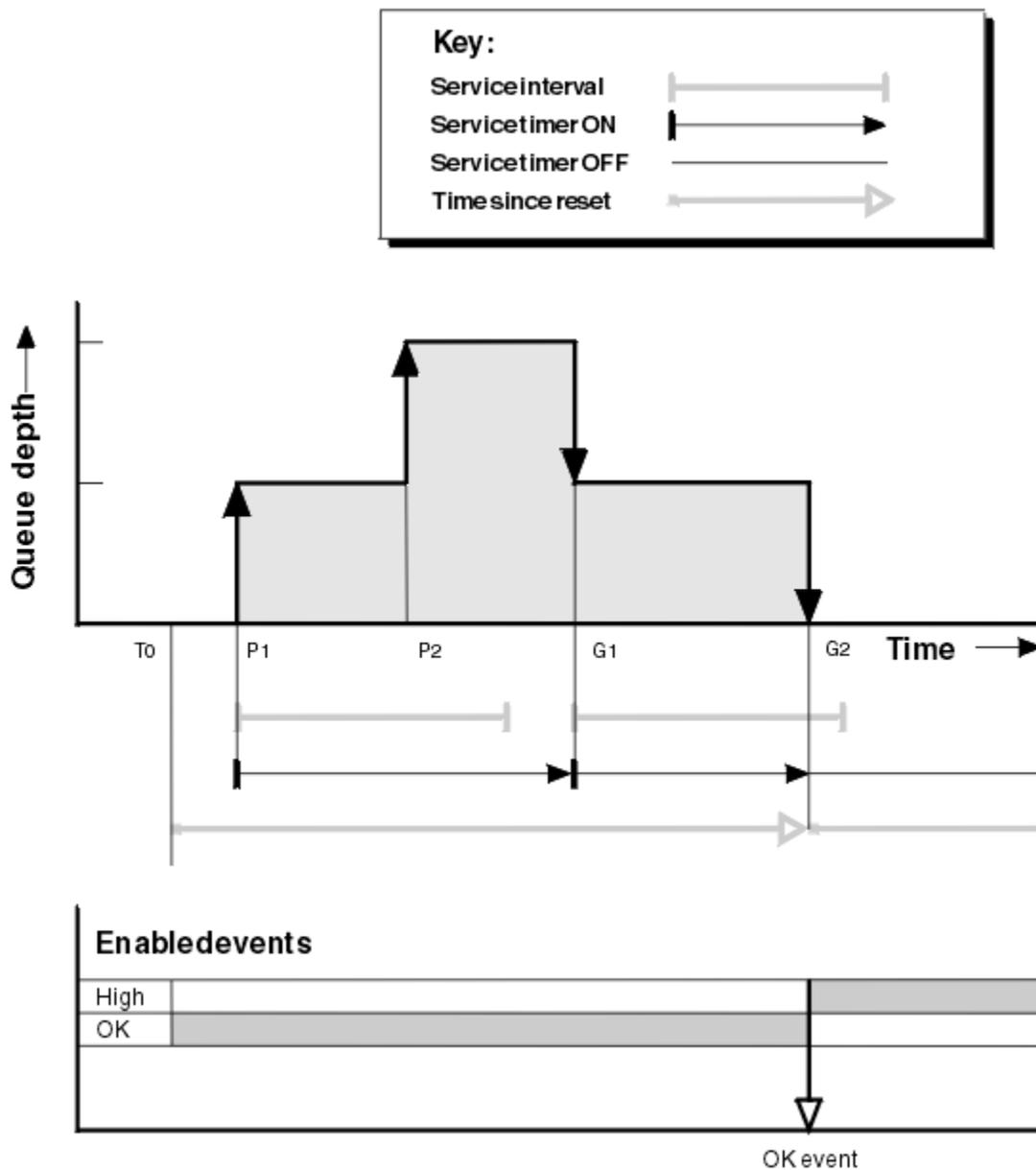


Abbildung 5. Ereignisse des Warteschlangenserviceintervalls-Beispiel 2

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 30 sind die Ereignisstatistikdaten für dieses Beispiel zusammengefasst.

Tabelle 6. Ereignisstatistikzusammenfassung für Beispiel 2	
	Ereignis 2
Zeitpunkt des Ereignisses	T (G2)
Typ des Ereignisses	OK
Zurücksetzungs-zeit (TimeS)	T (G2)-T (0)
HighQDepth	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.

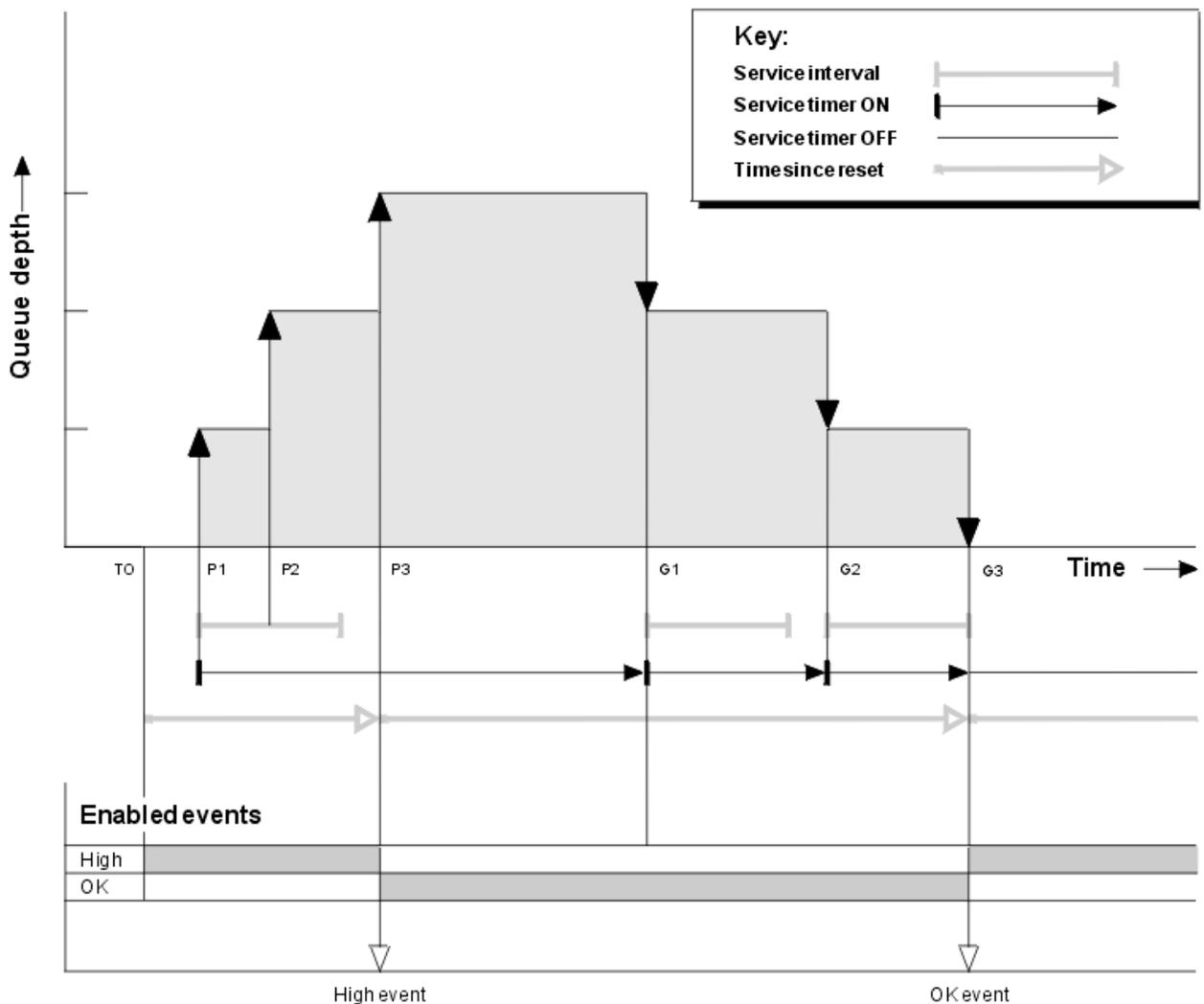


Abbildung 6. Ereignisse des Warteschlangenserviceintervalls-Beispiel 3

Commentary

1. Zum Zeitpunkt T (0) werden die Warteschlangenstatistikdaten zurückgesetzt und das Warteschlangen-serviceintervall 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 die Warteschlange durch den dritten Eintrag geleert, und der Servicezeitgeber ist *gleich* für das 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 31 sind die Ereignisstatistikdaten für dieses Beispiel zusammengefasst.

	Ereignis 1	Ereignis 2
Zeitpunkt des Ereignisses	T (P3)	T (G3)
Typ des Ereignisses	Hoch	OK
Zurücksetzungs-zeit (TimeS)	T (P3)-T (0)	T (G3)-T (P3)
HighQDepth	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 WebSphere 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 WebSphere MQ die folgenden Warteschlangenlängenergebnisse 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 34

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 "Warteschlangenlänge hoch" wird generiert, wenn eine Nachricht in die Warteschlange gestellt wird, wodurch die Warteschlangenlänge größer oder gleich dem Wert ist, der durch die Höchstgrenze für Warteschlangenlänge bestimmt wird.
 - 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 vom Typ "Warteschlangenlänge niedrig" wird generiert, wenn eine Nachricht aus einer Warteschlange durch eine get-Operation entfernt wird, wodurch die Warteschlangenlänge kleiner oder gleich dem Wert ist, der durch die Untergrenze Warteschlangenlänge bestimmt wird.
 - 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 PERFMV.
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:

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 Grenzwert für die Warteschlangenlänge festgelegt wird.

IBM WebSphere MQ generiert die Nachricht mit dem niedrigen Ereignis 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 keine Ereignisse mit hoher Warteschlangenlänge und ein niedriges Ereignis der Warteschlangenlänge zurückgesetzt.

Beispiel

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

```
ALTER QMGR PERFMV(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)
```

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

```
ALTER QMGR PERFMEV(ENABLED)

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

Zugehörige Konzepte

[„Ereignisse der Warteschlangenlänge“](#) auf Seite 31

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

Zugehörige Tasks

[„Warteschlangenlänge aktivieren“](#) auf Seite 32

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 35 zeigt die Variation der Warteschlangenlänge im Verlauf der Zeit an.

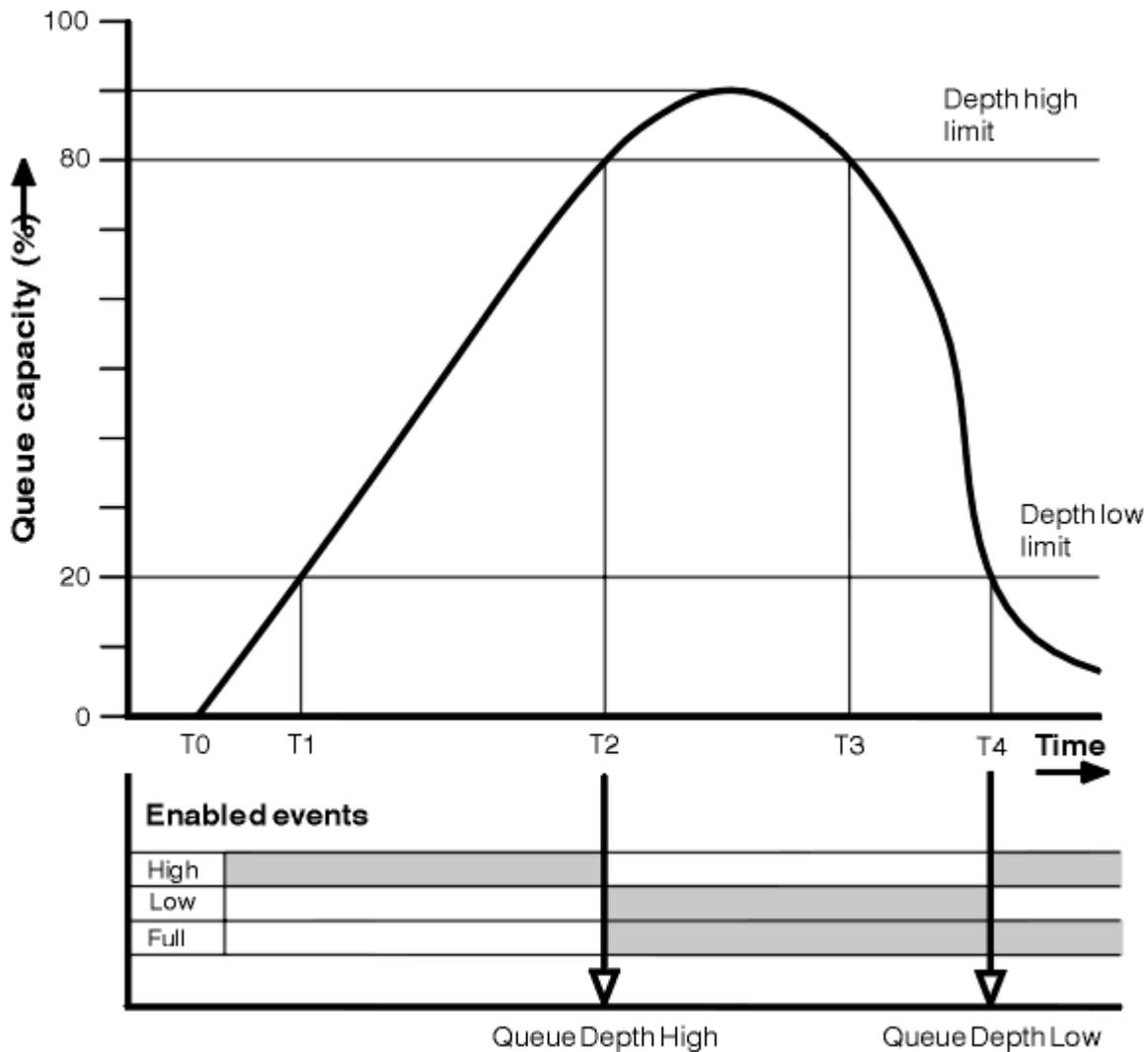


Abbildung 7. Ereignisse der Warteschlangenlänge (1)

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 36 sind die Warteschlangenereignisstatistik zusammengefasst, und [Tabelle 9](#) auf Seite 36 fasst zusammen, welche Ereignisse aktiviert sind.

<i>Tabelle 8. Ereignisstatistikzusammenfassung für Warteschlangentiefenereignisse (Beispiel 1)</i>		
	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

<i>Tabelle 9. Zusammenfassung, die anzeigt, welche Ereignisse aktiviert sind</i>			
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 37](#) zeigt die Variation der Warteschlangenlänge im Verlauf der Zeit an.

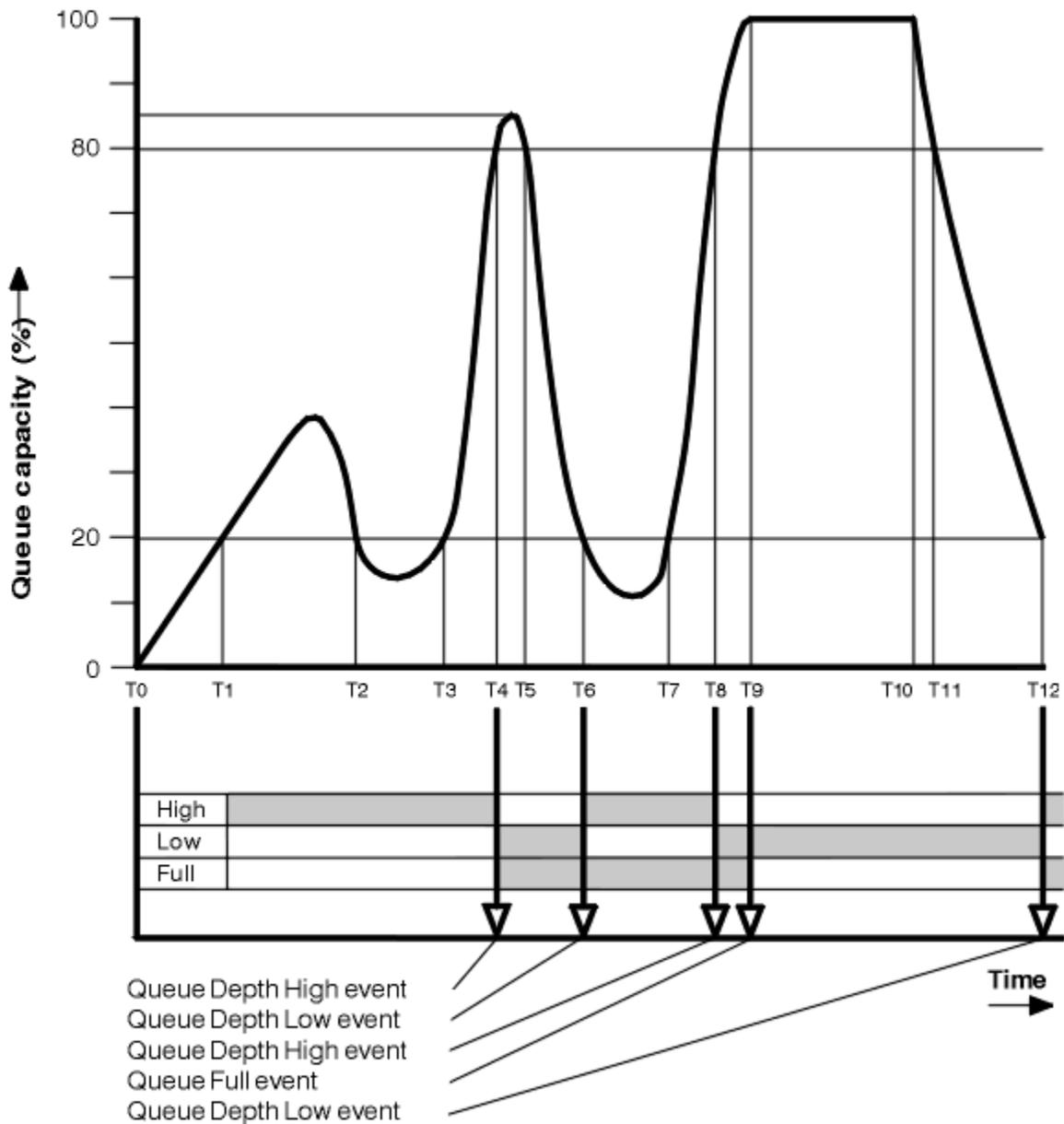


Abbildung 8. Ereignisse der Warteschlangenlänge (2)

Commentary

- Es wird zu den folgenden Zeiten kein Ereignis "Queue Depth Low" generiert:
 - T (1) (Warteschlangenlänge steigt und ist nicht aktiviert)
 - T (2) (Nicht aktiviert)
 - T (3) (Warteschlangenlänge wird erhöht, und nicht aktiviert)
- Bei T (4) tritt ein Ereignis "Queue Depth High" (Warteschlangenlänge hoch) auf. Auf diese Weise werden sowohl die Warteschlangenlänge als auch die Warteschlangenlänge niedrig.
- Bei T (9) tritt ein Ereignis "Warteschlange voll" **nach** der ersten Nachricht auf, die nicht in die Warteschlange gestellt werden kann, da die Warteschlange voll ist.
- Bei T (12) tritt ein Ereignis "Queue Depth Low" (Warteschlangenlänge niedrig) auf

Zusammenfassung der Ereignisstatistik

Tabelle 10 auf Seite 38 fasst die Warteschlangenereignisstatistik zusammen und Tabelle 11 auf Seite 38 fasst zusammen, welche Ereignisse für dieses Beispiel zu unterschiedlichen Zeiten aktiviert sind.

Tabelle 10. Ereignisstatistikzusammenfassung für Warteschlangentiefenereignisse (Beispiel 2)

	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)
HighQDepth	800	855	800	1.000	1.000
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.

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.

Im Falle von Änderungsobjektereignissen werden zwei Nachrichten generiert, eine mit den Informationen vor der Änderung, die andere mit den Informationen nach.

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

Einführung in Programmierbare Befehlsformate

„Konfigurationsereignisse“ auf Seite 38

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

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 Befehlereignisse ebenfalls aktiviert sind. Wenn ein MQSC- oder PCF-Befehl bewirkt, dass ein Konfigurationsereignis und ein Befehlereignis 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 38](#)

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](#)

Ereigniskonfigurationsereignis aktualisieren

Das Konfigurationsereignis für die Aktualisierungsobjektconfiguration 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 38](#)

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. Auf verteilten Plattformen ist das Befehlsdatenformat beispielsweise immer PCF, selbst wenn die ursprüngliche Anforderung als MQSC-Befehl übergeben wurde.

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.

Wenn 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 42

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

Zugehörige Verweise

REFRESH 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 42](#)

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

[„Befehlsereignisgenerierung“ auf Seite 43](#)

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

Protokollfunktionsergebnisse sind Benachrichtigungen, die ein Warteschlangenmanager in einen neuen Protokollspeicherbereich geschrieben hat.

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.

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.

Generierung von Protokollfunktionseignis

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

In den folgenden Situationen wird eine Protokollfunktionseignisnachricht 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 Journalempfänger 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.

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

Wenn Protokollfunktionseignisse 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 Protokollfunktionseignis 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 44](#)

Protokollfunktionseignisse sind Benachrichtigungen, die ein Warteschlangenmanager in einen neuen Protokollspeicherbereich geschrieben hat.

Zugehörige Verweise

[LoggerEvent \(MQLONG\)](#)

[RESET QMGR](#)

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 Protokoll dateien 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 Protokollfunktionseignisnachricht 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 44](#)

Protokollfunktionseignisse sind Benachrichtigungen, die ein Warteschlangenmanager in einen neuen Protokollspeicherbereich geschrieben hat.

Zugehörige Verweise

[LoggerEvent \(MQLONG\)](#)

„Generierung von Protokollfunktionsereignis“ auf Seite 45

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 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 output
/* a message to stdout when a logger event occurs
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (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.
/* <NOC_COPYRIGHT>
/* **** */
/*
/* Function: AMQSLOG is a sample program which monitors the logger event
/* queue for new event messages, reads those messages, and puts the 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_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"};

/* **** */
/* 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;
    PMQCHAR   pBuffer = NULL;

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

    /*****
    */
    /* Parse any command line options
    */
    /*****
    */

    if (argc > 1)
        stncpy(QMName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);

    pBuffer = (char *)malloc(MAX_MESSAGE_LENGTH);
    if (!pBuffer)
    {
        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 != MQCC_OK)
    {
        printf("Error in call to MQCONN, Reason %d, CompCode %d\n", Reason,
              CompCode);
        goto MOD_EXIT;
    }

    /* Open the logger event queue for input */

    stncpy(ObjDesc.ObjectQMgrName,QMName, MQ_Q_MGR_NAME_LENGTH);
    stncpy(ObjDesc.ObjectName, LogEvQ, MQ_Q_NAME_LENGTH);

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

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

        goto MOD_EXIT;
    }
    else
    {
        ProcessPCF(hConn, hEventQueue, pBuffer);
    }

    MOD_EXIT:

    if (pBuffer != NULL) {

```

```

    free(pBuffer);
}

/*****
/* 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 msg*/
/*
/* Output Parameters: None */
/*
/* Logic: Wait for messages to appear on the logger event queue and display */
/* their 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 Reason = 0;
    MQLONG CompCode;
    MQLONG MsgLen;
    PMQCHAR Parm = NULL;

    Gmo.Options |= MQGMO_WAIT;
    Gmo.Options |= MQGMO_CONVERT;
    Gmo.WaitInterval = MQWI_UNLIMITED;
    /*****
    /* Process response Queue */
    *****/
    while (Reason == MQCC_OK)
    {
        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 != MQCC_OK)
        {
            switch(Reason)
            {
                case MQRC_NO_MSG_AVAILABLE:
                    printf("Timed out");
                    break;

                default:
                    printf("MQGET failed RC(%d)\n", Reason);
                    break;
            }
            goto MOD_EXIT;
        }

        /*****
        /* Only expect PCF event messages on this queue */
        *****/
        if (memcmp(Mqmd.Format, MQFMT_EVENT, sizeof(Mqmd.Format)))
        {

```

```

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)
{
printf("-----\n");
printf("Event Message Received\n");

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

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

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

pPCFCmd = (char *) (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("%-32d",pCfst -> Parameter);
}

fwrite( pCfst -> String, pCfst -> StringLength, 1, stdout);
pPCFCmd += pCfst -> StrucLength;
break;
default:
printf("Unrecognised datatype %d returned\n",pCfst->Type);
goto MOD_EXIT;
}
putchar('\n');
}
printf("-----\n");
}
MOD_EXIT:
return;
}

/*****
/* Function: ParmToString */
*****/
/* Input Parameters: Parameter for which to get string description */
/* Output Parameters: None */
/* Logic: Takes a parameter as input and returns a pointer to a string */
/* description for that parameter, or NULL if the parameter does not */
/* have an associated string description */

```

```

/*****/
static PMQCHAR ParmToString(MQLONG Parameter){
    long i;
    for (i=0 ; i< sizeof(ParmTable)/sizeof(ParmTableEntry); i++)
    {
        if (ParmTable[i].ConstVal == Parameter ParmTable[i].Desc)
            return ParmTable[i].Desc;
    }
    return NULL;
}

```

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 45](#)

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

[„Befehlereignisverwendung“ auf Seite 44](#)

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

Zugehörige Verweise

[„Generierung von Protokollfunktionsereignis“ auf Seite 45](#)

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 zur Überwachung von Instrumentierungsereignissen

Auf dieser Seite können Sie ein C-Beispielprogramm zur Überwachung von Instrumentierungsereignissen anzeigen.

Dieses Beispielprogramm ist nicht Teil eines IBM WebSphere MQ -Produkts und wird daher nicht als physisches Element bereitgestellt. Das Beispiel ist insofern unvollständig, als es nicht alle möglichen Ergebnisse der angegebenen Aktionen auflistet. Sie können dieses Beispiel, insbesondere die PCF-Formate der Ereignisnachrichten, aber als Grundlage für eigene Programme verwenden, die Ereignisse ausgeben. Das Programm muss vor der Ausführung auf eigenen Systemen entsprechend angepasst werden.

```

/*****/
/*
/* Program name: EVMON
/*
/* Description: C program that acts as an event monitor
/*
/*
/*

```

```

/*****
/*
/* Function:
/*
/*
/* EVMON is a C program that acts as an event monitor - reads an
/* event queue and tells you if anything appears on it
/*
/* Its first parameter is the queue manager name, the second is
/* the event queue name. If these are not supplied it uses the
/* defaults.
/*
/*****
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#ifndef min
#define min(a,b) ((a) < (b)) ? (a) : (b)
#endif

/*****
/* includes for MQI
/*****
#include <cmqc.h>
#include <cmqcfh.h>
void printfmqcfst(MQCFST* pmqcfst);
void printfmqcfm(MQCFIN* pmqcfst);
void printreas(MQLONG reason);

#define PRINTREAS(param) \
    case param: \
        printf("Reason = %s\n",#param); \
        break;

/*****
/* global variable
/*****
MQCFH *evtmsg; /* evtmsg message buffer

int main(int argc, char **argv)
{
/*****
/* declare variables
/*****
int i; /* auxiliary counter
/*****
/* Declare MQI structures needed
/*****
MQOD od = {MQOD_DEFAULT}; /* Object Descriptor
MQMD md = {MQMD_DEFAULT}; /* Message Descriptor
MQGMO gmo = {MQGMO_DEFAULT}; /* get message options
/*****
/* note, uses defaults where it can
/*****

MQHCONN Hcon; /* connection handle
MQHOBJ Hobj; /* object handle
MQLONG O_options; /* MQOPEN options
MQLONG C_options; /* MQCLOSE options
MQLONG CompCode; /* completion code
MQLONG OpenCode; /* MQOPEN completion code
MQLONG Reason; /* reason code
MQLONG CReason; /* reason code for MQCONN
MQLONG buflen; /* buffer length
MQLONG evtmsglen; /* message length received
MQCHAR command[1100]; /* call command string ...
MQCHAR p1[600]; /* ApplId insert
MQCHAR p2[900]; /* evtmsg insert
MQCHAR p3[600]; /* Environment insert
MQLONG mytype; /* saved application type
char QMName[50]; /* queue manager name
MQCFST *paras; /* the parameters
int counter; /* loop counter
time_t ltime;

/*****
/* Connect to queue manager
/*****
QMName[0] = 0; /* default queue manager

```

```

if (argc > 1)
    strcpy(QMName, argv[1]);
MQCONN(QMName,
        &Hcon,
        &CompCode,
        &CReason);
/* queue manager
/* connection handle
/* completion code
/* reason code

/*****
/* Initialize object descriptor for subject queue
/*****
strcpy(od.ObjectName, "SYSTEM.ADMIN.QMGR.EVENT");
if (argc > 2)
    strcpy(od.ObjectName, argv[2]);

/*****
/* Open the event queue for input; exclusive or shared. Use of
/* the queue is controlled by the queue definition here
/*****

O_options = MQOO_INPUT_AS_Q_DEF /* open queue for input
+ MQOO_FAIL_IF_QUIESCING /* but not if qmgr stopping
+ MQOO_BROWSE;
MQOPEN(Hcon,
        &od,
        O_options,
        &Hobj,
        &CompCode,
        &Reason);
/* connection handle
/* object descriptor for queue
/* open options
/* object handle
/* completion code
/* reason code

/*****
/* Get messages from the message queue
/*****
while (CompCode != MQCC_FAILED)
{
/*****
/* I don't know how big this message is so just get the
/* descriptor first
/*****
gmo.Options = MQGMO_WAIT + MQGMO_LOCK
+ MQGMO_BROWSE_FIRST + MQGMO_ACCEPT_TRUNCATED_MSG;
/* wait for new messages
gmo.WaitInterval = MQWI_UNLIMITED; /* no time limit
buflen = 0; /* amount of message to get

/*****
/* clear selectors to get messages in sequence
/*****
memcpy(md.MsgId, MQMI_NONE, sizeof(md.MsgId));
memcpy(md.CorrelId, MQCI_NONE, sizeof(md.CorrelId));

/*****
/* wait for event message
/*****
printf("...>\n");
MQGET(Hcon,
        Hobj,
        &md,
        &gmo,
        buflen,
        evtmsg,
        &evtmsglen,
        &CompCode,
        &Reason);
/* connection handle
/* object handle
/* message descriptor
/* get message options
/* buffer length
/* evtmsg message buffer
/* message length
/* completion code
/* reason code

/*****
/* report reason, if any
/*****
if (Reason != MQRC_NONE && Reason != MQRC_TRUNCATED_MSG_ACCEPTED)
{
    printf("MQGET ==> %ld\n", Reason);
}
else
{
    gmo.Options = MQGMO_NO_WAIT + MQGMO_MSG_UNDER_CURSOR;
    buflen = evtmsglen; /* amount of message to get
    evtmsg = malloc(buflen);
    if (evtmsg != NULL)
    {

```

```

/*****
/* clear selectors to get messages in sequence */
/*****
memcpy(md.MsgId, MQMI_NONE, sizeof(md.MsgId));
memcpy(md.CorrelId, MQCI_NONE, sizeof(md.CorrelId));

/*****
/* get the event message */
/*****
printf("...>\n");
MQGET(Hcon, /* connection handle */
      Hobj, /* object handle */
      &md, /* message descriptor */
      &gmo, /* get message options */
      buflen, /* buffer length */
      evtmsg, /* evtmsg message buffer */
      &evtmsglen, /* message length */
      &CompCode, /* completion code */
      &Reason); /* reason code */

/*****
/* report reason, if any */
/*****
if (Reason != MQRC_NONE)
{
    printf("MQGET ==> %ld\n", Reason);
}
}
else
{
    CompCode = MQCC_FAILED;
}
}

/*****
/* . . . process each message received */
/*****

if (CompCode != MQCC_FAILED)
{
    /*****
    /* announce a message */
    /*****
    printf("\a\a\a\a\a");
    time(&lttime);
    printf(ctime(&lttime));

    if (evtmsglen != buflen)
        printf("DataLength = %ld?\n", evtmsglen);
    else
    {
        /*****
        /* right let's look at the data */
        /*****
        if (evtmsg->Type != MQCFT_EVENT)
        {
            printf("Something's wrong this isn't an event message,"
                  " its type is %ld\n", evtmsg->Type);
        }
        else
        {
            if (evtmsg->Command == MQCMD_Q_MGR_EVENT)
            {
                printf("Queue Manager event: ");
            }
            else
            if (evtmsg->Command == MQCMD_CHANNEL_EVENT)
            {
                printf("Channel event: ");
            }
            else
            :

            {
                printf("Unknown Event message, %ld.",
                      evtmsg->Command);
            }
        }

        if (evtmsg->CompCode == MQCC_OK)
            printf("CompCode(OK)\n");
    }
}

```

```

else if (evtmsg->CompCode == MQCC_WARNING)
    printf("CompCode(WARNING)\n");
else if (evtmsg->CompCode == MQCC_FAILED)
    printf("CompCode(FAILED)\n");
else
    printf("* CompCode wrong * (%ld)\n",
        evtmsg->CompCode);

if (evtmsg->StrucLength != MQCFH_STRUC_LENGTH)
{
    printf("it's the wrong length, %ld\n",evtmsg->StrucLength);
}

if (evtmsg->Version != MQCFH_VERSION_1)
{
    printf("it's the wrong version, %ld\n",evtmsg->Version);
}

if (evtmsg->MsgSeqNumber != 1)
{
    printf("it's the wrong sequence number, %ld\n",
        evtmsg->MsgSeqNumber);
}

if (evtmsg->Control != MQCFC_LAST)
{
    printf("it's the wrong control option, %ld\n",
        evtmsg->Control);
}

printreas(evtmsg->Reason);
printf("parameter count is %ld\n", evtmsg->ParameterCount);
/*****
/* get a pointer to the start of the parameters */
*****/

paras = (MQCFST *) (evtmsg + 1);
counter = 1;
while (counter <= evtmsg->ParameterCount)
{
    switch (paras->Type)
    {
        case MQCFT_STRING:
            printfmqfst(paras);
            paras = (MQCFST *) ((char *) paras
                + paras->StrucLength);

            break;
        case MQCFT_INTEGER:
            printfmqcfin((MQCFIN*)paras);
            paras = (MQCFST *) ((char *) paras
                + paras->StrucLength);

            break;
        default:
            printf("unknown parameter type, %ld\n",
                paras->Type);
            counter = evtmsg->ParameterCount;
            break;
    }
    counter++;
}
} /* end evtmsg action */
free(evtmsg);
evtmsg = NULL;
} /* end process for successful GET */
} /* end message processing loop */

/*****
/* close the event queue - if it was opened */
*****/
if (OpenCode != MQCC_FAILED)
{
    C_options = 0; /* no close options */
    MQCLOSE(Hcon, /* connection handle */
        &Hobj, /* object handle */
        C_options,
        &CompCode, /* completion code */
        &Reason); /* reason code */
/*****
/* Disconnect from queue manager (unless previously connected) */
*****/

```

```

if (CReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&Hcon,          /* connection handle      */
          &CompCode,      /* completion code       */
          &Reason);      /* reason code           */

    /*****
    /* END OF EVMON
    /*
    /*
    /*****
}

#define PRINTPARAM(param)          \
    case param:                    \
    {                               \
        char *p = #param;          \
        strncpy(thestring,pmqcfst->String,min(sizeof(thestring), \
        pmqcfst->StringLength)); \
        printf("%s %s\n",p,thestring); \
    }                               \
    break;

#define PRINTAT(param)            \
    case param:                    \
    printf("MQIA_APPL_TYPE = %s\n",#param); \
    break;

void printfmqcfst(MQCFST* pmqcfst)
{
    char thestring[100];

    switch (pmqcfst->Parameter)
    {
        PRINTPARAM(MQCA_BASE_Q_NAME)
        PRINTPARAM(MQCA_PROCESS_NAME)
        PRINTPARAM(MQCA_Q_MGR_NAME)
        PRINTPARAM(MQCA_Q_NAME)
        PRINTPARAM(MQCA_XMIT_Q_NAME)
        PRINTPARAM(MQCACF_APPL_NAME)
        :
        default:
            printf("Invalid parameter, %ld\n",pmqcfst->Parameter);
            break;
    }
}

void printfmqcfin(MQCFIN* pmqcfst)
{
    switch (pmqcfst->Parameter)
    {
        case MQIA_APPL_TYPE:
            switch (pmqcfst->Value)
            {
                PRINTAT(MQAT_UNKNOWN)
                PRINTAT(MQAT_OS2)
                PRINTAT(MQAT_DOS)
                PRINTAT(MQAT_UNIX)
                PRINTAT(MQAT_QMGR)
                PRINTAT(MQAT_OS400)
                PRINTAT(MQAT_WINDOWS)
                PRINTAT(MQAT_CICS_VSE)
                PRINTAT(MQAT_VMS)
                PRINTAT(MQAT_GUARDIAN)
                PRINTAT(MQAT_VOS)
            }
            break;
        case MQIA_Q_TYPE:
            if (pmqcfst->Value == MQQT_ALIAS)
            {
                printf("MQIA_Q_TYPE is MQQT_ALIAS\n");
            }
            else
            {
                :
            }
            if (pmqcfst->Value == MQQT_REMOTE)
            {

```

```

        printf("MQIA_Q_TYPE is MQQT_REMOTE\n");
        if (evtmsg->Reason == MQRC_ALIAS_BASE_Q_TYPE_ERROR)
        {
            printf("but remote is not valid here\n");
        }
    }
    else
    {
        printf("MQIA_Q_TYPE is wrong, %ld\n",pmqcfst->Value);
    }
}
break;

        case MQIACF_REASON_QUALIFIER:
        printf("MQIACF_REASON_QUALIFIER %ld\n",pmqcfst->Value);
        break;

    case MQIACF_ERROR_IDENTIFIER:
        printf("MQIACF_ERROR_IDENTIFIER %ld (X'%lX')\n",
            pmqcfst->Value,pmqcfst->Value);
        break;

    case MQIACF_AUX_ERROR_DATA_INT_1:
        printf("MQIACF_AUX_ERROR_DATA_INT_1 %ld (X'%lX')\n",
            pmqcfst->Value,pmqcfst->Value);
        break;

    case MQIACF_AUX_ERROR_DATA_INT_2:
        printf("MQIACF_AUX_ERROR_DATA_INT_2 %ld (X'%lX')\n",
            pmqcfst->Value,pmqcfst->Value);
        break;
:
default :
    printf("Invalid parameter, %ld\n",pmqcfst->Parameter);
    break;
}
}

void printreas(MQLONG reason)
{
    switch (reason)
    {
        PRINTREAS(MQRCCF_CFH_TYPE_ERROR)
        PRINTREAS(MQRCCF_CFH_LENGTH_ERROR)
        PRINTREAS(MQRCCF_CFH_VERSION_ERROR)
        PRINTREAS(MQRCCF_CFH_MSG_SEQ_NUMBER_ERR)
        :
        PRINTREAS(MQRC_NO_MSG_LOCKED)
        PRINTREAS(MQRC_CONNECTION_NOT_AUTHORIZED)
        PRINTREAS(MQRC_MSG_TOO_BIG_FOR_CHANNEL)
        PRINTREAS(MQRC_CALL_IN_PROGRESS)
        default:
            printf("It's an unknown reason, %ld\n",
                reason);
            break;
    }
}
}
}

```

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.

„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.

Zugehörige Verweise

C-Programmierung

„Beispielprogramm zur Überwachung der Ereigniswarteschlange der Protokollfunktion“ auf Seite 46
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 WebSphere MQ-Anwendung zur Routenanzeige (dspmqrte)
- Aktivitätsaufzeichnung
- 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 Konzepte

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 Übertragungwarteschlange 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 WebSphere 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 Übertragungwarteschlange 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-Steuerkomponente WebSphere MQ, die eine Nachricht verarbeitet, die folgenden Operationen umfassen:

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 WebSphere MQ -Anwendung zur Routenanzeige kann Traceroute-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 Aktivi-

tätsberichte 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 57

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 59

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 66

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 Warteschlangenmanagernetzwerk 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 WebSphere MQ Version 6.0 und nachfolgende Releases

MCAs und benutzerdefinierte Anwendungen, die mit Warteschlangenmanagern unter WebSphere MQ Version 6.0 oder einem nachfolgenden Release verbunden sind, können Informationen zu den Aktivitäten aufzeichnen, die für eine Nachricht ausgeführt werden. Die Aufzeichnung von Aktivitätsinformationen wird von den WS-Managerattributen ACTIVREC und ROUTEREC gesteuert. Wenn ein Warteschlangenmanagernetz nur aus Warteschlangenmanagern unter WebSphere MQ Version 6.0 oder nachfolgenden Releases besteht, können vollständige Nachrichtenrouten ermittelt werden.

WebSphere MQ vor Version 6.0

Anwendungen, die mit WebSphere MQ -Warteschlangenmanagern vor Version 6.0 **Nicht** verbunden sind, zeichnen die Aktivitäten auf, die sie für eine Nachricht ausgeführt haben. Wenn ein Warteschlangenmanagernetz einen WebSphere MQ -Warteschlangenmanager vor Version 6.0 enthält, kann nur ein Teil der Nachrichtenroute bestimmt werden.

Wie Aktivitätsinformationen gespeichert werden

WebSphere 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 aufgezeichnete 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 61

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 66

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 72

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

„Aktivitätsbericht-Nachrichtendaten“ auf Seite 110

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 zusammen mit Trace-Route-Nachrichten verwenden, indem Sie die WebSphere MQ -Anwendung für die Routenanzeige verwenden.

Aktivitätsberichtsformat

Aktivitätsberichte sind PCF-Nachrichten, die von Anwendungen generiert werden, die eine Aktivität im Namen einer Nachricht ausgeführt haben. Aktivitätsberichte sind WebSphere MQ -Standardberichtsnachrichten, die wie folgt einen Nachrichtendeskriptor und Nachrichtendaten 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 Traceroute-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 Sie *nicht* möchten, dass Aktivitätsberichte für eine Nachricht generiert werden, wenn 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 Warteschlangenmanager-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 Warteschlange 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 bestellen, sofern sie nicht für eine Trace-Route-Nachricht generiert werden. In diesem Fall können Sie die WebSphere MQ -Anwendung für die Routenanzeige verwenden, um die Aktivitätsberichte zu sortieren.

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 im Aktivitätsbericht beschrieben wird, hat den *OperationType* send und Details, die die Nachricht über den Kanal CH1an den Zielwarteschlangenmanager QM1gesendet wurde. Dies bedeutet, dass die nächste Aktivität, die für die Nachricht ausgeführt wurde, auf dem WS-Manager QM1aufgetreten ist und dass sie mit einer receive -Operation von Kanal CH1begonnen 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 Warteschlangenmanagern im Warteschlangenmanager-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 von WebSphere MQ vor Version 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 WebSphere MQ -Anwendung für die Routenanzeige können Sie Traceroute-Nachrichten konfigurieren, generieren und in ein Warteschlangenmanagernetz einreihen.

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

Zugehörige Konzepte

„Trace-Route-Nachrichtenreferenz“ auf Seite 127

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 Traceroute-Nachricht ausgeführt wird, wird eine *Activity* -PCF-Gruppe an das Ende des PCF-Blocks in den Nachrichtendaten der Traceroute-Nachricht geschrieben.

Weitere Aktivitätsinformationen werden im Traceroute-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 Traceroute-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 67.

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 Traceroute-Nachricht bei der Ankunft in die Zielwarteschlange gestellt wird, oder ob sie verworfen wird. Wenn die Trace-Route-Nachricht an die Zielwarteschlange gestellt wird, können Sie die Trace-Route-Nachricht aus dieser Warteschlange abrufen. Anschließend können Sie die WebSphere MQ -Anwendung für die Routenanzeige verwenden, um die Aktivitätsinformationen anzuzeigen.

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 `MQRROUTE_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. Sie können die WebSphere MQ -Anwendung für die Routenanzeige verwenden, um die Aktivitätsinformationen anzuzeigen.

Um eine Traceroute-Antwortnachricht anzufordern, setzen Sie den Parameter *Accumulate* in der PCF-Gruppe von *TraceRoute* auf `MQRROUTE_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 Warte-

schlangenmanager 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 62.

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 71.
- 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 71

Eine Trace-Route-Nachricht enthält bestimmte Nachrichtendeskriptoren und Nachrichtendatenteile. Um eine Trace-Route-Nachricht zu generieren, erstellen Sie die Nachricht entweder manuell oder verwenden Sie die WebSphere MQ -Anwendung zur Routenanzeige.

Zugehörige Tasks

[„Aktivitätsaufzeichnung steuern“](#) auf Seite 62

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 Traceroute-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 Traceroute-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 Traceroute-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 Traceroute-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 Traceroute-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 Warteschlangenmanager-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 127.

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-Nachrichtenaufzeichnung 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 Traceroute-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 Traceroute-Messaging aktiviert ist, schreiben Sie eine Aktivität-PCF-Gruppe an das Ende des PCF-Blocks in den Nachrichtendaten einer Traceroute-Nachricht.
 5. Liefern Sie die Trace-Route-Nachricht an eine lokale Warteschlange.
 - Wenn der Parameter, *Deliver*, als MQROUTE_DELIVER_NO angegeben wird, wird die Traceroute-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. Um eine Trace-Route-Nachricht zu generieren, erstellen Sie die Nachricht entweder manuell oder verwenden Sie die WebSphere MQ -Anwendung zur 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 Traceroute-Nachrichtendaten, wenn *Format* auf MQFMT_ADMIN gesetzt ist
- Ein eingebetteter PCF-Header (MQEPH), Traceroute-Nachrichtendaten und zusätzliche benutzerdefinierte Nachrichtendaten, wenn *Format* auf MQFMT_EMBEDDED_PCF gesetzt ist.

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

Manuelle Generierung

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

Die WebSphere MQ -Anwendung für die Routenanzeige

Verwenden Sie die WebSphere MQ -Anwendung für die Routenanzeige, `dspmqrte`, um eine Traceroute-Nachricht zu konfigurieren, zu generieren und in ein Warteschlangenmanagernetz einzureihen. Setzen Sie den Parameter *Format* im Nachrichtendeskriptor auf `MQFMT_ADMIN`. Sie können der Trace-Route-Nachricht, die von der Anwendung WebSphere MQ zur Routenanzeige generiert wird, keine Benutzerdaten hinzufügen.

Einschränkung: `dspmqrte` kann nicht auf Warteschlangenmanagern vor WebSphere MQ Version 6.0 oder auf WebSphere MQ für z/OS -Warteschlangenmanagern ausgegeben werden. Wenn der erste Warteschlangenmanager, über den die Trace-Route-Nachricht weitergeleitet wird, ein Warteschlangenmanager dieses Typs sein soll, stellen Sie über den optionalen Parameter `-cals` WebSphere MQ Version 6.0 oder höher eine Verbindung zum Warteschlangenmanager 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 WebSphere MQ -Anwendung für die Routenanzeige kann die Nachrichtengröße nicht 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 WebSphere MQ -Routenanwendung kann keine Nachrichtendaten angeben.

PCF-Gruppe "TraceRoute"

Die Attribute in der PCF-Gruppe von *TraceRoute* steuern das Verhalten einer Traceroute-Nachricht. Die *TraceRoute* -PCF-Gruppe befindet sich in den Nachrichtendaten jeder Traceroute-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 78).

Tabelle 12. PCF-Gruppe "TraceRoute"

Parameter	Typ
TraceRoute	MQCFGR
Details	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 der WebSphere MQ Version 6 (oder höher) gelöscht.
- Die Traceroute-Nachricht wird von einem Warteschlangenmanager der WebSphere MQ Version 6 (oder höher) in eine lokale Warteschlange (Zielwarteschlange oder Warteschlange für nicht zustellbare Nachrichten) eingereicht.
- Die Anzahl der Aktivitäten, die in der Traceroute-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 Traceroute-Nachricht wird nur an Warteschlangenmanager weitergeleitet, die den Wert des Parameters *Deliver* aus der Gruppe *TraceRoute* berücksichtigen.

MQROUTE_FORWARD_ALL

Die Traceroute-Nachricht wird an jeden Warteschlangenmanager 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 75 fort.
 - Wenn der ferne Warteschlangenmanager nicht in der Lage ist, Tracing-Route-Messaging zu unterstützen, fährt der Algorithmus mit Schritt „2“ auf Seite 75 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 75 fort.
3. Ermitteln Sie den Wert des Parameters *Deliver* aus der PCF-Gruppe *TraceRoute* in der Traceroute-Nachricht.
 - Wenn *Deliver* als MQROUTE_DELIVER_YES angegeben wird, wird die Traceroute-Nachricht an den fernen Warteschlangenmanager weitergeleitet.
 - Wenn *Deliver* als MQROUTE_DELIVER_NO angegeben wird, wird der Algorithmus mit Schritt „4“ auf Seite 75 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 75 fort.
5. Ermitteln Sie den Wert des Parameters *Forward* aus der PCF-Gruppe *TraceRoute* in der Traceroute-Nachricht.
 - Wenn *Forward* als MQROUTE_FORWARD_IF_SUPPORTED angegeben wird, wird die Traceroute-Nachricht mit dem Feedback MQFB_NOT_FORWARDED zurückgewiesen.
 - Wenn *Forward* als MQROUTE_FORWARD_ALL angegeben wird, kann die Traceroute-Nachricht an den fernen Warteschlangenmanager 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 Traceroute-Messaging zu aktivieren und anzugeben, dass alle generierten Traceroute-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 WebSphere MQ -Anwendung für die 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 Traceroute-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 Traceroute-Nachricht nur dann anfordern, wenn Sie die Position der Traceroute-Nachricht kennen und den Parameter *Accumulate* in der *TraceRoute* -PCF-Gruppe entweder als *MQRROUTE_ACCUMULATE_IN_MSG* oder als *MQRROUTE_ACCUMULATE_AND_REPLY* angegeben hat.

Damit die Traceroute-Nachricht an die Zielwarteschlange zugestellt werden kann, muss der Parameter *Deliver* in der *TraceRoute* -PCF-Gruppe als *MQRROUTE_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 WebSphere MQ -Anwendung für die 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 sie manuell bestellen oder die WebSphere MQ -Anwendung für die Routenanzeige verwenden, um die Aktivitätsinformationen automatisch zu bestellen und anzuzeigen.
2. Die Aktivitätsinformationen studieren und die benötigten Informationen abrufen

Zusätzliche Aktivitätsinformationen

Da eine Traceroute-Nachricht über ein Warteschlangenmanager-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 Traceroute-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 WebSphere 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 WebSphere 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

<i>Tabelle 13. Gruppename</i>	
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 WebSphere 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> .
Wert:	Der Name, der angezeigt werden soll.

ParameterValue

Tabelle 15. Parameterwert	
Beschreibung	Enthält den Wert, der von der IBM WebSphere MQ-Anwendung für die Routenanzeige angezeigt werden soll.
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 WebSphere MQ-Anwendung für die Routenanzeige eine Trace-Route-Nachricht generiert und es werden die an diese zurückgegebenen Aktivitätsinformationen angezeigt.

Beispiel 1

Zusätzliche Aktivitätsinformationen werden von einer Benutzeranwendung in einem Format aufgezeichnet, bei dem die Parameter-ID *nicht* von der WebSphere MQ -Routenanwendung erkannt wird.

1. Die WebSphere MQ -Anwendung für die Routenanzeige wird verwendet, um eine Trace-Route-Nachricht zu generieren und in ein Warteschlangenmanagernetz einzureihen. 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.
 - Beim Empfang der Traceroute-Antwortnachricht zeigt die WebSphere MQ -Anwendung für die Routenanzeige die aufgelaufenen Aktivitätsinformationen an.

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 enthält weitere Informationen zu der Aktivität, die ausgeführt wurde. Er wird jedoch in einem Format geschrieben, bei dem die Parameter-ID *nicht* von der WebSphere MQ -Anwendung für die Routenanzeige erkannt wird.

3. Die Trace-Route-Nachrichten erreichen die Zielwarteschlange und eine Trace-Route-Antwortnachricht wird an die WebSphere MQ -Anwendung zur Routenanzeige zurückgegeben. Die zusätzlichen Aktivitätssinformationen werden wie folgt angezeigt:

```
65536: 'Red'
```

Die WebSphere MQ -Routenanwendung 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 dafür, wann die MQ -Anwendung für die Routenanzeige von WebSphere die Parameter-ID des PCF-Parameters erkennt, finden Sie unter [„Beispiel 2“](#) auf Seite 80.

Beispiel 2

Weitere Aktivitätssinformationen werden von einer Benutzeranwendung in einem Format aufgezeichnet, das die Erkennung der Parameterkennung durch die IBM WebSphere MQ-Anwendung für die Routenanzeige erlaubt.

1. Mit der IBM WebSphere MQ-Anwendung für die Routenanzeige wird eine Trace-Route-Nachricht generiert und auf die gleiche Weise in ein Warteschlangenmanagernetz gestellt, wie in [„Beispiel 1“](#) auf Seite 79 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 Standardaktivitätssinformationen 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 WebSphere MQ-Anwendung für die Routenanzeige angezeigt werden soll, mit dem der Wert von <i>ColorValue</i> in den Kontext gestellt wird.
ID:	MQCA_VALUE_NAME.

Tabelle 17. Farbname (Forts.)	
Beschreibung	Enthält den Namen, der von der IBM WebSphere MQ-Anwendung für die Routenanzeige angezeigt werden soll, mit dem der Wert von ColorValue in den Kontext gestellt wird.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	ColorInfo.
Wert:	'Farbe'

ColorValue

Tabelle 18. Farbwert	
Beschreibung	Enthält den Wert, der von der IBM WebSphere 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 aufgezeichnet, das die Erkennung der Parameterkennung durch die IBM WebSphere MQ-Anwendung für die Routenanzeige erlaubt.

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

```
Color: 'Red'
```

Die IBM WebSphere 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.

WebSphere MQ-Anwendung zur Routenanzeige

In der WebSphere MQ-Anwendung zur Routenanzeige (**dspmqrte**) können Sie über eine Befehlszeilenschnittstelle mit Traceroute-Nachrichten und zugehörigen Aktivitätsinformationen arbeiten.

Anmerkung: Um eine Clientanwendung für einen Warteschlangenmanager ausführen zu können, muss das Client Attachment Feature installiert sein.

Die WebSphere MQ-Anwendung zur Routenanzeige kann zu folgenden Zwecken verwendet werden:

- 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 WebSphere MQ-Anwendung zur Routenanzeige eine Traceroute-Nachricht in ein Warteschlangenmanagernetz eingereicht hat, können die zurückgemeldeten zugehörigen Aktivitätsinformationen sofort sortiert und angezeigt werden. Alternativ dazu ist es auch möglich, Aktivitätsinformationen zu einer zuvor erstellten Traceroute-Nachricht über die WebSphere MQ-Anwendung zur Routenanzeige zu sortieren und anzuzeigen.

Zugehörige Verweise

[dspmqrte](#)

Parameter für Trace-Route-Nachrichten

Verwenden Sie diese Seite, um eine Übersicht über die Parameter zu erhalten, die von der WebSphere MQ-Anwendung für die Routenanzeige, **dspmqrte**, bereitgestellt werden, um die Merkmale einer Trace-Route-Nachricht zu ermitteln, einschließlich der Art und Weise, wie sie behandelt wird, wenn sie über ein Warteschlangenmanagernetz weitergeleitet wird.

Zugehörige Verweise

[dspmqrte](#)

Warteschlangenmanagerverbindung

Auf dieser Seite können Sie den Warteschlangenmanager angeben, zu dem die WebSphere MQ-Anwendung für die Routenanzeige eine Verbindung herstellt.

-c

Gibt an, dass die WebSphere MQ-Anwendung zur Routenanzeige eine Verbindung als Clientanwendung herstellt.

Wenn Sie diesen Parameter nicht angeben, stellt die MQ-Anwendung für die Routenanzeige von WebSphere keine Verbindung als Clientanwendung her.

-m *QMgrName*

Der Name des Warteschlangenmanagers, zum dem die WebSphere MQ-Anwendung zur 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 WebSphere MQ-Anwendung zur Routenanzeige verwendet wird, um eine Traceroute-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. Das Ziel wird mit *-q TargetQName* oder *-ts TargetTopicString* angegeben.

Wenn Sie diesen Parameter nicht angeben, wird der Warteschlangenmanager, mit dem die WebSphere MQ-Anwendung zur 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

Für Publish/Subscribe-Anwendungen verwenden Sie diese Seite, um die Themenzeichenfolge einer Trace-Route-Nachricht für die WebSphere MQ -Routenanwendung anzugeben, die veröffentlicht werden soll.

-ts *TopicName*

Gibt eine Themenzeichenfolge an, für die die WebSphere MQ-Anwendung zur Routenanzeige eine Traceroute-Nachricht veröffentlichen soll, und setzt diese Anwendung in den Themenmodus. In diesem Modus verfolgt die Anwendung alle Nachrichten, die sich aus der Veröffentlichungsanforderung ergeben.

Sie können auch die WebSphere MQ -Anwendung für die Routenanzeige verwenden, um die Ergebnisse eines Aktivitätsberichts anzuzeigen, der für Veröffentlichungsnachrichten 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 WebSphere 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. Mögliche Werte für *Persistenz* 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 *Persistenz* 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 *Priorität* ist größer oder 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. Mögliche Werte für die *Berichtsoption* 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 WebSphere 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:

Die Berichtsoption activity , angegeben mit -ro

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.

Ohne Angabe dieses Parameters werden die Aktivitätsinformationen **nicht** in der Traceroute-Nachricht erfasst.

-ar

Fordert an, dass eine Trace-Route-Antwortnachricht mit allen aufgelaufenen Aktivitätsinformationen in den folgenden Situationen generiert wird:

- Die Traceroute-Nachricht wird von einem Warteschlangenmanager aus IBM WebSphere MQ verworfen.

- Die Traceroute-Nachricht wird durch einen Warteschlangenmanager aus IBM WebSphere MQ in eine lokale Warteschlange eingereiht.
- 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.

Ohne Angabe dieses Parameters werden die Aktivitätsinformationen **nicht** in der Traceroute-Nachricht erfasst.

-d Ja

Bei der Ankunft wird die Trace-Route-Nachricht in die Zielwarteschlange gestellt, auch wenn der WS-Manager keine Trace-Route-Messaging unterstützt.

Ohne Angabe dieses Parameters wird die Traceroute-Nachricht **nicht** in die Zielwarteschlange eingereiht.

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

In **low** angegebene Aktivitäten werden aufgezeichnet. Darüber hinaus werden Veröffentlichungsaktivitäten und Aktivitäten, die von MCAs ausgeführt werden, aufgezeichnet.

hoch

In **low** und **medium** angegebene Aktivitäten 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 WebSphere MQ-Anwendung für die Routenanzeige speichert die zurückgegebenen Nachrichten standardmäßig in einer temporären dynamischen Warteschlange. Wenn die IBM WebSphere 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 WebSphere 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 Traceroute-Nachricht persistent ist oder der Parameter *-n* angegeben wurde, darf die als Warteschlange für Antwortnachrichten angegebene Warteschlange **keine** temporäre dynamische Warteschlange sein.

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 WebSphere 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:

yes	Bei der Ankunft wird die Trace-Route-Nachricht in die Zielwarteschlange gestellt, auch wenn der WS-Manager keine Trace-Route-Messaging unterstützt.
nein	Beim Eingang wird die Traceroute-Nachricht nicht in die Zielwarteschlange eingereicht.

Ohne Angabe dieses Parameters wird die Traceroute-Nachricht **nicht** in die Zielwarteschlange eingereicht.

-f Forward

Gibt den Typ des Warteschlangenmanagers an, an den die Trace-Route-Nachricht weitergeleitet werden kann. Ausführliche Informationen zu dem Algorithmus, mit dem Warteschlangenmanager bestimmen, ob eine Nachricht an einen fernen Warteschlangenmanager weitergeleitet werden soll, finden Sie unter „PCF-Gruppe "TraceRoute"“ auf Seite 72. Die gültigen Werte für *Forward* sind:

alle

Die Trace-Route-Nachricht wird an einen beliebigen WS-Manager weitergeleitet.

Warnung: Bei einer Weiterleitung an einen IBM WebSphere MQ -Warteschlangenmanager vor Version 6.0 wird die Trace-Route-Nachricht nicht erkannt und kann trotz des Werts des Parameters *-d Zustellung* an eine lokale Warteschlange zugestellt werden.

unterstützt

Die Traceroute-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 Traceroute-Nachricht nur an einen Warteschlangenmanager 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.

Die Berichtsoption discard , angegeben mit -ro

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 WebSphere 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 Traceroute-Antwortnachricht (-ar) oder einer der Berichtsgenerierungsoptionen aus (-ro *ReportOption*) begleitet wird, muss mit -rq *ReplyToQ* eine bestimmte Antwortwarteschlange (kein Modell) angegeben werden. Standardmäßig werden nur Aktivitätsberichtsnachrichten 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 WebSphere 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 für die Traceroute-Nachricht zurückgegebene Aktivitätsinformationen in dem Format angezeigt, das vom Parameter -v angegeben wird.

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 WebSphere 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 WebSphere MQ-Anwendung zur Routenanzeige verwendet wird, um vorher zusammengestellte Aktivitätsinformationen anzuzeigen, gibt *Name der Zielwarteschlange* den Namen der Warteschlange an, in der die Aktivitätsinformationen gespeichert sind.

-i *CorrelId*

Dieser Parameter wird verwendet, wenn mit der IBM WebSphere 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 Aktivitätenberichte oder eine Traceroute-Antwortnachricht zu einer Traceroute-Nachricht anzugeben. Geben Sie in *CorrelId* die Nachrichten-ID der ursprünglichen Traceroute-Nachricht an.

Das Format der *CorrelId* ist eine 48 Zeichen umfassende hexadezimale Zeichenfolge.

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 WebSphere 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 WebSphere 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.

outline *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

Mögliche 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

Alles Nicht-PCF-Gruppenparameter in *Message*-PCF-Gruppen, die sich seit der letzten Operation geändert haben, 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 WebSphere 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 78 beschrieben wird.

Beispiele für die WebSphere MQ -Anwendung für die Routenanzeige

Die folgenden Beispiele zeigen, wie Sie die WebSphere MQ -Anwendung für die Routenanzeige verwenden. In jedem Beispiel sind zwei WS-Manager (QM1 und QM2) über zwei Kanäle miteinander verbunden (QM2.TO.QM1 und QM1.TO.QM2).

Beispiel 1-Aktivitätsberichte anfordern

Aktivitätsinformationen aus einer Trace-Route-Nachricht anzeigen, die an die Zielwarteschlange zugestellt wird

In diesem Beispiel stellt die WebSphere MQ -Anwendung zur Routenanzeige eine Verbindung zum Warteschlangenmanager QM1 her und wird zum Generieren und Zustellen einer Trace-Route-Nachricht an die Zielwarteschlange TARGET.Q auf dem fernen Warteschlangenmanager QM2. 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 an die WebSphere MQ -Anwendung für die Routenanzeige mithilfe von Aktivitätsberichten zurückgegeben werden, werden in der richtigen Reihenfolge angezeigt.

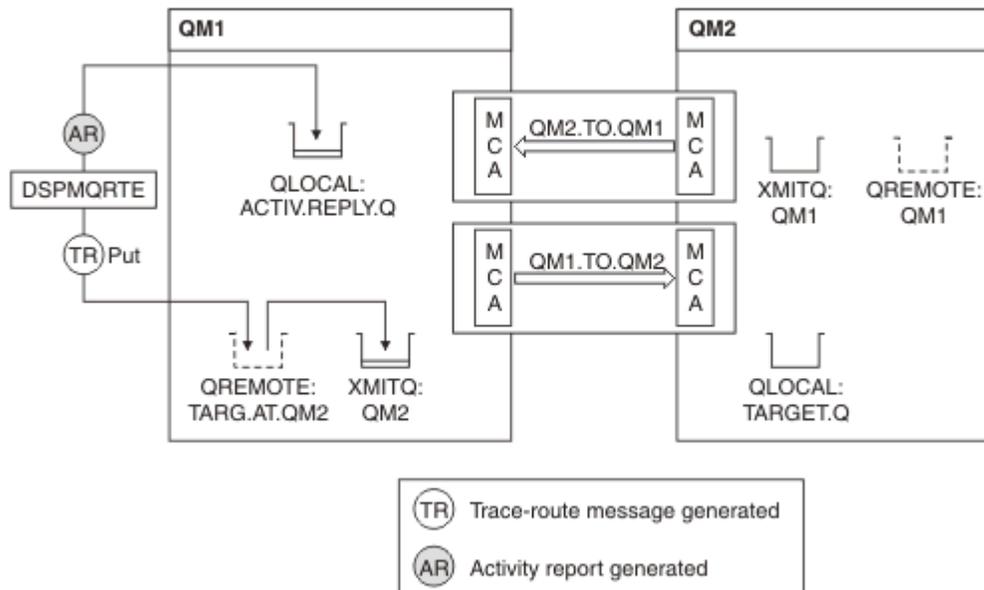


Abbildung 9. Aktivitätenberichte anfordern, Diagramm 1

- 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, zu dem die WebSphere 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, an die angefordert wird, dass alle Antworten auf die Trace-Route-Nachricht gesendet werden.

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.

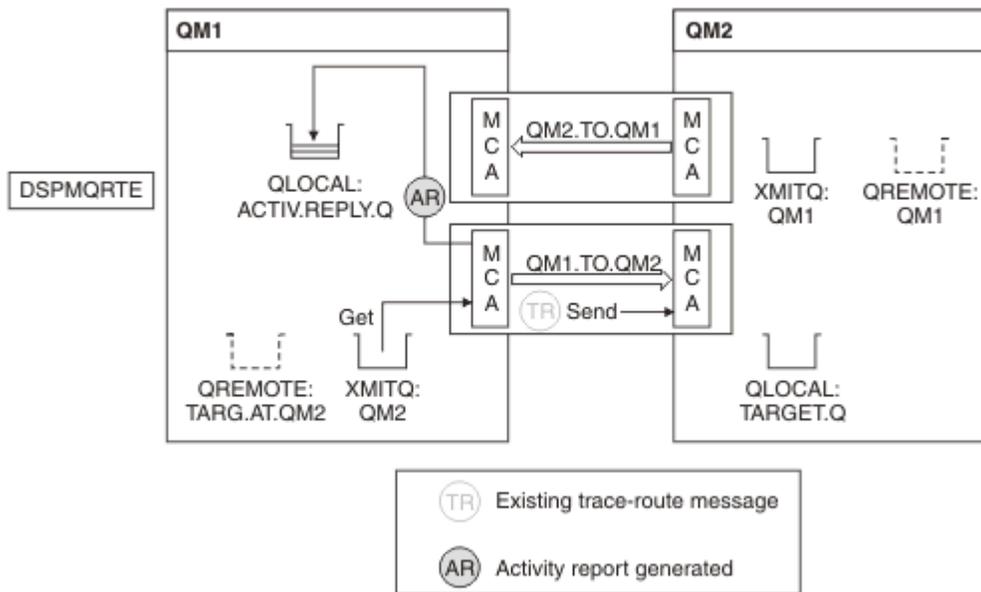


Abbildung 10. Aktivitätenberichte anfordern, Diagramm 2

- 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 75, „4“ auf Seite 75 und „5“ auf Seite 75), und der Nachrichtenkanalmanager wählt die Nachricht aus.
- Der MCA generiert dann einen Aktivitätsbericht und stellt ihn in die Antwortwarteschlange (ACTIV.REPLY.Q).

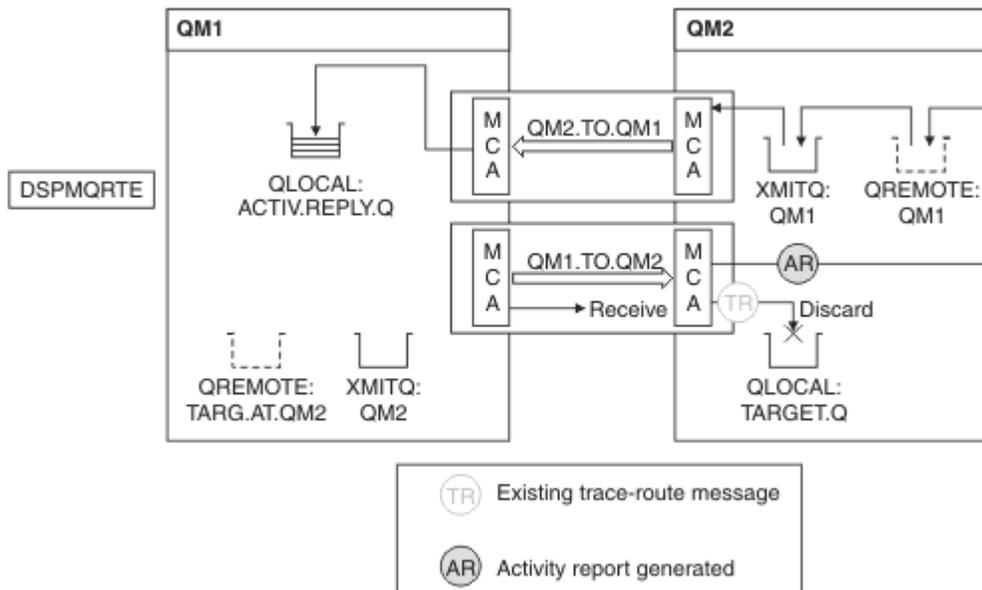


Abbildung 11. Aktivitätenberichte anfordern, Diagramm 3

- 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 5.3.1 oder einer früheren Version stammt, erhöht der MCA den Parameter DiscontinuityCount 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.

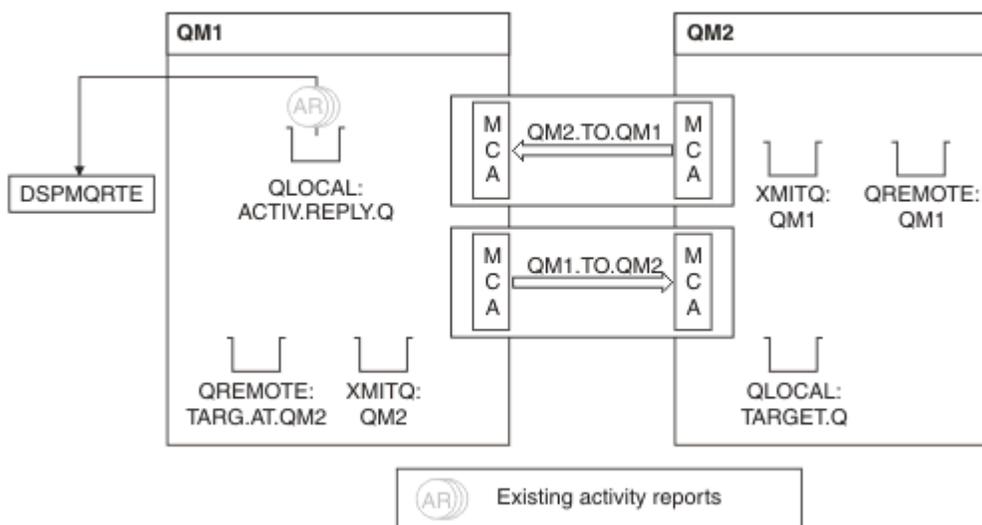


Abbildung 12. Aktivitätenberichte anfordern, Diagramm 4

- 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 WebSphere MQ -Anwendung zur Routenanzeige eine Verbindung zum Warteschlangenmanager QM1 her und wird zum Generieren und Zustellen einer Trace-Route-Nachricht an die Zielwarteschlange TARGET.Q auf dem fernen Warteschlangenmanager QM2. 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.

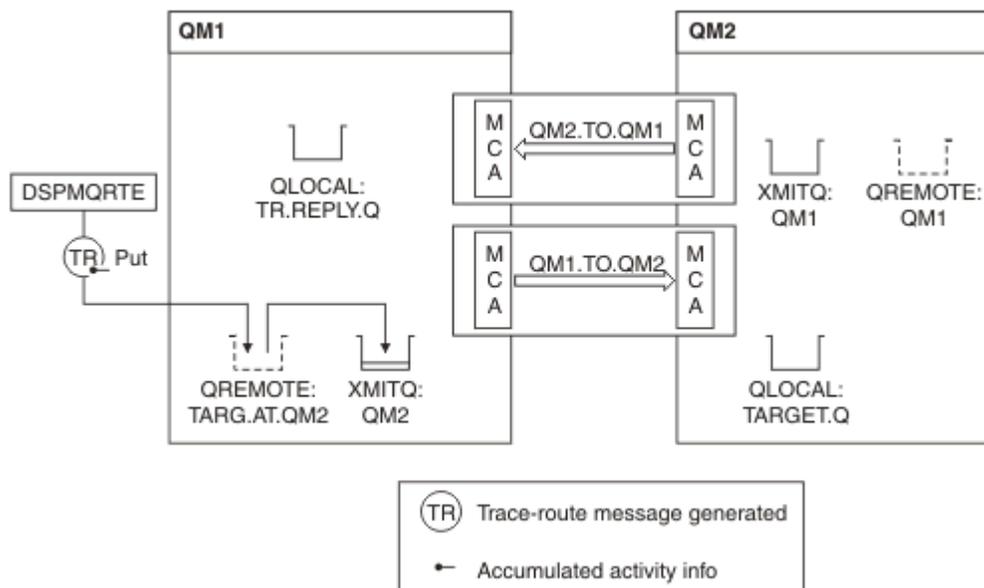


Abbildung 13. Anfordern einer Trace-Route-Antwortnachricht, Diagramm 1

- Das Attribut `ROUTEREC` der einzelnen WS-Manager (QM1 und QM2) wird auf `MSG` gesetzt.
- Der folgende Befehl wird ausgegeben:

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq TR.REPLY.Q -ac -ar -ro discard
```

QM1 ist der Name des Warteschlangenmanagers, zu dem die WebSphere 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, an die angefordert wird, dass alle Antworten auf die Trace-Route-Nachricht gesendet werden. Die Option -ac gibt an, dass die Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden. Die Option -ar gibt an, dass die gesamte aufgelaufene Aktivität an die Warteschlange für Antwortnachrichten gesendet wird, die durch die Option -rq angegeben wird (d. r. TR.REPLY.Q). Die Option -ro gibt an, dass die Berichtsoption MQRO_DISCARD_MSG festgelegt ist. Dies bedeutet, dass in diesem Beispiel keine Aktivitätsberichte generiert werden.

- DSPMQRTE akkumuliert die Aktivitätsinformationen in der Trace-Route-Nachricht, bevor die Nachricht auf die Zielroute gestellt wird. Das WS-Manager-Attribut ROUTEREC darf nicht DISABLED sein, damit dies geschehen kann.

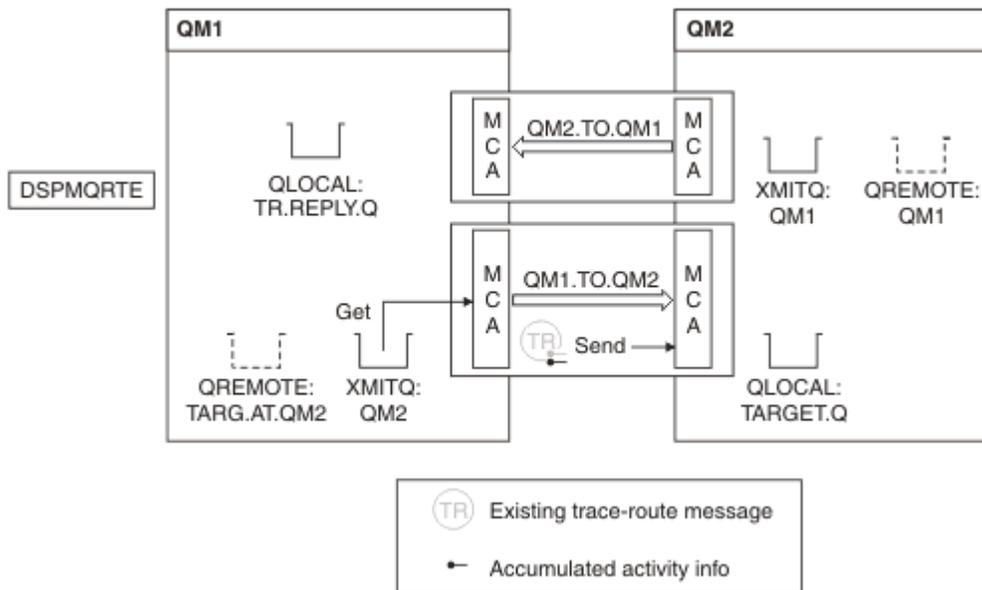


Abbildung 14. Antwort auf Trace-Route-Antwortnachricht anfordern, Diagramm 2

- Die Nachricht ist eine Trace-Route-Nachricht, daher beginnt der sendende MCA mit dem Aufzeichnen von Informationen über die Aktivität.
- Das WS-Manager-Attribut ROUTEREC auf WSM1 ist nicht DISABLED, daher sammelt der MCA die Aktivitätsinformationen in der Nachricht, bevor die Nachricht an den Warteschlangenmanager QM2 weitergeleitet wird.

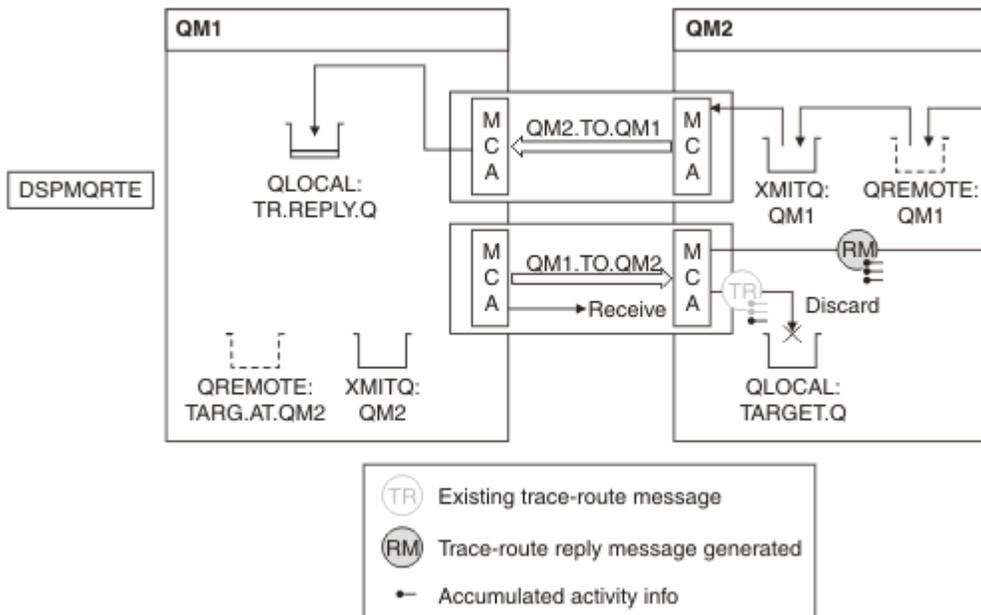


Abbildung 15. Antwort auf Trace-Route-Antwortnachricht anfordern, Diagramm 3

- 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.

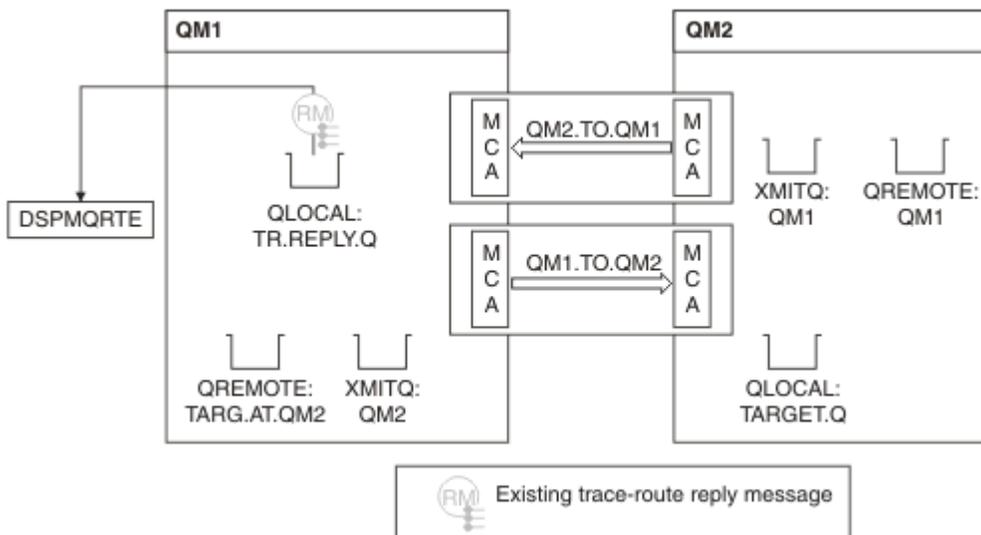


Abbildung 16. Antwort auf Trace-Route-Antwortnachricht anfordern, Diagramm 4

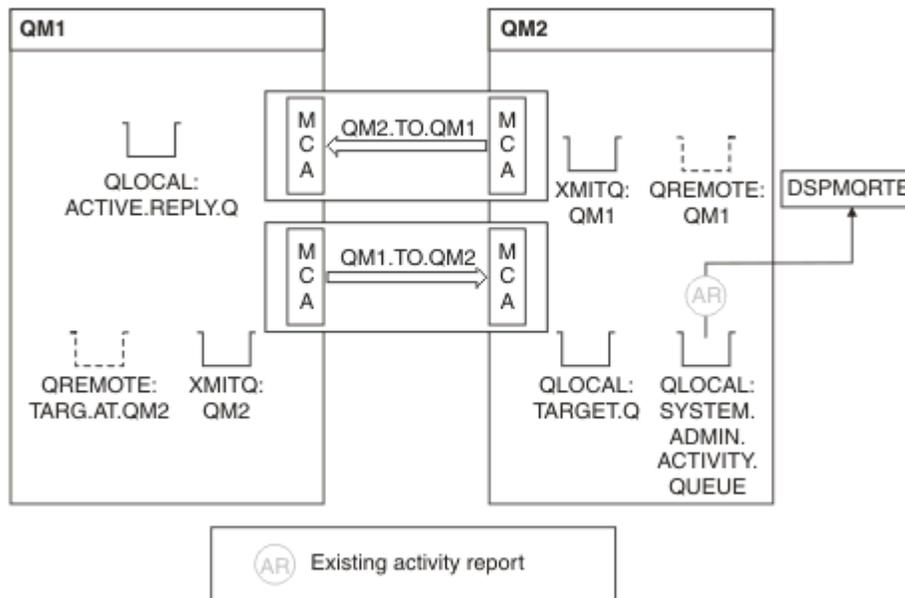


Abbildung 18. Aktivitätsberichte an die Systemwarteschlange übergeben, Diagramm 2

- 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 generiert die WebSphere MQ -Anwendung für die Routenanzeige eine Verbindung zum Warteschlangenmanager QM1, generiert eine Trace-Route-Nachricht und versucht anschließend, sie an die Zielwarteschlange TARGET.Q auf dem fernen Warteschlangenmanager QM2. In diesem Beispiel erreicht die Trace-Route-Nachricht nicht die Zielwarteschlange. Der verfügbare Aktivitätsbericht wird zur Diagnose des Problems verwendet.

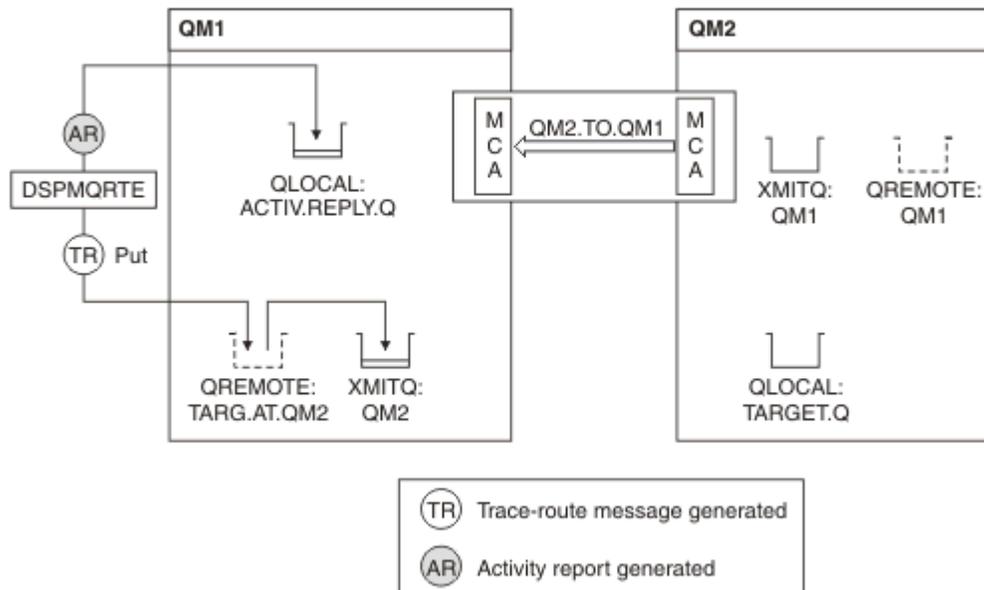


Abbildung 19. Kanalfehler diagnostizieren

- 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 WebSphere MQ-Berichtsnachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Aktivitätsberichte sind PCF-Nachrichten, die von Anwendungen generiert 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 Traceroute-Nachricht generiert wurde, die *TraceRoute* -PCF-Gruppe.

Tabelle 19 auf Seite 102 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 Markierungen 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 ⁸ EmbeddedMQMD Name des Warteschlangenmanagers Name der Gruppe mit gemeinsamer Warteschlange Warteschlangenname ¹ ^{2 3} ⁷ Aufgelöster Warteschlangenname ¹ ³ ⁷ Name der fernen Warteschlange ³ ⁷ Name des fernen Warteschlangenmanagers ² ^{3 4} ⁵ ⁷ Subskriptionsebene ⁹ Subskriptionskennung ⁹ Feedback ² ¹⁰ Kanalname ⁴ ⁵ Kanaltyp ⁴ ⁵ Name der Übertragungswarteschlange ⁵ TraceRoute ⁶ Details 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 -Nachrichtendeskriptorstruktur, unterstützt unter AIX, HP-UX, z/OS, IBM i, Solaris, Linux, Windows und allen WebSphere 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 <i>Report</i> 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 <i>Report</i> 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 die *Version* MQMD_VERSION_2 ist, ist 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 *Encoding* 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 *CodedCharSetId* 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 *Format* des ursprünglichen Nachrichtendeskriptors kopiert.
Andernfalls MQFMT_NONE.

Flags

Beschreibung: Flags, die Attribute der Struktur angeben oder ihre Verarbeitung steuern.
Datentyp: MQLONG.
Wert: **MQEPH_CCSID_EMBEDDED**
Gibt an, dass der Zeichensatz der Parameter, die Zeichendaten enthalten, einzeln im Feld *CodedCharSetId* in jeder Struktur angegeben wird.

PCFHeader

Beschreibung: Header für programmierbares Befehlsformat
Datentyp: MQCFH.
Wert: Siehe [„Aktivitätenbericht MQCFH \(PCF-Header\)“](#) auf Seite 108.

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: **MQCFH_REPORT**
Nachricht ist ein Bericht.

StrucLength

Beschreibung: Strukturlänge.
Datentyp: MQLONG.
Wert: **MQCFH_STRUC_LENGTH**
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: **MQCMD_ACTIVITY_MSG**
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 Aktivitätsberichtsnachricht 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 Traceroute-Nachricht generiert wurde, die *TraceRoute* -PCF-Gruppe. Die *Activity* -PCF-Gruppe ist in diesem Abschnitt ausführlich beschrieben.

Einige Parameter, die als operationsspezifische Aktivitätsberichtsnachrichtendaten beschrieben werden, werden nur zurückgegeben, wenn bestimmte Operationen ausgeführt wurden.

Für einen Aktivitätsbericht enthält die Aktivitätsberichtsnachrichtendaten 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: *ActivityApplName*
ActivityApplType
ActivityDescription
Operation
TraceRoute
Zurückgegeben: Immer.

ActivityApplName

Beschreibung: Name der Anwendung, die die Aktivität ausgeführt hat.
ID: MQCACF_APPL_NAME.
Datentyp: MQCFST.
In PCF-Gruppe enthalten: *Activity*.
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: *Activity*.
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 Aktivitätsberichtsrichtendaten 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 Nachrichtendescriptor 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:	<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: 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_QUEUE.
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:	<i>Operation.</i>
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:	<i>Operation.</i>
Maximale Länge:	MQ_QSG_NAME_LENGTH
Zurückgegeben:	Wenn die Aktivität auf einem Warteschlangenmanager WebSphere MQ für z/OS ausgeführt wurde.

TraceRoute

Beschreibung:	Gruppierte Parameter, die Attribute der Trace-Route-Nachricht angeben.
ID:	MQGACF_TRACE_ROUTE.
Datentyp:	MQCFGR.
Enthalten in PCF- Gruppe:	<i>Activity.</i>
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>
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 Traceroute-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 Aktivitätsberichtsdatenparameter, 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 Aktivitätsberichtsnachrichtendatenparameter, 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_MESSAGE 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:	<i>Operation.</i>
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:	<i>Operation.</i>
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:	<i>Operation.</i>
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:	<i>Operation.</i>
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 Aktivitätsberichtsnachrichtendatenparameter, 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 Standardnachrichten WebSphere MQ , 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 Traceroute-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 die WebSphere MQ -Routenanwendung verwenden, um eine Trace-Route-Nachricht zu generieren, wird *Format* auf MQFMT_ADMIN gesetzt.

Der Inhalt der Traceroute-Nachrichtendaten wird durch den Parameter *Accumulate* aus der *TraceRoute*-PCF-Gruppe wie folgt bestimmt:

- Wenn *Accumulate* auf MQROUTE_ACCUMULATE_NONE gesetzt ist, enthält die Traceroute-Nachrichtendaten die *TraceRoute*-PCF-Gruppe.
- Wenn *Accumulate* auf MQROUTE_ACCUMULATE_IN_MSG oder MQROUTE_ACCUMULATE_AND_REPLY gesetzt ist, enthält die Traceroute-Nachrichtendaten die *TraceRoute*-PCF-Gruppe und null oder mehr *Activity*-PCF-Gruppen.

Tabelle 20 auf Seite 129 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 Markierungen PCF-Header (MQCFH) Strukturtyp Strukturlänge Strukturversion Befehls-ID Nachrichtenfolgennummer Steueroptionen Beendigungscode Ursachencode Parameteranzahl	TraceRoute Details 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.**

Report

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.

Expiry

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:	MQFMT_ADMIN Admin-Nachricht. Es folgen keine Benutzerdaten der <i>TraceRoute</i> -PCF-Gruppe. MQFMT_EMBEDDED_PCF Eingebettete PCF-Nachricht Die Benutzerdaten folgen der <i>TraceRoute</i> -PCF-Gruppe.

Priority

Beschreibung:	Nachrichtenpriorität
Datentyp:	MQLONG.
Wert:	Stellen Sie sie entsprechend den Anforderungen ein.

Persistence

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:	MQEPH_NONE Es wurden keine Markierungen angegeben. MQEPH_CCSID_EMBEDDED 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 „Trace-Route-Nachricht MQCFH (PCF-Header)“ auf Seite 134.

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:	MQCFT_TRACE_ROUTE Nachricht ist eine Trace-Route-Nachricht.

StrucLength

Beschreibung:	Strukturlänge.
Datentyp:	MQLONG.
Wert:	MQCFH_STRUC_LENGTH 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:	MQCMD_TRACE_ROUTE Trace-Route-Nachricht.

MsgSeqNumber

Beschreibung:	Nachrichtenfolgenummer. Dies ist die Folgenummer der Nachricht in einer Gruppe zusammengehöriger Nachrichten.
Datentyp:	MQLONG.
Werte:	1.

Control

Beschreibung:	Steueroptionen.
Datentyp:	MQLONG.
Werte:	MQCF_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 Traceroute-Nachrichtendaten ist.

Der Inhalt der Traceroute-Nachrichtendaten hängt vom Parameter *Accumulate* in der *TraceRoute* -PCF-Gruppe ab. Traceroute-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>MQRROUTE_DETAIL_LOW Aktivitäten, die von einer vom Benutzer geschriebenen Anwendung ausgeführt werden, werden aufgezeichnet.</p> <p>MQRROUTE_DETAIL_MEDIUM Die in MQRROUTE_DETAIL_LOW angegebenen Aktivitäten werden aufgezeichnet. Darüber hinaus werden Aktivitäten, die von MCAs ausgeführt werden, aufgezeichnet.</p> <p>MQRROUTE_DETAIL_HIGH Die in 'MQRROUTE_DETAIL_LOW' und 'MQRROUTE_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:	Eine positive Ganzzahl Die maximale Anzahl der Aktivitäten. MQROUTE_UNLIMITED_ACTIVITIES 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:	MQROUTE_ACCUMULATE_NONE Die Aktivitätsinformationen werden in den Nachrichtendaten der Trace-Route-Nachricht nicht kumuliert. MQROUTE_ACCUMULATE_IN_MSG Die Aktivitätsinformationen werden in den Nachrichtendaten der Trace-Route-Nachricht akkumuliert. MQROUTE_ACCUMULATE_AND_REPLY 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 <u>Weiterleiten</u> beschrieben ist.
ID:	MQIACF_ROUTE_FORWARDING.

Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .
Wert:	<p>MQROUTE_FORWARD_IF_SUPPORTED Die Traceroute-Nachricht wird nur an Warteschlangenmanager weitergeleitet, die den Wert des Parameters <i>Deliver</i> aus der Gruppe <i>TraceRoute</i> berücksichtigen.</p> <p>MQROUTE_FORWARD_ALL Die Traceroute-Nachricht wird an jeden Warteschlangenmanager weitergeleitet, unabhängig davon, ob der Wert des Parameters <i>Deliver</i> berücksichtigt wird.</p>

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:	<i>TraceRoute</i> .
Wert:	<p>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.</p> <p>MQROUTE_DELIVER_NO Bei der Ankunft wird die Trace-Route-Nachricht gelöscht.</p>

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 Standardnachrichten von WebSphere MQ, 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 Traceroute-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 139 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 Aufnahmedatum Aufnahmezeit Ursprungsdaten der Anwendung Gruppen-ID Nachrichtenfolgennummer Offset Nachrichtenmarkierungen Ursprüngliche Länge	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 EmbeddedMQMD Name des Warteschlangenmanagers Name der Gruppe mit gemeinsamer Warteschlange Warteschlangenname ^{1 2 3} Aufgelöster Warteschlangenname ^{1 3} Name der fernen Warteschlange ³ Ferner Warteschlangenmanager-Name ^{2 3 4 5} Feedback ² Kanalname ^{4 5} Kanaltyp ^{4 5} Name der Übertragungswarteschlange ⁵ TraceRoute Details Aufgezeichnete Aktivitäten Nicht aufgezeichnete Aktivitäten Diskontinuitätszähler Max. Aktivitäten Kumulieren Deliver
<p>Anmerkung:</p> <ol style="list-style-type: none"> 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 110 beschrieben.

Abrechnungs-und Statistiknachrichten

Warteschlangenmanager generieren Abrechnungs-und Statistiknachrichten, um Informationen zu den MQI-Operationen aufzuzeichnen, die von IBM WebSphere MQ -Anwendungen ausgeführt werden, oder um Informationen zu den Aktivitäten aufzuzeichnen, die in einem IBM WebSphere MQ -System auftreten.

Abrechnungsnachrichten

Mit Abrechnungsnachrichten werden Informationen zu den MQI-Operationen aufgezeichnet, die von IBM WebSphere MQ-Anwendungen ausgeführt werden (siehe „[Abrechnungsnachrichten](#)“ auf Seite 141).

Statistiknachrichten

Statistiknachrichten werden verwendet, um Informationen zu den Aktivitäten aufzuzeichnen, die in einem IBM WebSphere MQ -System auftreten (siehe „[Statistiknachrichten](#)“ auf Seite 145). Einige in Statistiknachrichten aufgezeichnete Aktivitäten beziehen sich auf interne Warteschlangenmanageroperationen.

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.

Abrechnungsnachrichten

Abrechnungsnachrichten erfassen Informationen zu MQI-Operationen, die von WebSphere 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 WebSphere MQ -Anwendungen mit langer Laufzeit werden temporäre Abrechnungsnachrichten wie folgt 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 159

Verwenden Sie diese Seite, um die Struktur einer MQI-Abrechnungsnachricht anzuzeigen.

„Nachrichtendaten der Warteschlangenabrechnung“ auf Seite 170

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 Nachrichtenkatgorie 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.

MQI-Abrechnungsdaten

Verwenden Sie das Warteschlangenmanagerattribut ACCTMQI, um die Erfassung von MQI-Abrechnungsdaten zu steuern

Verwenden Sie zum Ändern des Werts dieses Attributs den MQSC-Befehl ALTER QMGRund 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.

Verwenden Sie zum Ändern des Werts des Warteschlangenattributs 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. 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)
```

MQCONNX-Optionen

Verwenden Sie den Parameter **ConnectOpts** im Aufruf MQCONNX, 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 Abrechnungsnachrichten

Abrechnungsnachrichten werden generiert, wenn eine Anwendung die Verbindung zum WS-Manager trennt. Temporäre Abrechnungsnachrichten werden auch für WebSphere MQ -Anwendungen mit langer Laufzeit 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 WebSphere MQ -Anwendungen mit langer Laufzeit geschrieben, wenn das Intervall seit der Herstellung der Verbindung 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)
```

Statistikenachrichten

Statistikenachrichten zeichnen Informationen zu den Aktivitäten auf, die in einem WebSphere MQ -System auftreten. Bei einer Statistikenachricht handelt es sich um eine PCF-Nachricht, die eine Reihe von PCF-Strukturen enthält.

Statistikenachrichten werden in den konfigurierten Intervallen an die Systemwarteschlange (SYSTEM.ADMIN.STATISTICS.QUEUE) zugestellt, wenn eine Aktivität vorhanden ist.

Statistikenachrichten befinden sich in den folgenden Kategorien:

MQI-Statistikenachrichten

MQI-Statistikenachrichten 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.

Warteschlangenstatistikenachrichten

Warteschlangenstatistikenachrichten 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 Warteschlangenstatistikenachricht kann bis zu 100 Datensätze enthalten, wobei jeder Datensatz sich auf die Aktivität pro Warteschlange bezieht, für die Statistikdaten erfasst wurden.

Statistikenachrichten 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.

Kanalstatistikenachrichten

Kanalstatistikenachrichten 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 Kanalstatistikenachricht enthält bis zu 100 Datensätze, wobei jeder Datensatz sich auf die Aktivität pro Kanal bezieht, für die Statistikdaten erfasst wurden.

Zugehörige Verweise

„MQI-Statistikdaten“ auf Seite 146

Verwenden Sie das Warteschlangenmanagerattribut STATMQI, um die Erfassung von MQI-Statistikdaten zu steuern.

„Warteschlangenstatistikdaten“ auf Seite 146

Verwenden Sie das Warteschlangenattribut STATQ und das Warteschlangenmanagerattribut STATQ, um die Erfassung von Warteschlangenstatistikdaten zu steuern.

„Kanalstatistikdaten“ auf Seite 148

Verwenden Sie das Kanalattribut STATCHL, um die Erfassung von Kanalstatistikdaten zu steuern. Sie können auch WS-Manager-Attribute festlegen, um die Informationssammlung zu steuern. Diese Attribute sind auf verteilten Plattformen und unter IBM iverfügbar.

Format der Statistikenachrichten

Statistikenachrichten umfassen eine Gruppe von PCF-Feldern, die aus einem Nachrichtendeskriptor und Nachrichtendaten bestehen.

Nachrichtendeskriptor

- Eine Statistikenachricht MQMD (Nachrichtendeskriptor)

Abrechnungsnachrichtendaten

- Eine Statistikenachricht MQCFH (PCF-Header)

- Statistiknachrichtendaten, die immer zurückgegeben werden
- Statistiknachrichtendaten, die zurückgegeben werden, falls verfügbar

Die Statistiknachricht MQCFH (PCF-Header) enthält Informationen zu dem Intervall, für das die Statistikdaten aufgezeichnet wurden.

Statistische Nachrichtendaten enthalten PCF-Parameter, die die Statistikinformationen speichern. Der Inhalt der Statistiknachrichten hängt wie folgt von der Nachrichtenkategorie ab:

MQI-Statistiknachricht

Die MQI-Statistik-Nachrichtendaten bestehen aus einer Reihe von PCF-Parametern, aber keine PCF-Gruppen.

Warteschlangenstatistiknachricht

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 Statistiknachrichten 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.

Kanalstatistiknachricht

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 Statistiknachrichten 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.

MQI-Statistikdaten

Verwenden Sie das Warteschlangenmanagerattribut STATMQI, um die Erfassung von MQI-Statistikdaten zu steuern.

Um den Wert dieses Attributs zu ändern, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter STATMQI an. Statistiknachrichten werden nur für Warteschlangen generiert, die nach der Aktivierung der Statistikerfassung geöffnet werden. Der Parameter STATMQI kann die folgenden Werte haben:

ON

MQI-Statistikinformationen werden für jede Verbindung zum Warteschlangenmanager erfasst.

OFF

Die MQI-Statistikinformationen werden nicht erfasst. Dies ist der Standardwert.

Wenn Sie beispielsweise die Erfassung von MQI-Statistikdaten aktivieren möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QMGR STATMQI(ON)
```

Warteschlangenstatistikdaten

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 WebSphere 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:

distributed Kanalstatistikdaten

Verwenden Sie das Kanalattribut STATCHL, um die Erfassung von Kanalstatistikdaten zu steuern. Sie können auch WS-Manager-Attribute festlegen, um die Informationssammlung zu steuern. Diese Attribute sind auf verteilten Plattformen und unter IBM verfügbar.

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 WebSphere MQ -Objekte. Verwenden Sie daher nicht dieselben Attribute wie Kanalobjekte. 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 148](#) sind die Ebenen zusammengefasst, die für die Erfassung von Kanalstatistikdaten verfügbar sind:

Ebene	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.

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.

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.

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

Statistikdaten gelten nicht für automatisch definierte Clustersenderkanäle.

QMGR

Die Erfassung statistischer Daten für automatisch definierte Clustersenderkanäle wird durch den Wert des WS-Managerattributs STATCHL gesteuert.

Dies ist der Standardwert.

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 Statistiken

Statistiken 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 Statistiken angibt. Das Standardstatistikintervall beträgt 1800 Sekunden (30 Minuten). Verwenden Sie zum Ändern des Statistikintervalls 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 Statistiken aufzeichneten Informationen zu verwenden, führen Sie eine Anwendung wie das Beispielprogramm **amqsmn** aus, um die aufzeichneten Informationen in ein geeignetes Format umzuwandeln.

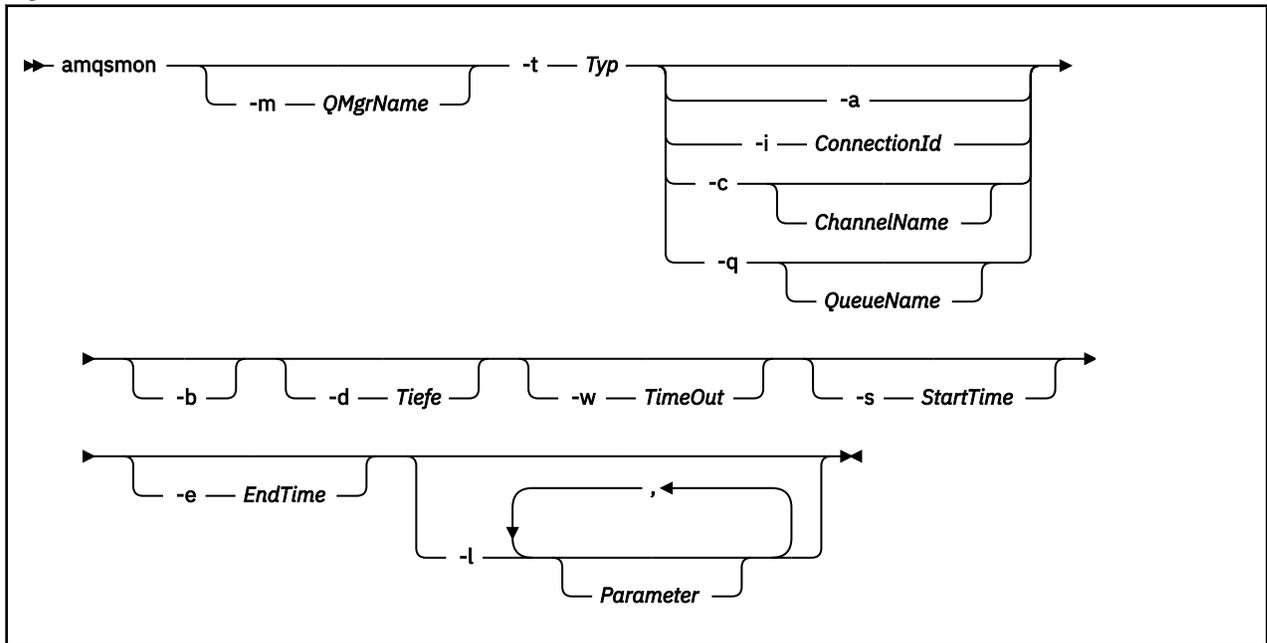
Abrechnungs- und Statistiken werden in die Systemabrechnungs- und Statistikwarteschlangen geschrieben. **amqsmn** ist ein Beispielprogramm, das im Lieferumfang von WebSphere MQ enthalten ist und Nachrichten aus den Abrechnungs- und Statistikwarteschlangen verarbeitet und die Informationen auf dem Bildschirm in lesbarer Form anzeigt.

Da **amqsmn** ein Beispielprogramm ist, können Sie den bereitgestellten Quellcode als Vorlage zum Schreiben Ihrer eigenen Anwendung verwenden, um Abrechnungs- oder Statistiken zu verarbeiten, oder den **amqsmn**-Quellcode ändern, um Ihre eigenen speziellen Anforderungen zu erfüllen.

amqsmn (Formatierte Überwachungsdaten anzeigen)

Verwenden Sie das Beispielprogramm **amqsmn**, um die in Abrechnungs- und Statistiken enthaltenen Informationen in einem lesbaren Format anzuzeigen. Das Programm **amqsmn** liest Abrechnungsnachrichten aus der Abrechnungswarteschlange SYSTEM.ADMIN.ACCOUNTING.QUEUE. und liest Statistiken 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 ist, werden die Abrechnungs- und Statistiknachrichten, aus denen die Datensätze stammen, gelöscht. Da Abrechnungs- und Statistiknachrichten auch Datensätze aus anderen Warteschlangen enthalten, können unsichtbare Datensätze gelöscht werden, wenn *-b* nicht angegeben ist.

-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 unsichtbare Datensätze gelöscht werden, wenn *-b* nicht angegeben ist.

Dieser Parameter ist nur beim Anzeigen von Statistiknachrichten verfügbar (*-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 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 unsichtbare Datensätze gelöscht werden, wenn *-b* nicht angegeben ist.

-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 amqsmn beendet, wenn keine Nachrichten mehr verarbeitet werden müssen.

-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 [amqsmo-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.

amqsmo-Beispiele

Verwenden Sie diese Seite, um Beispiele für die Ausführung des Beispielprogramms 'amqsmo' (Formatierte Überwachungsdaten anzeigen) anzuzeigen.

1. Mit dem folgenden Befehl werden alle MQI-Statistiken vom WS-Manager saturn.queue.manager angezeigt:

```
amqsmo -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, 0, 0]
OpenFailCount: [0, 0, 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, 0, 0]
CloseFailCount: [0, 0, 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, 0, 0]
InqFailCount: [0, 31, 0, 0, 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, 0, 0]
SetFailCount: [0, 0, 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. 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, 0]
OpenFailCount: [0, 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, 0]
CloseFailCount: [0, 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, 0]
InqFailCount: [0, 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, 0]
SetFailCount: [0, 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 -a -l 7006,3024
```

Die Ausgabe dieses Befehls folgt:

```
ConnectionId: x'414d514354524556312020202020208d0b374203090020'  
ApplicationName: 'runmqsc'  
  
ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'  
ApplicationName: 'amqsput'  
  
ConnectionId: x'414d514354524556312020202020208d0b3742010c0020'  
ApplicationName: 'runmqsc'  
  
ConnectionId: x'414d514354524556312020202020208d0b3742010d0020'  
ApplicationName: 'amqsput'  
  
ConnectionId: x'414d514354524556312020202020208d0b3742150d0020'  
ApplicationName: 'amqsget'  
  
5 Records Processed.
```

Nachrichtenreferenz für Abrechnung und Statistik

Verwenden Sie diese Seite, um eine Übersicht über das Format der Abrechnungs- und Statistiknachrichten und die in diesen Nachrichten zurückgegebenen Informationen zu erhalten.

Abrechnungs- und Statistiknachrichten sind Standardnachrichten von WebSphere MQ, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Die Nachrichtendaten enthalten Informationen zu den MQI-Operationen, die von WebSphere MQ -Anwendungen ausgeführt werden, oder Informationen zu den Aktivitäten in einem WebSphere 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 ¹
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 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
<p>Anmerkung:</p> <p>1. Die angezeigten Parameter sind die Parameter, die für eine MQI-Abrechnungsnachricht zurückgegeben werden. Die tatsächlichen Abrechnungs- oder Statistikenachrichtendaten hängen von der Nachrichtenkategorie ab.</p>		

Abrechnungs- und Statistiknachricht MQMD (Nachrichtendeskriptor)

Verwenden Sie diese Seite, um die Unterschiede zwischen dem Nachrichtendeskriptor von Abrechnungs- und Statistiknachrichten und dem Nachrichtendeskriptor von Ereignisnachrichten zu verstehen.

Die Parameter und Werte im Nachrichtendeskriptor der Abrechnungs- und Statistiknachricht sind dieselben wie im Nachrichtendeskriptor von Ereignisnachrichten, mit der folgenden Ausnahme:

Format

Beschreibung:	Formatname der Nachrichtendaten.
Datentyp:	MQCHAR8.
Wert:	MQFMT_ADMIN Admin-Nachricht.

Einige der Parameter, die im Nachrichtendeskriptor der Abrechnungs- und Statistiknachricht 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 Statistiknachrichten

Die Nachrichtendaten in Abrechnungs- und Statistiknachrichten basieren auf dem programmierbaren Befehlsformat (PCF), das in PCF-Befehlsabfragen und -antworten verwendet wird. Die Nachrichtendaten in Abrechnungs- und Statistiknachrichten bestehen aus einem PCF-Header (MQCFH) und einem Accounting- oder Statistikbericht.

Abrechnungs- und Statistiknachricht MQCFH (PCF-Header)

Der Nachrichtenheader von Abrechnungs- und Statistiknachrichten ist eine MQCFH-Struktur. Die Parameter und Werte in der Nachrichtenheader der Abrechnungs- und Statistiknachricht sind dieselben wie im Nachrichtenheader von Ereignisnachrichten mit den folgenden Ausnahmen:

Command

Beschreibung:	Befehls-ID. Gibt die Nachrichtenkategorie für die Abrechnung oder die Statistiknachricht an.
Datentyp:	MQLONG.
Werte:	MQCMD_ACCOUNTING_MQI MQI-Abrechnungsnachricht. MQCMD_ACCOUNTING_Q Warteschlangenabrechnungsnachricht. MQCMD_STATISTICS_MQI MQI-Statistiknachricht. MQCMD_STATISTICS_Q Warteschlangenstatistiknachricht. MQCMD_STATISTICS_CHANNEL Kanalstatistiknachricht.

Version

Beschreibung:	Strukturversionsnummer.
Datentyp:	MQLONG.

Wert: **MQCFH_VERSION_3**
Version-3 für Abrechnungs-und Statistiknachrichten.

Abrechnungs-und Statistiknachrichtendaten

Der Inhalt der Nachrichtendaten für Abrechnungs-und Statistikdaten ist abhängig von der Kategorie der Abrechnungs-oder Statistiknachricht 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-Statistiknachricht

Die MQI-Statistik-Nachrichtendaten bestehen aus einer Reihe von PCF-Parametern, aber keine PCF-Gruppen.

Warteschlangenstatistiknachricht

Die Nachrichtendaten der Warteschlangenstatistik bestehen aus einer Reihe von PCF-Parametern und im Bereich von 1 bis 100 *QStatisticsData* -PCF-Gruppen.

Kanalstatistiknachricht

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.

Nachrichtename:	MQI-Abrechnungsnachricht.
Plattformen:	Alle außer WebSphere 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 WebSphere 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 WebSphere 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

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 geöffneten Objekte. 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:	<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 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 indexiert 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 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 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 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: 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 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_CTL5
Datentyp: MQCFIL
Zurückgegeben: Wenn verfügbar.

CtlFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQCTL-Anforderungen.

ID: MQIAMO_CTL5_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 indexiert ist (siehe [Referenznote 2](#)).

Hinweis: Nachrichten, die einen Warteschlangentalias 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 Byte, die an Subskribenten zugestellt werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).

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 WebSphere 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 WebSphere 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 WebSphere 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:

- QName*
- CreateDate*
- CreateTime*
- QType*
- QDefinitionType*
- OpenCount*
- OpenDate*
- OpenTime*
- CloseDate*
- CloseTime*
- PutCount*
- PutFailCount*
- Put1Count*
- Put1FailCount*
- PutBytes*
- PutMinBytes*
- PutMaxBytes*
- GetCount*
- GetFailCount*
- GetBytes*
- GetMinBytes*
- GetMaxBytes*
- BrowseCount*
- BrowseFailCount*
- BrowseBytes*
- BrowseMinBytes*
- BrowseMaxBytes*
- TimeOnQMin*
- TimeOnQAvg*
- TimeOnQMax*

Zurückgegeben: Immer

QName

Beschreibung: Der Name der Warteschlange.

ID: MQCA_Q_NAME

Datentyp: MQCFST

In PCF-Gruppe enthalten: *QAccountingData*

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: *QAccountingData*
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:	Die Häufigkeit, mit der diese Warteschlange von der Anwendung in diesem Intervall geöffnet wurde.
ID:	MQIAMO_OPENS
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
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: *QAccountingData*
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 indiziert ist (siehe Referenznote 2).

ID: MQIAMO_PUTS
Datentyp: MQCFIL
In PCF-Gruppe
enthalten: *QAccountingData*
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: *QAccountingData*
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 indiziert 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 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 Referenznote 2).
ID:	MQIAMO64_PUT_BYTES
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
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 indexiert ist (siehe Referenznote 2).
ID:	MQIAMO_PUT_MIN_BYTES
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
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 indexiert ist (siehe Referenznote 2).
ID:	MQIAMO_PUT_MAX_BYTES
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

GeneratedMsgCount

Beschreibung:	Die Anzahl der generierten Nachrichten. Generierte Nachrichten: <ul style="list-style-type: none">• Warteschlangenlänge-Hi-Ereignisse• Niedrige Ereignisse in der Warteschlange
ID:	MQIAMO_GENERATED_MSGS
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
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
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: *QAccountingData*
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 indexiert ist (siehe [Referenznote 2](#)).
ID: MQIAMO64_GET_BYTES
Datentyp: MQCFIL64
In PCF-Gruppe
enthalten: *QAccountingData*
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 indexiert ist (siehe [Referenznote 2](#)).
ID: MQIAMO_GET_MIN_BYTES
Datentyp: MQCFIL
In PCF-Gruppe
enthalten: *QAccountingData*
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 indexiert ist (siehe [Referenznote 2](#)).
ID: MQIAMO_GET_MAX_BYTES
Datentyp: MQCFIL
In PCF-Gruppe
enthalten: *QAccountingData*
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
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

BrowseFailCount

Beschreibung:	Die Anzahl der nicht erfolgreichen MQGET-Aufrufe ohne Löschen.
ID:	MQIAMO_BROWSES_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
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:	<i>QAccountingData</i>
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:	<i>QAccountingData</i>
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

CBCount

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: MQCFIN
Zurückgegeben: Wenn verfügbar.

CBFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQCB-Anforderungen.

ID: MQIAMO_CBS_FAILED
Datentyp: MQCFIN
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 indexiert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64_Q_TIME_MIN
Datentyp: MQCFIL64
In PCF-Gruppe enthalten: *QAccountingData*
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 indexiert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64_Q_TIME_AVG
Datentyp: MQCFIL64
In PCF-Gruppe enthalten: *QAccountingData*

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 [Referenznote 2](#)).

ID: MQIAMO64_Q_TIME_MAX

Datentyp: MQCFIL64

In PCF-Gruppe enthalten: *QAccountingData*

Zurückgegeben: Wenn verfügbar

MQI-Statistikdaten, Nachrichtendaten

Verwenden Sie diese Seite, um die Struktur einer MQI-Statistiknachricht anzuzeigen.

Nachrichtenname: MQI-Statistiknachricht.

Plattformen: Alle außer WebSphere 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 indiziert wird: <ul style="list-style-type: none">• MQDISCONNECT_NORMAL• MQDISCONNECT_IMPLICIT• MQDISCONNECT_Q_MGR
ID:	MQIAMO_DISCS.
Datentyp:	MQCFIL.
Zurückgegeben:	Wenn verfügbar.

OpenCount

Beschreibung:	Die Anzahl der erfolgreich geöffneten Objekte. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indiziert 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 indiziert 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 indiziert ist (siehe Referenznote 1).
ID:	MQIAMO_CLOSES.
Datentyp:	MQCFIL.
Zurückgegeben:	Wenn verfügbar.

CloseFailCount

Beschreibung:	Die Anzahl der erfolgreichen Versuche zum Schließen von Objekten. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indiziert ist (siehe Referenznote 1).
ID:	MQIAMO_CLOSES_FAILED.
Datentyp:	MQCFIL.
Zurückgegeben:	Wenn verfügbar.

InqCount

Beschreibung:	Die Anzahl der Objekte, die erfolgreich 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 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 Referenznote 2).
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 Referenznote 2).
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 Referenznote 2).
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 Referenznote 2).
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 [Referenznote 2](#)).

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.

CBCount

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 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 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 indexiert 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 WebSphere 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: Gruppierete 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: <ul style="list-style-type: none">• 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:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

QMaxDepth

Beschreibung:	Die maximale Warteschlangenlänge während des Überwachungszeitraums
ID:	MQIAMO_Q_MAX_DEPTH
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
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:	<i>QStatisticsData</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-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: *QStatisticsData*

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: *QStatisticsData*
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 [Referenznote 2](#).

ID: MQIAMO_GETS
Datentyp: MQCFIL
In PCF-Gruppe
enthalten: *QStatisticsData*
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: *QStatisticsData*
Zurückgegeben: Wenn verfügbar

GetBytes

Beschreibung: Die Anzahl der Byte, die in destruktiven Schreibanforderungen für persistente und nicht persistente Nachrichten gelesen werden. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe [Referenznote 2](#).

ID: MQIAMO64_GET_BYTES
Datentyp: MQCFIL64
In PCF-Gruppe
enthalten: *QStatisticsData*
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:	<i>QStatisticsData</i>
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:	<i>QStatisticsData</i>
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 WebSphere MQ die Warteschlange umgehen konnte, und nicht, wie oft eine Anwendung im Wartestatus war.
ID:	MQIAMO_MSGS_NOT_QUEUED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
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:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

PurgeCount

Beschreibung:	Die Anzahl der bereinigten Nachrichten.
ID:	MQIAMO_MSGS_PURGED
Datentyp:	MQCFIN

In PCF-Gruppe enthalten: *QStatisticsData*
Zurückgegeben: Wenn verfügbar

CBCount

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: MQCFIN
Zurückgegeben: Wenn verfügbar.

CBFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQCB-Anforderungen.

ID: MQIAMO_CBS_FAILED
Datentyp: MQCFIN
Zurückgegeben: Wenn verfügbar.

Nachrichtendaten der Kanalstatistik

Verwenden Sie diese Seite, um die Struktur einer Kanalstatistiknachricht anzuzeigen.

Nachrichtenname:	Kanalstatistiknachricht.
Plattformen:	Alle außer WebSphere 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: MQCHT_SENDER Senderkanal. MQCHT_SERVER Serverkanal. MQCHT_RECEIVER Empfängerkanal. MQCHT_REQUESTER Requesterkanal. MQCHT_CLUSRCVR Clusterempfängerkanal. MQCHT_CLUSSDR 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 Stapel, die vom Kanal verarbeitet wurden, die gesendet wurden, weil der Wert der Kanalattribute BATCHSZ oder BATCHLIM erreicht wurde.
ID: MQIAMO_FULL_BATCHES.
Datentyp: MQCFIN.
Zurückgegeben: Wenn verfügbar.

IncomplBatchCount

Beschreibung: Die Anzahl der vom Kanal verarbeiteten Stapel, 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 159
- „Nachrichtendaten der Warteschlangenabrechnung“ auf Seite 170
- „MQI-Statistikdaten, Nachrichtendaten“ auf Seite 181
- „Nachrichtendaten der Warteschlangenstatistik“ auf Seite 193
- „Nachrichtendaten der Kanalstatistik“ auf Seite 200

1. Dieser Parameter bezieht sich auf WebSphere MQ -Objekte. Dieser Parameter ist ein Array von Werten (MQCFIL oder MQCFIL64), die durch die folgenden Konstanten indexiert werden:

<i>Table 24. Feldgruppe nach Objekttyp indexiert</i>	
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 Authentifizierungsinformationsobjekte 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 WebSphere MQ -Nachrichten. Dieser Parameter ist ein Array von Werten (MQCFIL oder MQCFIL64), die durch die folgenden Konstanten indexiert 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 WebSphere 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.

Traceinformationen zur Anwendungsaktivität erfassen

Eine Tracenachricht der Anwendungsaktivität ist eine PCF-Nachricht. Sie können den Aktivitätstrace mit Hilfe einer Konfigurationsdatei konfigurieren. Zur Erfassung der Traceinformationen zur Anwendungsaktivität legen Sie das Warteschlangenmanagerattribut ACTVTRC fest. 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. Sie können den Aktivitätstrace unter Verwendung einer Konfigurationsdatei mit dem Namen `mqat.ini` konfigurieren.

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 ACTVCONO-Einstellungen außer Kraft gesetzt, die durch die Einstellungen in der Datei `mqat.ini` überschrieben werden.

Traceinträge 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 215.

Die einfachste Methode zur Anzeige der Inhalte von Tracenachrichten zur Anwendungsaktivität ist, das „Beispielprogramm 'amqsact'“ auf Seite 215 zu verwenden.

Vorgehensweise

1. „Festlegen von ACTVTRC zum Steuern der Erfassung von Aktivitätstraceinformationen“ auf Seite 207.
2. „Festlegen von MQCONNX-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen“ auf Seite 208.
3. „Aktivitätstraceverhalten mit mqat.ini konfigurieren“ auf Seite 208.
4. „Leistungswirkung des Trace der Anwendungsaktivität optimieren“ auf Seite 215.

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 Erfassung von API-Aktivitätstraceinformationen ist aktiviert.

OFF

Die Erfassung von API-Aktivitätstraceinformationen ist inaktiviert.

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 Struktur `MQCNO` überschrieben werden. Weitere Informationen finden Sie im Abschnitt „Festlegen von MQCONNX-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen“ auf Seite 208.

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 215 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 215.

Festlegen von MQCONNX-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen

Wenn das Warteschlangenmanagerattribut **ACTVCONO** auf **ENABLED** gesetzt ist, können Sie den Parameter **ConnectOpts** im Aufruf **MQCONNX** verwenden, um Anwendungsaktivitätsberichte auf Verbindungsbasis zu aktivieren bzw. zu inaktivieren. Diese Optionen überschreiben das Verhalten des Aktivitätstrace, das durch das Warteschlangenmanager-Attribut **ACTVTRC** definiert ist, und kann durch Einstellungen in der Aktivitätstracekonfigurationsdatei `mqt.ini` überschrieben 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 Attribut **QMGR ACTVCONO** auf **DISABLED** gesetzt ist, wird kein Fehler an die Anwendung zurückgegeben und die Aktivitätstraceerfassung wird durch die Warteschlangenmanagerattribute oder die Aktivitätstracekonfigurationsdatei `mqt.ini` definiert.

2. Setzen Sie den Parameter **ConnectOpts** im **MQCONNX**-Aufruf auf **MQCNO_ ACTIVITY_ TRACE_ ENABLED**.

Der Parameter **ConnectOpts** im **MQCONNX**-Aufruf kann die folgenden Werte haben:

MQCNO_ ACTIVITY_ TRACE_ DISABLED

Der Aktivitätstrace wird für die Verbindung inaktiviert.

MQCNO_ ACTIVITY_ TRACE_ ENABLED

Der Aktivitätstrace wird für die Verbindung aktiviert.

Anmerkung: Wenn in einer Anwendung für **MQCONNX** beide Einstellungen, **MQCNO_ ACTIVITY_ TRACE_ ENABLED** und **MQCNO_ ACTIVITY_ TRACE_ DISABLED**, aktiviert sind, schlägt der Aufruf mit dem Ursachencode **MQRC_ OPTIONS_ ERROR** fehl.

3. Stellen Sie sicher, dass diese Einstellungen für die Aktivitätstrace nicht durch Einstellungen in der Aktivitätstracekonfigurationsdatei `mqt.ini` überschrieben werden.

Weitere Informationen finden Sie im Abschnitt [„Aktivitätstraceverhalten mit `mqt.ini` konfigurieren“](#) auf Seite 208.

Nächste Schritte

Die einfachste Methode zur Anzeige der Inhalte von Tracenachrichten zur Anwendungsaktivität ist, das [„Beispielprogramm 'amqsact'“](#) auf Seite 215 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 215.

Aktivitätstraceverhalten mit `mqt.ini` konfigurieren

Das Verhalten des Aktivitätstrace wird mithilfe der Konfigurationsdatei `mqt.ini` konfiguriert. Diese Datei folgt dem gleichen Zeilengruppenschlüssel- und Parameterwert-Paar-Format wie die Dateien `mqs.ini` und `qm.ini`.

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 `mqat.ini` im Datenverzeichnis des Warteschlangenmanagers `C:\Program Files\IBM\WebSphere MQ\qmgrs\queue_manager_name`. Benutzer, die Anwendungen ausführen, für die ein Trace durchgeführt wird, benötigen die Berechtigung zum Lesen dieser Datei

Anmerkung: Bei Warteschlangenmanagern, die von IBM WebSphere MQ Version 7.1 oder einer früheren Version migriert wurden, fehlt die Datei `mqat.ini`. In diesen Fällen muss die Datei `mqat.ini` manuell erstellt werden und es müssen 660 Berechtigungen in der Datei festgelegt werden.

Die Syntaxregeln für das Format der Datei lauten wie folgt:

- 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 Zeilengruppennamen, 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 (**AppName** und **DebugPath**) werden jedoch von der Groß-/Kleinschreibung abhängig gemacht.

Zeilengruppenschlüssel

In der Konfigurationsdatei sind zwei Zeilengruppenschlüsseltypen zulässig: die Zeilengruppe `AllActivityTrace` und die Zeilengruppe `ApplicationTrace`.

Zeilengruppe 'AllActivityTrace'

Die Zeilengruppe `AllActivityTrace` definiert Einstellungen für den Aktivitätstrace, der auf alle IBM WebSphere MQ -Verbindungen angewendet wird, sofern sie nicht überschrieben werden.

Einzelne Werte in der Zeilengruppe `AllActivityTrace` können durch spezifischere Informationen in einer Zeilengruppe `ApplicationTrace` überschrieben werden.

Werden mehrere `AllActivityTrace`-Zeilengruppen angegeben, werden die Werte in der letzten Zeilengruppe verwendet. Parameter, die im ausgewählten `AllActivityTrace` fehlen, übernehmen Standardwerte. Parameter und Werte aus vorherigen `AllActivityTrace`-Zeilengruppen werden ignoriert

Zeilengruppe 'ApplicationTrace'

Die Zeilengruppe `ApplicationTrace` definiert Einstellungen, die auf einen bestimmten Namen und/oder Typ einer IBM WebSphere MQ -Verbindung angewendet werden können.

Diese Zeilengruppe enthält `AppName` -und `ApplClass` -Werte, die gemäß den Abgleichregeln verwendet werden, die in Verbindungsabgleichsregeln definiert sind, um festzustellen, ob die Zeilengruppe für eine bestimmte Verbindung gilt.

Parameter/Wert-Paare

In der folgenden Tabelle sind die Parameter/Wert-Paare aufgelistet, die in der Konfigurationsdatei für den Aktivitätstrace verwendet werden können.

Tabelle 26. Parameter/Wert-Paare, die in der Aktivitätstracekonfigurations-Konfigurationsdatei verwendet werden können

Name	Zeilengruppentyp	Werte (Standardwert in Fettdruck)	Beschreibung
Verfolgen	ApplicationTrace	EIN /AUS	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.
ActivityInterval	AllActivityTrace ApplicationTrace	0 -99999999 (0=off)	Zeitintervall in Sekunden zwischen Tracenachrichten. Der Aktivitätstrace verwendet keinen Zeitgeberthread, sodass die Tracenachricht nicht genau zu dem Zeitpunkt geschrieben wird, zu dem die Zeit verstrichen ist, sondern geschrieben wird, wenn die erste MQI-Operation nach Ablauf des Zeitintervalls ausgeführt wird. Wenn dieser Wert 0 ist, dann wird die Tracenachricht geschrieben, wenn die Verbindung getrennt wird (oder wenn die Aktivitätsanzahl erreicht wird).
ActivityCount	AllActivityTrace ApplicationTrace	0 -99999999 (0=off)	Die Anzahl der MQI-oder XA-Operationen zwischen Tracenachrichten. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung getrennt wird (oder wenn das Aktivitätsintervall abgelaufen ist).

Tabelle 26. Parameter/Wert-Paare, die in der Aktivitätstracekonfigurations-Konfigurationsdatei verwendet werden können (Forts.)

Name	Zeilengruppentyp	Werte (Standardwert in Fettdruck)	Beschreibung
TraceLevel	AllActivityTrace ApplicationTrace	LOW/ MEDIUM /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.
TraceMessage-Data	AllActivityTrace ApplicationTrace	0 -104 857 600 (100Mb)	Die Menge der Nachrichtendaten, die in Byte für MQGET-, MQPUT-, MQPUT1- und Callback-Operationen aufgezeichnet wurden.

Tabelle 26. Parameter/Wert-Paare, die in der Aktivitätstracekonfigurations-Konfigurationsdatei verwendet werden können (Forts.)

Name	Zeilengruppentyp	Werte (Standardwert in Fettdruck)	Beschreibung
ApplName	ApplicationTrace	Zeichenfolge (Erforderlicher Parameter-kein Standardwert)	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. Bei verteilten Plattformen ist dies nur der Dateinamensabschnitt von MQAXC.ApplName wird mit dem Wert in der Zeilengruppe abgeglichen. Zeichen links vom Pfadtrennzeichen rechts werden ignoriert, wenn der Vergleich durchgeführt wird. Für z/OS -Anwendungen wird der gesamte MQAXC.ApplName-Wert mit dem Wert in der Zeilengruppe abgeglichen. Am Ende des Werts ApplName kann ein einzelnes Platzhalterzeichen (*) 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.
ApplClass	ApplicationTrace	USER /MCA/INTERN/ALL	Die Klasse der Anwendung. In der folgenden Tabelle wird erläutert, wie die AppType -Werte IBM WebSphere MQ -Verbindungen entsprechen.

Die folgende Tabelle zeigt, wie die Werte für AppClass den Feldern APICallerType und APIEnvironment in der Kontextstruktur des Verbindungs-API-Exits entsprechen.

Tabelle 27. 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)
INTERN	MQXACT_EXTERNAL	MQXE_COMMAND_SERVER MQXE_MQSC	'runmqsc' und Befehlsserver
INTERN	MQXACT_INTERNAL	(Beliebiger Wert)	"vertrauenswürdige" und interne Anwendungen und Prozesse, z. B. amqzdmaa
ALL	(Beliebiger Wert)	(Beliebiger Wert)	Trace für alle Benutzer- und internen 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 **amqspuic** zu erstellen:

```

ApplicationTrace:
  ApplClass=MCA                                # Application type
                                              # Values: (USER | MCA | INTERNAL | ALL)
                                              # Default: USER
  ApplName=amqspuic    # Application name (may be wildcarded)
                                              # (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
                                              # Values: 0-99999999 (0=off)
                                              # Default: 0
  TraceLevel=MEDIUM                          # Amount of data traced for each operation
                                              # Values: LOW | MEDIUM | HIGH
                                              # Default: MEDIUM
  TraceMessageData=1000                      # Amount of message data traced
                                              # Values: 0-100000000
                                              # Default: 0

```

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 AllActivityangegebener Wert wird für die Verbindung verwendet, sofern der Wert nicht auch in einer Zeilengruppe ApplicationTrace auftritt und die Zeilengruppe die Übereinstimmungskriterien für die Verbindung erfüllt, die in den Punkten [2](#), [3](#) und [4](#) beschrieben sind.
2. Die ApplClass wird mit dem Typ der IBM WebSphere MQ -Verbindung abgeglichen. Wenn die ApplClass nicht mit dem Verbindungstyp übereinstimmt, wird die Zeilengruppe für diese Verbindung ignoriert.
3. Der Wert ApplName in der Zeilengruppe wird mit dem Dateinamensteil des Felds ApplName aus der API-Exitkontextstruktur (MQAXC) für die Verbindung abgeglichen. Der Dateinamensabschnitt wird von den Zeichen rechts vom letzten Pfadtrennzeichen (/ oder \) abgeleitet. Wenn die Zeilengruppe

ApplName ein Platzhalterzeichen (*) enthält, werden nur die Zeichen links neben dem Platzhalterzeichen mit der entsprechenden Anzahl Zeichen aus den Verbindungen ApplName 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 Wert der Zeilengruppe ApplName nicht mit der Verbindung ApplName übereinstimmt, wird die Zeilengruppe für diese Verbindung ignoriert.

4. Wenn mehrere Zeilengruppen mit den Verbindungen ApplName und ApplClass übereinstimmen, wird die Zeilengruppe mit dem spezifischsten ApplName verwendet. Der spezifischste ApplName ist als derjenige definiert, der die meisten Zeichen verwendet, um die Verbindungen ApplName abzugleichen. Wenn die INI-Datei beispielsweise eine Zeilengruppe mit ApplName= "FRE*" und eine andere Zeilengruppe mit ApplName= "FREE*" enthält, wird die Zeilengruppe mit ApplName= "FREE*" als beste Übereinstimmung für eine Verbindung mit ApplName= "path/FREEDOM" ausgewählt, weil sie vier Zeichen entspricht (während ApplName= "FRE*" nur drei Zeichen entspricht).
5. Wenn nach der Anwendung der Regeln in den Punkten 2, 3 und 4 mehrere Zeilengruppen vorhanden sind, die den Verbindungen ApplName und ApplClass entsprechen, werden die Werte der letzten Übereinstimmung verwendet und alle anderen Zeilengruppen werden ignoriert.

Beispiel für Anwendungsaktivität-Trace-Datei

Das folgende Beispiel zeigt, wie die Konfigurationsdaten in der INI-Datei des Aktivitätstrace angegeben sind. Dieses Beispiel wird als Beispiel mit dem Namen mqat.ini im C-Beispielverzeichnis geliefert (dasselbe Verzeichnis wie amqsact.c file).

```
AllActivityTrace:
  ActivityInterval=0          # Time interval between trace messages
                              # Values: 0-99999999 (0=off)
                              # Default: 0
  ActivityCount=0            # Number of operations between trace msgs
                              # Values: 0-99999999 (0=off)
                              # Default: 0
  TraceLevel=MEDIUM         # Amount of data traced for each operation
                              # Values: LOW | MEDIUM | HIGH
                              # Default: MEDIUM
  TraceMessageData=0        # Amount of message data traced
                              # Values: 0-100000000
                              # Default: 0

ApplicationTrace:
  ApplClass=USER             # Application type
                              # Values: (USER | MCA | INTERNAL | ALL)
                              # Default: USER
  ApplName=AppName*         # Application name (may be wildcarded)
                              # (matched to app name without path)
                              # Default: *
  Trace=OFF                  # Activity trace switch for application
                              # Values: ( ON | OFF )
                              # Default: OFF
ActivityInterval=0          # Time interval between trace messages
                              # Values: 0-99999999 (0=off)
                              # Default: 0
  ActivityCount=0            # Number of operations between trace msgs
                              # Values: 0-99999999 (0=off)
                              # Default: 0
  TraceLevel=MEDIUM         # Amount of data traced for each operation
                              # Values: LOW | MEDIUM | HIGH
                              # Default: MEDIUM
  TraceMessageData=0        # Amount of message data traced
                              # Values: 0-100000000
                              # Default: 0
```

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 215.

Leistungswirkung des Trace der Anwendungsaktivität optimieren

Die Aktivierung der Anwendungsaktivitätstrace kann zu einer LeistungseinbuÙe führen. Dies kann dadurch reduziert werden, dass nur die Anwendungen, die Sie benötigen, die Anzahl der Anwendungen, die die Warteschlange in die Warteschlange stellen, und durch die Optimierung von **ActivityInterval**, **ActivityCount** und **TraceLevel** in `mqat.ini` erhöht 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 voreingestellten Werte von **ActivityInterval**, **ActivityCount** und **TraceLevel** in der `mqat.ini`-Datei geben einen Standardsaldo von Details und Leistung an. 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 dazu eine anwendungsspezifische ApplicationTrace-Zeilengruppe in `mqat.ini` oder indem Sie die Anwendung so ändern, dass `MQCNO_ACTIVITY_TRACE_ENABLED` im Optionsfeld in der **MQCNO**-Struktur in einem `MQCONN`-Aufruf angegeben wird. Siehe „Aktivitätstraceverhalten mit `mqat.ini` konfigurieren“ auf Seite 208 und „Festlegen von `MQCONN`-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen“ auf Seite 208.

- Überprüfen Sie vor dem Starten des Trace, ob mindestens eine Anwendung aktiv ist und bereit ist, die Aktivitätstrace-Nachrichtendaten aus dem `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 **TraceLevel** in der `mqat.ini`-Datei fest, um die erforderliche Mindestmenge an Daten zu erfassen.

`TraceLevel=LOW` hat die geringste Auswirkung auf die Messaging-Leistung. Weitere Informationen finden Sie im Abschnitt „Aktivitätstraceverhalten mit `mqat.ini` konfigurieren“ auf Seite 208.

- Optimieren Sie die **ActivityCount**- und **ActivityInterval**-Werte in `mqat.ini`, um die Anzahl der generierten Aktivitätstrace-Nachrichten anzupassen.

Wenn Sie einen Trace für mehrere Anwendungen erstellen, werden die Aktivitätstracenachrichten möglicherweise schneller erstellt, als sie aus `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

Beispielprogramm 'amqsact'

amqsact formatiert Tracenachrichten zur Anwendungsaktivität für Sie und wird mit WebSphere MQ bereitgestellt.

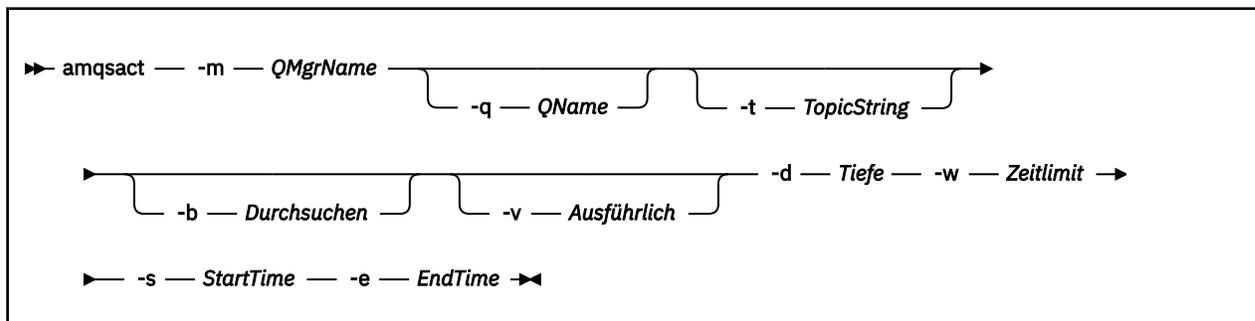
Das kompilierte Programm befindet sich im Verzeichnis 'samples':

- Auf UNIX and Linux `MQ_INSTALLATION_PATH/samp/bin`
- Auf 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: '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: FFFFFFFB5FFFFFFF6 4251 2000 FFFFFFFE601
QueueManager: 'TESTQM'
Completion Code: MQCC_OK
Reason Code: 0

```

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 Anwendungsaktivitäten sind Standardnachrichten von IBM WebSphere MQ, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Die Nachrichtendaten enthalten Informationen zu den MQI-Operationen, die von IBM WebSphere MQ-Anwendungen ausgeführt werden, oder Informationen zu den Aktivitäten in einem IBM WebSphere MQ-System.

Nachrichtendeskriptor

- Eine MQMD-Struktur

Nachrichtendaten

- Ein PCF-Header (MQCFH)
- Tracenachrichtendaten der Anwendungsaktivität, die immer zurückgegeben werden
- Tracenachrichtendaten der Anwendungsaktivität, die betriebsspezifisch 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

HostName

Beschreibung: Der Hostname der Maschine, auf der der Warteschlangenmanager ausgeführt wird.

ID: MQCACF_HOST_NAME

Datentyp: MQCFST

IntervalStartDate

Beschreibung: Das Datum des Startzeitbeginns 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: Befehlsebene IBM WebSphere MQ
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_OTHER MQXE_MCA MQXE_MCA_SVRCONN MQXE_COMMAND_SERVER MQXE_MQSC
ID: MQIACF_API_ENVIRONMENT
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

Pointer Size

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

Platform

Beschreibung: Die Plattform, auf der der WS-Manager ausgeführt wird. Der Wert ist einer der MQPL_*-Werte.
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 dann in der Gruppe AppActivityData für die Operation enthalten, wenn die Tracefunktion für "mittlere" oder "hohe" Aktivitäten 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 dann in der Gruppe AppActivityData für die Operation enthalten, wenn die "hohe" Tracefunktion für 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

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.

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) Anmerkung: 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

Expiry

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

Priority

Beschreibung:	Nachrichtenpriorität
---------------	----------------------

PCF-Parameter: MQIACF_PRIORITY
Tracestufe: 2
Typ MQCFIN

Persistence

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

ObjectName

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. 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 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)

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)

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

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

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)

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

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)
Anmerkung: 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

ObjectName

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

Expiry

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

Priority

Beschreibung: Nachrichtenpriorität
PCF-Parameter: MQIACF_PRIORITY
Tracestufe: 2
Typ: MQCFIN

Persistence

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)

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

IntAttrrs

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

CharAttrrs

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

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

ObjectName

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 >= 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 >= 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 >= 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 >= 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

ResolvedLocalQName¹²

Beschreibung: Enthält den Namen der lokalen Warteschlange, nachdem die Namensauflösung ausgeführt wurde. (z. B. für ferne Warteschlangen ist dies 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

ResolvedLocalQMgrName¹²

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

ResolvedQName¹²

Beschreibung: Der Warteschlangenname 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

ResolvedQMgrName¹²

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

PCF-Gruppenheader der Verteilerliste der Anwendungsaktivität

Wenn die Funktion MQOPEN eine Verteilerliste öffnet, enthält die MQOPEN-Parameter eine AppActivity-DistList-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:

MQCFGR, Feld	Wert	Beschreibung
Typ	MQCFT_GROUP	
StrucLength	Länge der MQCFGR-Struktur in Byte	
Parameter	MQGACF_APP_DIST_LIST	Verteilerlistengruppenparameter
ParameterCount	4	Die Anzahl der Parameterstrukturen im Anschluss an die MQCFGR-Struktur, die in dieser Gruppe enthalten sind.

ObjectName

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 ObjectName angegebene Warteschlange definiert ist.
PCF-Parameter:	MQCACF_OBJECT_Q_MGR_NAME
Tracestufe:	2

¹ 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.

² Der Parameter "ResolvedLocalQName" ist nur enthalten, wenn er sich vom Parameter "ResolvedQName" unterscheidet.

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 erhalten, 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

ObjectName

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

ResolvedLocalQName³

Beschreibung: Enthält den Namen der lokalen Warteschlange, nachdem die Namensauflösung ausgeführt wurde.

PCF-Parameter: MQCACF_RESOLVED_LOCAL_Q_NAME

Tracestufe: 2

Typ: MQCFST

ResolvedLocalQMgrName³

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

Expiry

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

Priority

Beschreibung: Nachrichtenpriorität
PCF-Parameter: MQIACF_PRIORITY
Tracestufe: 2
Typ: MQCFIN

Persistence

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.
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_STRUCTURE
 Tracestufe: 3
 Typ: MQCFBS
 Länge: Die Länge der MQPMO-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

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 248. 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

³ Der Parameter "ResolvedLocalQName" ist nur enthalten, wenn er sich vom Parameter "ResolvedQName" unterscheidet.

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

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:	Feedback. 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

ObjectName

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)
Anmerkung: 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

ResolvedLocalQName⁴

Beschreibung: Enthält den Namen der lokalen Warteschlange, nachdem die Namensauflösung ausgeführt wurde.
PCF-Parameter: MQCACF_RESOLVED_LOCAL_Q_NAME
Tracestufe: 2
Typ: MQCFST

ResolvedLocalQMgrName⁴

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

Expiry

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

Priority

Beschreibung: Nachrichtenpriorität
PCF-Parameter: MQIACF_PRIORITY
Tracestufe: 2
Typ: MQCFIN

Persistence

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)

MQPUT1 AppActivityDistList PCF-Header-Struktur

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 248. Die PCF-

⁴ Der Parameter "ResolvedLocalQName" ist nur enthalten, wenn er sich vom Parameter "ResolvedQName" unterscheidet.

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

ObjectName

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: Feedback. 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

IntAttr

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.

CharAttr

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.

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

ObjectName

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.

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

Action

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

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)

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 dann in der Gruppe AppActivityData für die Operation enthalten, wenn die Tracefunktion für "mittlere" oder "hohe" Aktivitäten 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 dann in der Gruppe AppActivityData für die Operation enthalten, wenn die "hohe" Tracefunktion für 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: Markierungen
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: Markierungen
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: Markierungen
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: Markierungen
PCF-Parameter: MQIACF_XA_FLAGS
Tracestufe: 1
Typ: MQCFIN

XARetCode

Beschreibung: Rückkehrcode
PCF-Parameter: MQIACF_XA_RETCODE
Tracestufe: 1
Typ: MQCFIN

XACOMLETE

Die Anwendung hat die Funktion XACOMLETE 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: Markierungen
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: Markierungen
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: Markierungen
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: Markierungen
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: Markierungen
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: Markierungen
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: Markierungen
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:	Markierungen
PCF-Parameter:	MQIACF_XA_FLAGS
Tracestufe:	1
Typ:	MQCFIN

XARetCode

Beschreibung:	Rückkehrcode
PCF-Parameter:	MQIACF_XA_RETCODE
Tracestufe:	1
Typ:	MQCFIN

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 helfen, den stabilen Zustand ihres IBM WebSphere MQ -Systems zu verstehen. 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-Manager-Ebene 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-Manager-Ebene gesteuert.

Automatisch definierte Clustersenderkanäle sind keine WebSphere MQ -Objekte. Verwenden Sie daher nicht dieselben Attribute wie Kanalobjekte. 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 29 auf Seite 283](#) sind die Überwachungsstufen zusammengefasst, die für die Echtzeitüberwachung von Kanälen verfügbar sind:

Ebene	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 282](#)

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.

[Mit Warteschlangenmanagern arbeiten](#)

Zugehörige Tasks

[„Warteschlangen- und Kanalüberwachungsdaten anzeigen“ auf Seite 285](#)

Überwachungsinformationen in Echtzeit für eine Warteschlange oder einen Kanal können Sie in IBM WebSphere MQ Explorer oder mit dem entsprechenden MQSC-Befehl abrufen. 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.

Warteschlangen-und Kanalüberwachungsdaten anzeigen

Überwachungsinformationen in Echtzeit für eine Warteschlange oder einen Kanal können Sie in IBM WebSphere MQ Explorer oder mit dem entsprechenden MQSC-Befehl abrufen. 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. Um Echtzeitüberwachungsdaten für eine Warteschlange anzuzeigen, verwenden Sie entweder IBM WebSphere MQ Explorer oder den MQSC-Befehl `DISPLAY QSTATUS` und geben Sie den optionalen Parameter `MONITOR` an.
2. Um Echtzeitüberwachungsdaten für einen Kanal anzuzeigen, verwenden Sie entweder den IBM WebSphere MQ Explorer oder den MQSC-Befehl `DISPLAY CHSTATUS` und geben Sie den optionalen Parameter `MONITOR` an.

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.T0.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.T0.QM2) MONITOR
```

Die Überwachungsfelder und die Überwachungsstufe des Senderkanals QM1.T0.QM2 werden wie folgt angezeigt:

```
CHSTATUS(QM1.T0.QM2)
XMITQ(Q1)
CONNAM(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)
RQMNAME(QM2)
```

Zugehörige Konzepte

„Real-time Monitoring“ auf Seite 282

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, wobei Sie statt <appltag> den APPLTAG-Wert aus der Ausgabe des vorherigen Befehls eingeben:

```
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 Anwendungsanwendung 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 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)
```

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 30 auf Seite 291](#) , um festzustellen, welche Aktion zu ergreifen ist:

<i>Tabelle 30. Substatus mit Statusbindung oder Anforderung angezeigt</i>		
MCA-Unterzustand initialisieren ¹	Antwortet MCA-Unterstatus ²	Anmerkungen
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 unter „ Feststellen, ob der Kanal Nachrichten schnell genug verarbeiten kann “ auf Seite 293.
	CHADEXIT	Der Exit für die Kanalautodefinition wird gerade ausgeführt. Weitere Informationen finden Sie unter „ Feststellen, ob der Kanal Nachrichten schnell genug verarbeiten kann “ auf Seite 293.
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 unter „ Feststellen, ob der Kanal Nachrichten schnell genug verarbeiten kann “ auf Seite 293.
SERIALIZE	SERIALIZE	Dieser Unterzustand gilt nur für Kanäle mit der Disposition SHARED.
NETZVERBIND		Dieser Unterzustand wird angezeigt, wenn die Verbindung aufgrund einer falschen Netzkonfiguration verzögert wird.
SSLHANDSHAKE	SSLHANDSHAKE	Ein SSL-Handshake besteht aus einer Reihe gesendeter und empfangener Byte. 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.

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 287 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:

MSG

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 Bestimmungswarteschlange 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.

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. Wenn der Kanal einen Stapel gestartet hat, der noch nicht abgeschlossen ist, wie durch einen Wert ungleich null in `CURMSG` angegeben wird, wartet der Kanal möglicherweise auf das andere Ende des Kanals, um den Batch zu bestätigen. Sehen Sie sich das Feld `SUBSTATE` in der Ausgabe an, und lesen Sie die Informationen in [Tabelle 31](#) auf Seite 292:

AbsenderSUBSTATE	Empfänger-SUBSTATE	Anmerkungen
MQGET	RECEIVE	Normale Zustände eines Kanals in Ruhe.

Tabelle 31. Sender-und Empfänger-MCA-Unterzustände (Forts.)

AbsenderSUBSTATE	Empfänger-SUBSTATE	Anmerkungen
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

Auf dieser Seite können Sie einige Gründe anzeigen, warum die Ausführung eines Batch sehr lange dauern kann.

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 folgenden Faktoren können beeinflussen, wie lange der Senderkanal wartet:

Prozedur

- Überprüfen Sie, ob das Netz langsam ist.
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.
- 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 287 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.T0.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.T0.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

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.T0.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.T0.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.T0.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 289 beschriebenen Tasks, um die Probleme mit den gefundenen Kanälen zu diagnostizieren, die die Erstellung des Builds verursachen.

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 WebSphere MQ die Möglichkeit, die Übertragungswarteschlange eines oder mehrerer Kanäle, die nicht aktiv sind, mit dem Befehl **runswchl** zu wechseln.

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
- Mit dem Befehl DISPLAY CLUSQMGR können Sie die Übertragungswarteschlange abfragen, die momentan von jedem Clustersenderkanal verwendet wird.
- Führen Sie den Befehl **runswchl** im Abfragemodus aus, um den Umschaltstatus 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.

Windows -Leistungsüberwachung

In WebSphere MQ Version 7.0 und früheren Versionen war es möglich, die Leistung lokaler Warteschlangen auf Windows -Systemen mithilfe der Windows-Leistungsüberwachung zu überwachen. Ab WebSphere MQ Version 7.1 ist diese Methode der Leistungsüberwachung nicht mehr verfügbar.

Sie können Warteschlangen mit den im Abschnitt [„Real-time Monitoring“](#) auf Seite 282 beschriebenen Methoden auf allen unterstützten Plattformen überwachen.

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 zu geplanten Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zum Abrufen der Services von IBM WebSphere MQ zu schreiben.

Diese Informationen können jedoch auch Angaben über Diagnose, Bearbeitung und Optimierung enthalten. Die Informationen zu Diagnose, Bearbeitung und Optimierung sollten Ihnen bei der Fehlerbehebung für die Anwendungssoftware helfen.

Wichtig: Verwenden Sie diese Diagnose-, Änderungs- und Optimierungsinformationen nicht als Programmierschnittstelle, da sie Änderungen unterliegen.

Marken

IBM, das IBM Logo, ibm.com, sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml. 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: