

9.4

Überwachung und Leistung für IBM MQ

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 403 gelesen werden.

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

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

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

Inhaltsverzeichnis

Überwachung und Leistung.....	5
IBM MQ-Netz überwachen.....	5
OpenTelemetry -Integration.....	5
Statusprüfungsverhalten des Warteschlangenmanagers.....	6
Ereignisüberwachung.....	9
Nachrichtenüberwachung.....	62
Abrechnungs-und Statistiknachrichten.....	144
Anwendungsaktivitätstrace.....	212
Systemthemen für Überwachung und Aktivitätstrace.....	301
Real-time Monitoring.....	312
Cluster überwachen.....	325
Lastausgleich von Anwendungen überwachen.....	329
Monitoring performance and resource usage on z/OS.....	331
IBM MQ-Netz optimieren.....	388
Tuning Client-und Serververbindungskanäle.....	388
Verteilte Publish/Subscribe-Netze optimieren.....	390
Reduzieren der Anzahl unerwünschter Themen in der Themenstruktur.....	400
Aspera gateway kann die Leistung in Netzen mit hoher Latenz verbessern.....	402
Bemerkungen.....	403
Informationen zu Programmierschnittstellen.....	404
Marken.....	405

IBM MQ Überwachung und Leistung

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

Informationen zu diesem Vorgang

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

IBM MQ-Netz überwachen

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

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

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

V 9.4.0

ALW

OpenTelemetry -Integration

Sie können IBM MQ in ein OpenTelemetry -Tracesystem integrieren.

OpenTelemetry -Traceerstellung

Mit der OpenTelemetry -Traceerstellung können Sie beobachten, wie sich Anwendungen innerhalb eines Datenflusses verhalten. Der Datenfluss kann und kann häufig mehrere verschiedene Anwendungen umfassen. Die Traceerstellung kann Ihnen die gesamte Journey zeigen und Einblicke in das Verhalten jeder einzelnen Anwendung bieten. IBM MQ stellt einen Traceservice bereit, mit dem Sie ein OpenTelemetry -Tracesystem integrieren können.

Der IBM MQ OpenTelemetry -Traceservice wird als IBM MQ -API-Exit implementiert. Es wird als IBM -Unterstützungsprogramm bereitgestellt, d. h., Sie haben die Berechtigung zur Verwendung des Traceservice und erhalten Unterstützung als Teil Ihrer IBM MQ -Berechtigung. Beachten Sie, dass Ihre IBM MQ -Berechtigung nur die Verwendung der Komponente IBM MQ Tracing-Exit des IBM Instana -Unterstützungsprogramms zulässt.

Probleme, die sich aus der Verwendung des Exits ergeben, müssen dem IBM MQ Support gemeldet werden, es sei denn, Sie verwenden den Trace-Exit mit einem IBM Instana Überwachungssystem und IBM Instana -Berechtigung.

Der Exit kann hier heruntergeladen werden: <https://ibm.biz/mqinstanaexit>.

IBM MQ Open Telemetry-Traceservice installieren und konfigurieren

Ausführliche Informationen zur Installation und Konfiguration der IBM MQ -Tracefunktion finden Sie hier: [IBM MQ Tracefunktion](#).

Ausführliche Informationen zu den unterstützten Plattformen für den IBM Instana IBM MQ -Exit finden Sie hier: [Unterstützte Plattformen für lokale IBM MQ](#).

Multi

Statusprüfungsverhalten des Warteschlangenmanagers

Der Warteschlangenmanager führt regelmäßige Statusprüfungen durch, um eine stabile und zuverlässige Leistung sicherzustellen. In diesem Abschnitt werden einige Statusprüfungen beschrieben, die der Warteschlangenmanager durchführt, und es wird erläutert, wie sie auf der Basis von Umgebungsanforderungen konfiguriert werden.

In den meisten Umgebungen ist die Standardkonfiguration geeignet und es ist nicht erforderlich, die Häufigkeit dieser Prüfungen zu ändern. Selbst bei Verwendung der Standardeinstellungen kann es hilfreich sein, das Verhalten des Warteschlangenmanagers zu verstehen, wenn ein Problem erkannt wird, und die Umgebungsprobleme zu verstehen, die zum Fehlschlagen einer Prüfung führen können. In diesem Abschnitt werden einige dieser Verhaltensweisen erläutert.



Vorsicht: Nehmen Sie keine Änderungen an der Häufigkeit dieser Prüfungen vor, es sei denn, Sie werden vom IBM Support dazu aufgefordert.

Die verschiedenen Komponenten des Warteschlangenmanagers verwenden verschiedene Methoden zum Erkennen und Beheben von Inkonsistenzen. Dieser Abschnitt soll nicht alle derartigen Mechanismen beschreiben. IBM MQ -Prozesse verwenden beispielsweise verschiedene Mechanismen, um sicherzustellen, dass andere Prozesse, von denen sie abhängig sind, weiterhin ausgeführt werden. Die beschriebenen Verhaltensweisen sind diejenigen, die vom Ausführungscontroller in regelmäßigen Abständen vorgenommen werden, um Umgebungsbedingungen oder andere unerwartete Situationen zu erkennen. (Der Ausführungscontroller ist der primäre IBM MQ -Prozess, der die meisten anderen Warteschlangenmanagerprozesse startet und verwaltet.) Da es sich um regelmäßige Prüfungen handelt, werden sie in bestimmten Intervallen ausgeführt, die bis zu einem gewissen Grad geändert werden können, indem die entsprechenden Optimierungsparameter festgelegt werden.

Einige der beschriebenen Prüfungen werden von einem dedizierten Thread für die Statusprüfung durchgeführt. Wenn bei der Statusprüfung selbst ein Problem festgestellt wird, wird eine Warnung [AMQ5066](#) in die Fehlerprotokolle des Warteschlangenmanagers geschrieben.

The behaviors that are described in this topic are subject to change in future releases, for example if a different default value is observed to be more stable on a particular platform or configuration.

Allgemeine Statusprüfungen

Der Warteschlangenmanager führt in regelmäßigen Intervallen verschiedene Prüfungen durch. Standardmäßig werden diese Prüfungen alle 10 Sekunden durchgeführt (in einigen Fällen ermöglicht die Prüfung zwei Zyklen, bevor ein Fehler gemeldet wird, was zu einem 20-Sekunden-Intervall für solche Prüfungen führt). Während der Prüfungen stellt der Warteschlangenmanager sicher, dass die verschiedenen Prozesse, die als Teil des Warteschlangenmanagers ausgeführt werden, weiterhin aktiv sind. Bei einem nativen HA-Warteschlangenmanager wird geprüft, ob der WS-Manager erfolgreich Daten auf den Standby-Instanzen repliziert.

If a critical check fails at this time (for example, if the amqzmc0 process is no longer running) the queue manager will be unable to continue running. Die meisten Prüfungen werden jedoch durchgeführt, um Systemressourcen zu bereinigen, die nicht mehr benötigt werden, und können einfach dazu führen, dass eine Nachricht in die Fehlerprotokolle des Warteschlangenmanagers geschrieben wird.

In den meisten Fällen ist es nicht erforderlich, die Häufigkeit dieser allgemeinen Statusprüfungen zu ändern. Die meisten Ereignisse im Warteschlangenmanager oder in der Betriebsumgebung werden sofort erkannt, ohne dass der allgemeine Statusprüfungsprozess sie erkennen muss. Dieser Prozess dient als regelmäßige Prüfung auf alles, was an anderer Stelle im Warteschlangenmanager nicht erkannt wurde. Bei Bedarf können Sie die Häufigkeit mit dem Optimierungsparameter **ECHearTBeatLen** konfigurieren. Der Mindestwert ist 10000 Millisekunden (10 Sekunden). Der Maximalwert beträgt 60000 Millisekunden (60 Sekunden). Wenn dieser Wert auf den Maximalwert 60000 gesetzt ist, kann dies zu einer Verzögerung von zwei Minuten für bestimmte Prüfungen führen.

Prüft, ob der Protokollfortschritt ausgeführt wird.

Die Prüfungen des Warteschlangenmanagers, die in das Protokoll geschrieben werden, werden mit einer angemessenen Geschwindigkeit durchgeführt. Dies ist keine Prüfung, ob die Leistung der Protokollfunktion optimal ist, sondern wurde entwickelt, um Bedingungen zu erkennen, die weitere Aufmerksamkeit erfordern. Dies ist beispielsweise der Fall, wenn die Platte, auf der die Protokolldateien gespeichert werden, besonders langsam ist oder wenn der WS-Manager nicht genügend CPU-Zeit in einer containerisierten Umgebung erhält, um seine gesamte Arbeit auszuführen.

Wenn diese Prüfung fehlschlägt, hängt die vom Warteschlangenmanager ausgeführte Aktion vom Typ des verwendeten Warteschlangenmanagers ab:

- Auf einem Warteschlangenmanager ohne Hochverfügbarkeit:
 - Ein FDC `xecL_W_PERFORMANCE_BOTTLENECK` wird geschrieben. Dies kann als Hinweis darauf verwendet werden, dass ein Teil des Systems möglicherweise weitere Aufmerksamkeit benötigt. Der Warteschlangenmanager bleibt aktiv. Wenn die FDCs `xecL_W_PERFORMANCE_BOTTLENECK` im Fehlerverzeichnis angezeigt werden, müssen Sie möglicherweise mit Ihrem Speicher- oder Plattformteam zusammenarbeiten, um zu verstehen, ob die zugrunde liegenden Systemressourcen für die Ausführung von IBM MQ ausreichen. Wenn IBM MQ in Containern auf überbelegten Knoten ausgeführt wird, erhält IBM MQ möglicherweise nicht genügend geplante CPU-Zeit für die gesamte Messaging-Workload.
 - Ab IBM MQ 9.3.0 wird eine Warnung `AMQ5068W` in die Fehlerprotokolle des Warteschlangenmanagers geschrieben und es wird keine FDC `xecL_W_PERFORMANCE_BOTTLENECK` geschrieben. Wenn `AMQ5068W`-Nachrichten in den Protokollen angezeigt werden, kann es erforderlich sein, mit Ihrem Speicher- oder Plattformteam zusammenzuarbeiten, um zu verstehen, ob die zugrunde liegenden Systemressourcen für die Ausführung von IBM MQ ausreichen. Wenn IBM MQ in Containern auf überbelegten Knoten ausgeführt wird, erhält IBM MQ möglicherweise nicht genügend geplante CPU-Zeit für die gesamte Messaging-Workload. Wenn fünf `AMQ5068W`-Warnungen nacheinander geschrieben werden, wird eine `xecL_W_PERFORMANCE_BOTTLENECK` FDC geschrieben.
- Auf einem Warteschlangenmanager mit mehreren Instanzen:
 - Wenn die Statusprüfung des Protokollfortschritts fehlschlägt, wird die primäre Instanz beendet. Wenn eine Standby-Instanz verfügbar ist, wird sie gestartet und zur primären Instanz.
 - Ab IBM MQ 9.3.0 prüft die primäre Instanz vor der Beendigung, ob eine Standby-Instanz verfügbar ist. Wenn ein Standby-Warteschlangenmanager für die Übernahme durch die primäre Instanz verfügbar ist, wird er beendet. Außerdem wird eine Warnung `AMQ5068W` in die Fehlerprotokolle des Warteschlangenmanagers geschrieben.
- Auf einem nativen HA-Warteschlangenmanager verhält sich diese Prüfung auf dieselbe Weise wie ein Nicht-HA-Warteschlangenmanager.
- Auf einem RDQM (replizierter Datenwarteschlangenmanager) verhält sich diese Prüfung genauso wie ein Warteschlangenmanager ohne Hochverfügbarkeit.

Es ist möglich, dass ein Problem mit dem Fortschritt des IBM MQ -Protokolls durch ein Leistungsproblem im Warteschlangenmanager selbst verursacht wird.

Diese Prüfung erfolgt standardmäßig alle 60 Sekunden, obwohl der WS-Manager zwei Zyklen lang wartet, bevor er eine Aktion ausführt. Dies bedeutet, dass bei den Standardeinstellungen zwei Minuten vergehen müssen, bis der Warteschlangenmanager eine Fehlermeldung geschrieben hat (oder bei einem HA-Warteschlangenmanager übernommen wurde).

In den meisten Fällen ist das Standardverhalten auch dann geeignet, wenn das Dateisystem langsam ist oder dem Warteschlangenmanager eine geringe CPU-Zeit zugeordnet wird, weil andere Prüfungen wie Dateisperren (siehe „Prüft, ob Dateisperren noch bestehen“ auf Seite 8) und der Betrieb des Basisdateisystems zu einem Failover der primären Instanz führen, bevor diese Prüfung durchgeführt wird. Falls erforderlich, kann die Häufigkeit dieser Prüfung mit dem Optimierungsparameter **LivenessHeartBeatLen** konfiguriert werden. Der Maximalwert, auf den er konfiguriert werden kann, ist 600 Sekunden (10 Minuten). Der Mindestwert 0 bewirkt, dass die Prüfung vollständig inaktiviert wird. Bei einem Warteschlangenmanager ohne Hochverfügbarkeit ist die einzige Auswirkung der Prüfung eine zusätzliche Warnung in den Fehlerprotokollen des Warteschlangenmanagers. Bei einem Warteschlangenmanager mit mehreren Instanzen können Sie **LivenessHeartBeatLen** so konfigurieren, dass eine primäre Instanz des Warteschlangenmanagers schneller (durch Verringerung des Werts) oder langsam (durch Erhöhung des Werts) ausfällt. Eine Erhöhung des Werts, um die Häufigkeit der Protokollfortschrittsüberprüfung zu verringern, kann nützlich sein, wenn in Ihrer Umgebung gelegentlich eine sehr langsame Ein-/Ausgabe des Dateisystems auftritt, Sie jedoch die Ausführung der primären Instanz des Warteschlangenmanagers bevorzugen. Dies kann nützlich sein, wenn Sie über Anwendungen verfügen, die nicht so konzipiert sind, dass sie die Verbindung zur Standby-Instanz automatisch wiederherstellen und einen manuellen Eingriff erfordern, um sie erneut zu starten.

Anmerkung: Wenn die **ECHearHeartBeatLen** erhöht wurde, wirkt sich dies auf die Ablaufsteuerung der **LivenessHeartBeatLen** -Prüfungen aus. Protokollfortschrittsprüfungen werden durchgeführt, wenn die allgemeinen Statusprüfungen durchgeführt werden, sodass eine Verringerung der Häufigkeit der allgemeinen Statusprüfungen (**ECHearHeartBeatLen**) dazu führen kann, dass Protokollfortschrittsprüfungen bis 30 Sekunden nach dem konfigurierten **LivenessHeartBeatLen** durchgeführt werden.

Allgemeine Protokollierung der Dateisystemleistung

▶ V 9.4.0

Ab IBM MQ 9.4.0 wird eine Warnung [AMQ6729W](#) im Fehlerprotokoll des Warteschlangenmanagers ausgegeben, wenn allgemeine Lese-/Schreiboperationen in diesem Speicher länger als erwartet dauern. Sie können die **AMQ_IODELAY** -Umgebungsvariablen verwenden, um die Diagnose und die Ablaufsteuerungen zu optimieren, um entweder die Diagnose von Speicherleistungsproblemen zu erleichtern oder die Toleranz für solche Verzögerungen zu erhöhen. Weitere Informationen finden Sie unter [AMQ_IODELAY](#), [AMQ_IODELAY_INMS](#) und [AMQ_IODELAY_FFST](#).

Prüft, ob Dateisperren noch bestehen

Bei einem Warteschlangenmanager mit mehreren Instanzen prüft der Ausführungscontroller in regelmäßigen Abständen, ob er weiterhin die exklusive Sperre für die primäre Datei mit mehreren Instanzen hält. Wenn die Sperre aufgrund eines Problems mit dem NFS -Server verloren geht, wird die primäre Instanz in vielen Fällen fast sofort (vor dieser Prüfung) übernommen. Es werden zusätzliche regelmäßige Dateisperrenprüfungen durchgeführt, um sicherzustellen, dass der primäre Warteschlangenmanager bei einem ungewöhnlichen Dateisystemproblem einen Failover durchführt.

Standardmäßig werden diese Dateisperrenprüfungen alle 20 Sekunden durchgeführt. Bei Bedarf können Sie diesen Wert ändern, indem Sie den Optimierungsparameter **FileLockHeartBeatLen** festlegen. Der Standardwert für den Optimierungsparameter ist 10 Sekunden (der Warteschlangenmanager lässt zwei Zyklen der Prüfung zu, bevor eine Aktion ausgeführt wird, die zum Standardverhalten der Prüfung alle 20 Sekunden führt). Der Mindestwert des Optimierungsparameters beträgt 10 Sekunden, der Maximalwert 600 Sekunden (10 Minuten).

Anmerkung: Wenn die **ECHearHeartBeatLen** erhöht wurde, wirkt sich dies auf die Ablaufsteuerung der **FileLockHeartBeatLen** -Prüfungen aus. Dateisperrenprüfungen werden durchgeführt, wenn die allgemeinen Statusprüfungen durchgeführt werden, sodass eine Verringerung der Häufigkeit der allgemeinen Statusprüfungen (**ECHearHeartBeatLen**) dazu führen kann, dass Dateisperrenprüfungen bis zu 30 Sekunden nach dem konfigurierten **FileLockHeartBeatLen** durchgeführt werden.

Überprüft den Zustand der Benutzeranwendung

Der Warteschlangenmanager überprüft regelmäßig, ob alle lokal gebundenen Anwendungen, die nicht mehr aktiv sind, einen MQDISC-MQI-Aufruf ausgeführt haben, bevor er beendet wird. Diese Prüfungen werden gleichzeitig mit den in „Allgemeine Statusprüfungen“ auf Seite 6 beschriebenen allgemeinen Statusprüfungen durchgeführt. Das Standardintervall für solche Prüfungen beträgt daher 10000 Millisekunden (10 Sekunden) und die Änderung des Werts des Optimierungsparameters **ECHearBeatLen** ändert die Häufigkeit, mit der sie vorgenommen werden. Diese Prüfung soll in erster Linie sicherstellen, dass alle Ressourcen, die einer verbundenen Anwendung zugeordnet sind, freigegeben werden. Sie führt nicht dazu, dass ein HA-Warteschlangenmanager oder ein Warteschlangenmanager ohne Hochverfügbarkeit beendet wird oder von einer alternativen Instanz übernommen wird.

IBM MQ -Clientanwendungen, die ohne Ausgabe eines MQDISC-MQI-Aufrufs beendet wurden, werden vom Agentenprozess separat erkannt und alle Ressourcen, die der Verbindung zugeordnet sind, werden freigegeben.

Zugehörige Konzepte

[Hochverfügbarkeitskonfigurationen](#)

Ereignisüberwachung

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

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

- Erkennen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Unterstützung bei der Bestimmung der Ursachen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Generieren Sie einen Prüfprotokoll.
- Änderungen am Status des Warteschlangenmanagers ändern

Zugehörige Verweise

[„Ereignistypen“ auf Seite 12](#)

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

[Ereignisnachrichtenreferenz](#)

[Ereignisnachrichtenformat](#)

IBM MQ-Ereignisnachrichten veröffentlichen

Vorbereiten von IBM MQ, um Ereignisnachrichten zu veröffentlichen

Informationen zu diesem Vorgang

Ereignisnachrichten werden in speziell benannte Warteschlangen mit dem Namen SYSTEM.AD-MIN.<feature name>.EVENT geschrieben.

Wichtig anzumerken ist, dass der Name dieser Ereigniswarteschlangen wichtig ist. Standardmäßig werden auf einem Warteschlangenmanager alle Ereigniswarteschlangen als lokale Warteschlangen definiert. Sie können diese Warteschlangen jedoch löschen und neu definieren, möglicherweise als ferne Warteschlange, so dass alle Ereignisse in einen dedizierten Warteschlangenmanager zur Ereignisverarbeitung

getrichert werden. Alternativ dazu können Sie eine Aliaswarteschlange verwenden, die auf ein Topic-Objekt verweist.

In jedem Fall erfordert jede Umleitungsmethode, dass Ihre Anwendungen, die die Ereigniswarteschlangen lesen, den Namen der Warteschlange, aus der die Nachrichten gelesen werden, nicht fest codiert haben. Daher müssen Sie in der Lage sein, die Warteschlange zu konfigurieren, aus der die Anwendungen gelesen werden.

Die folgenden Befehle zeigen, wie Sie Ihre Ereigniswarteschlangen so umdefinieren können, dass die Ereignisnachrichten unter Verwendung der folgenden Voraussetzungen veröffentlicht werden. Sie haben:

- nicht mit der Verwendung von Ereignissen begonnen oder
- alle Nachrichten aus den vorhandenen Ereigniswarteschlangen entfernt und die lokalen Warteschlangen vor diesen Schritten gelöscht.

Diese Schritte zeigen nur, dass die Ereigniswarteschlangen QMGR und CHANNEL neu definiert werden. Dies kann jedoch für alle Ereignisse erweitert werden.

Anmerkung: Die Topic-Zeichenfolge ist so konzipiert, dass eine Anwendung für alle Ereignisse, die ein Platzhalterzeichen verwenden, oder für bestimmte Ereignisse abonniert werden kann, wie erforderlich.

Vorgehensweise

Setzen Sie die folgenden Befehle ab:

```
DEFINE TOPIC (ADMIN.QMGR.EVENT) TOPICSTR('Events/QMgr')
DEFINE TOPIC (ADMIN.CHANNEL.EVENT) TOPICSTR('Events/Channel')

DEFINE QALIAS (SYSTEM.ADMIN.QMGR.EVENT) TARGTYPE(TOPIC) TARGET (ADMIN.QMGR.EVENT)
DEFINE QALIAS (SYSTEM.ADMIN.CHANNEL.EVENT) TARGTYPE(TOPIC) TARGET (ADMIN.CHANNEL.EVENT)

DEFINE QLOCAL (ADMIN.EVENT)
DEFINE QLOCAL (ADMIN.QMGR.EVENT)

DEFINE SUB (EVENTS.ALL) TOPICSTR('Events+') PSPROP (NONE)
DESTCLAS (PROVIDED) DEST (ADMIN.EVENT)
DEFINE SUB (EVENTS.QMGR) TOPICSTR('Events/QMgr') PSPROP (NONE)
DESTCLAS (PROVIDED) DEST (ADMIN.QMGR.EVENT)
```

Wenn Ihre Ereignisleseanwendung in der Lage ist, Ereignisnachrichten aus einer beliebigen Warteschlange zu lesen, kann diese Anwendung so rekonfiguriert werden, dass sie aus einer der oben definierten Warteschlangen liest.

Die Konfiguration PSPROP(NONE) in den DEFINE SUB-Befehlen stellt sicher, dass keine der Nachrichteneigenschaften, die von der Publish/Subscribe-Engine hinzugefügt wurden, z. B. MQTopicString, der Ereignisnachricht hinzugefügt wird. Damit wird sichergestellt, dass vorhandene Anwendungen weiterhin unverändert funktionieren.

Darüber hinaus können Anwendungen auch direkt mit dem Aufruf MQSUB subscribieren, um die Informationen zu empfangen, anstatt den Verwaltungsbefehl DEFINE SUB zu verwenden.

Jetzt sind mehrere Anwendungen in der Lage, die Informationen zu Ereignissen zu lesen, die vom Warteschlangenmanager ausgegeben werden.

Instrumentierungsereignisse

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

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

Abbildung 1 auf Seite 11 veranschaulicht das Konzept der Instrumentierungsereignisse.

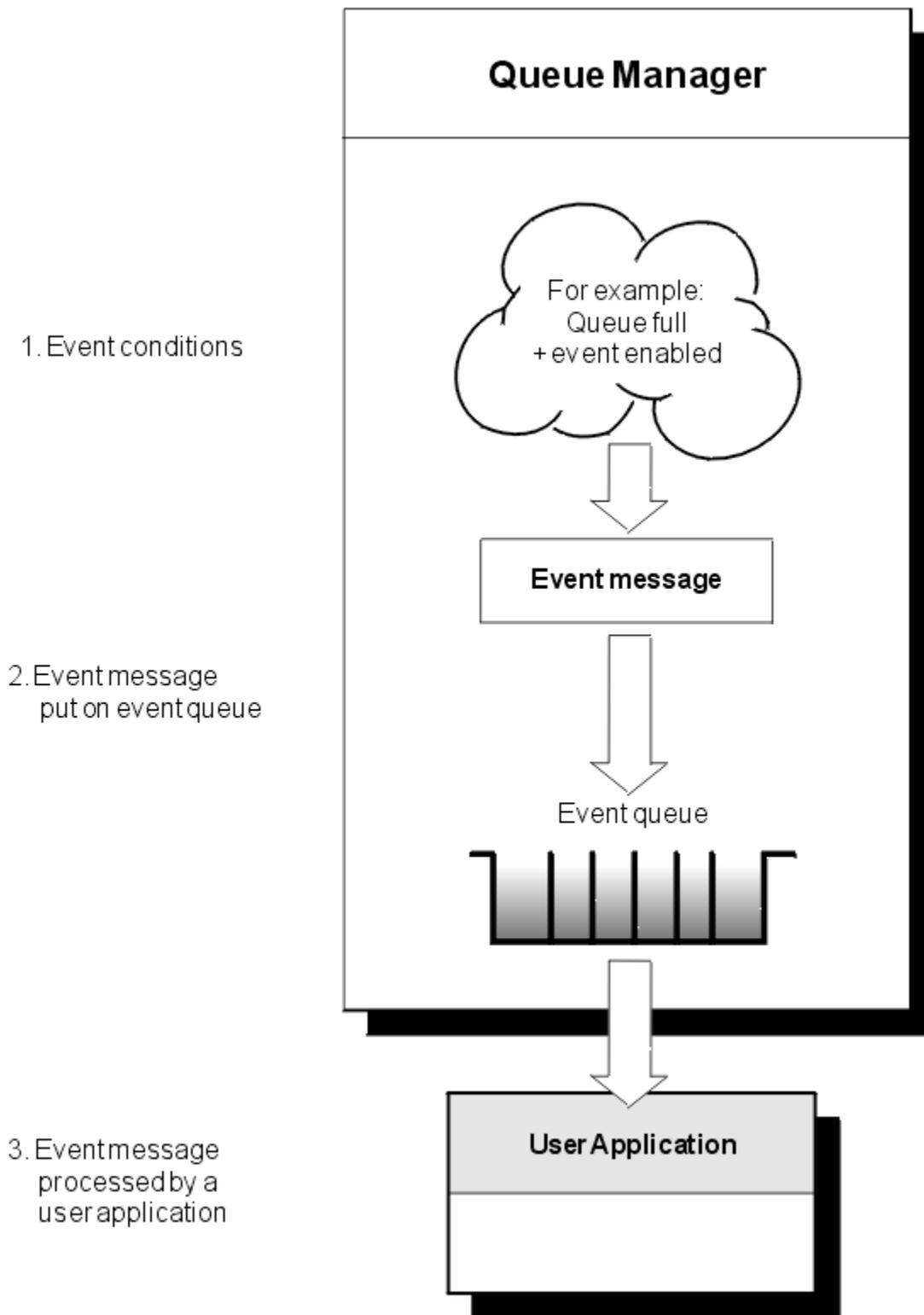


Abbildung 1. Informationen zu Instrumentierungs

Ereignisüberwachungsanwendungen

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

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

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

Ereignisbenachrichtigung über Ereigniswarteschlangen

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

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

Die zugehörigen Informationen beschreiben das Format von Ereignisnachrichten.

Bedingungen, die Ereignisse verursachen

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

- Es wird ein Schwellenwert für die Anzahl der Nachrichten in einer Warteschlange erreicht.
- Eine Kanalinstanz wird gestartet oder gestoppt.
- Ein WS-Manager wird aktiv oder wird zum Stoppen aufgefordert.
- Eine Anwendung versucht, eine Warteschlange unter Angabe einer Benutzerkennung zu öffnen, die auf den Systemen IBM MQ for IBM i, AIX, Linux®, and Windows nicht zugelassen 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 25

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

„Beispielprogramm zur Überwachung von Instrumentierungsereignissen auf Multiplatforms“ auf Seite 59

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

Ereignistypen

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

IBM MQ-Instrumentierungsereignisse weisen die folgenden Typen auf:

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

- Lokale Ereignisse

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

Diese Ereigniswarteschlange:

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

Enthält Nachrichten von:

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

Durch die Integration von Instrumentierungsereignissen in Ihre eigene Systemmanagementanwendung können Sie die Aktivitäten in vielen Warteschlangenmanagern, über viele verschiedene Knoten hinweg und für mehrere IBM MQ-Anwendungen überwachen. Insbesondere können Sie alle Knoten in Ihrem System von einem einzigen Knoten aus überwachen (für die Knoten, die IBM MQ-Ereignisse unterstützten); weitere Informationen finden Sie unter Abbildung 2 auf Seite 13.

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

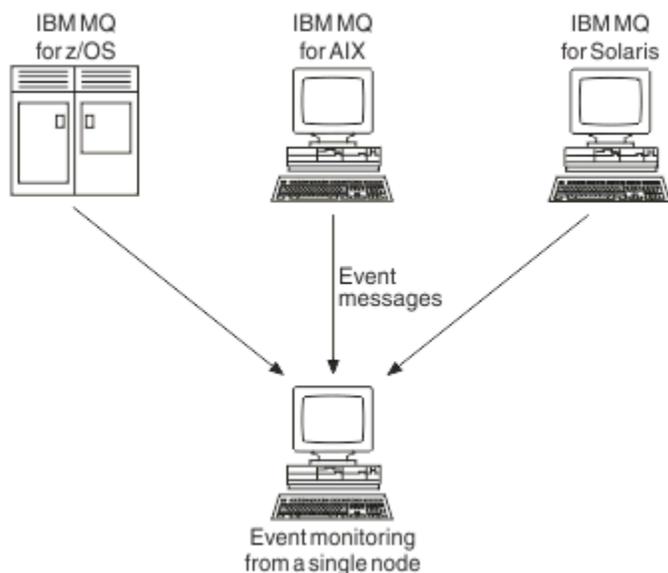


Abbildung 2. Warteschlangenmanager auf verschiedenen Plattformen überwachen, auf einem einzelnen Knoten

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

WS-Manager-Ereignisse

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

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

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

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

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

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

ALW Berechtigung (nur auf AIX, Linux, and Windows)

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 Autoritätsereignisse sind nur auf AIX, Linux, and Windows gültig.

Inhibit

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

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

Lokal

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

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

Fern

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

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

Start und Stopp

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

 z/OS unterstützt nur Startereignisse.

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

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

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

Kanal- und Brückenereignisse

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

Kanalereignisse werden unter den folgenden Umständen generiert:

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

Anmerkung: Clientverbindungen verursachen keinen Kanal Gestartet oder Gestoppte Ereignisse.

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

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

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

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

Die Kanalereignisnachrichten können die folgenden Ereignisdaten enthalten:

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

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

IMS-Bridge-Ereignisse (nur z/OS)



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

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

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

SSL-Ereignisse

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

Die SSL-Ereignisnachrichten können die folgenden Ereignisdaten enthalten:

- [Kanal-SSL-Fehler](#)
- [Kanal-SSL-Warnung](#)

Durchsatzereignisse

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

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

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

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

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

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

Es gibt zwei Typen von Leistungsereignis:

Ereignisse der Warteschlangenlänge

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

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

Ereignis 'Intervall für Warteschlangenwartung'

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

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

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

Konfigurationsergebnisse

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

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

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

Konfigurationsergebnisse funktionieren auf folgende Weise:

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

MQRC_CONFIG_CHANGE_OBJECT, MQRC_CONFIG_CREATE_OBJECT oder MQRC_CONFIG_DELETE_OBJECT treten für die jeweiligen MQSC- oder PCF-Befehle auf, die Sie bei einem Objekt ausgeben können.

MQRC_CONFIG_REFRESH_OBJECT tritt für diese Ereignisnachrichten auf, die beim Erstellen des Grundlinienbilds geschrieben werden.

Es gibt vier Typen von Konfigurationsergebnisereignis:

Objektergebnisse erstellen

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

Objektergebnisse ändern

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

Objektereignisse löschen

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

Objektereignisse aktualisieren

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

Befehlereignisse

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

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

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

Befehlereignisse enthalten die folgenden Ereignisdaten: Befehl .

Multi *Ereignisse der Protokollfunktion*

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

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

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

Die Ereignisnachricht der Protokollfunktion enthält die folgenden Ereignisdaten: Logger .

Zusammenfassung der Ereignisnachrichtendaten

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

Ereignistyp	Sehen Sie sich diese Themen
Berechtigungsereignisse	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-Brückenereignisse	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 MQ-Scriptbefehle (MQSC).
- Die entsprechenden IBM MQ-PCF-Befehle
-  Die Operationen und Steuerkonsolen für Warteschlangenmanager unter z/OS.
- IBM MQ Explorer.

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

Zugehörige Konzepte

„Instrumentierungsereignisse“ auf Seite 10

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.

[Operationen und Steuerkonsolen unter z/OS verwenden](#)

Zugehörige Tasks

[Verwaltungstasks automatisieren](#)

[Programmierbare Befehlsformate verwenden](#)

Zugehörige Verweise

„Ereignistypen“ auf Seite 12

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 Warteschlangenmanagerattribut an. In [Tabelle 1](#) auf [Seite 21](#) 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 Brückenereignisse steuern

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

Verwenden Sie zum Aktivieren oder Inaktivieren von Kanalereignissen den MQSC-Befehl **ALTER QMGR** und geben Sie das entsprechende Warteschlangenmanagerattribut an. In [Tabelle 2](#) auf [Seite 21](#) ist zusammengefasst, wie Sie Kanal- und Brückenereignisse aktivieren. Wenn Sie ein WS-Manager-Ereignis inaktivieren möchten, setzen Sie den entsprechenden Parameter auf DISABLED .

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

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 Ausnahmebedingung gesetzt wird, werden die folgenden Rückkehrcodes und die entsprechenden Ursachenqualifikationsmerkmale generiert:

- MQRC_CHANNEL_ACTIVATED
- MQRC_CHANNEL_CONV_ERROR
- MQRC_CHANNEL_NOT_ACTIVATED
- MQRC_CHANNEL_STOPPED
 - mit den folgenden 'ReasonQualifiers':
 - MQRQ_CHANNEL_STOPPED_ERROR
 - MQRQ_CHANNEL_STOPPED_RETRY
 - MQRQ_CHANNEL_STOPPED_DISABLED

- MQRC_CHANNEL_STOPPED_BY_USER
- MQRC_CHANNEL_BLOCKED
 - mit den folgenden 'ReasonQualifiers':
 - MQRQ_CHANNEL_BLOCKED_NOACCESS
 - MQRQ_CHANNEL_BLOCKED_USERID
 - MQRQ_CHANNEL_BLOCKED_ADDRESS

Leistungsereignisse steuern

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

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

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

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

Ereignisse der Warteschlangenlänge

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

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

Ereignis 'Intervall für Warteschlangenwartung'

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

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

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

Steuerung von Konfigurations-, Befehls- und Protokollfunktionseignissen

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

Konfigurationsereignisse

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

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

Befehlsereignisse

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

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

Ereignisse der Protokollfunktion

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

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

Ereigniswarteschlangen

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

Sie können Ereigniswarteschlangen wie folgt definieren:

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

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

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

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

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

Wenn eine Ereigniswarteschlange nicht verfügbar ist

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

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

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

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

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

Ausgelöste Ereigniswarteschlangen verwenden

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

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

Zugehörige Konzepte

„Ereignisse steuern“ auf Seite 20

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

„Format von Ereignisnachrichten“ auf Seite 24

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

Anwendungsprogrammierung mit gemeinsam genutzten Warteschlangen

Bedingungen für ein Auslöserereignis

Zugehörige Verweise

QSGDisp (MQLONG)

Format von Ereignisnachrichten

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

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

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

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

Zugehörige Verweise

[„MQMD des Aktivitätsberichts \(Nachrichtendeskriptor\)“ auf Seite 107](#)

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 111](#)

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

[„Aktivitätenbericht MQCFH \(PCF-Header\)“ auf Seite 112](#)

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

[Ereignisnachrichtenreferenz](#)

[Ereignisnachrichtenformat](#)

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

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

[Ereignisnachrichtenbeschreibungen](#)

Durchsatzereignisse

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

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

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

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

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

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

Zugehörige Verweise

[„Ereignistypen“ auf Seite 12](#)

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 26](#) sind die Ereignisstatistikdaten zusammengefasst, die Sie zum Analysieren des Verhaltens einer Warteschlange verwenden können. Alle Statistikdaten beziehen sich auf das, was seit dem letzten Zurücksetzen der Statistik passiert ist.

<i>Tabelle 3. Performance-Ereignisstatistik</i>	
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 eingereihten Nachrichten (die Anzahl der MQPUT-Aufrufe an die Warteschlange), da die Statistikdaten zum letzten Mal zurückgesetzt wurden.
MsgDeqCount	Die Anzahl der Nachrichten, die in die Warteschlange gestellt wurden (die Anzahl der MQGET-Aufrufe an die Warteschlange), da die Statistik zuletzt zurückgesetzt wurde.

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

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

Zugehörige Konzepte

„[Durchsatzereignisse](#)“ auf [Seite 25](#)

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 28](#)

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 28](#)

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

Zugehörige Tasks

„[Warteschlangenserviceintervallereignisse aktivieren](#)“ auf [Seite 29](#)

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

Zugehörige Verweise

[Hohe Warteschlangenlänge](#)

[Warteschlangenstatistik zurücksetzen](#)

[RESET QSTATS](#)

Ereignis 'Intervall für Warteschlangenwartung'

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

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

Ereignisse des Warteschlangenserviceintervalls werden in gemeinsam genutzten Warteschlangen nicht unterstützt.

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

Warteschlangenserviceintervall OK

Gibt an, dass nach einer der folgenden Operationen:

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

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

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

Hohes Warteschlangenserviceintervall

Gibt an, dass nach einer der folgenden Operationen:

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

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

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

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

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

Abbildung 3 auf Seite 27 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.

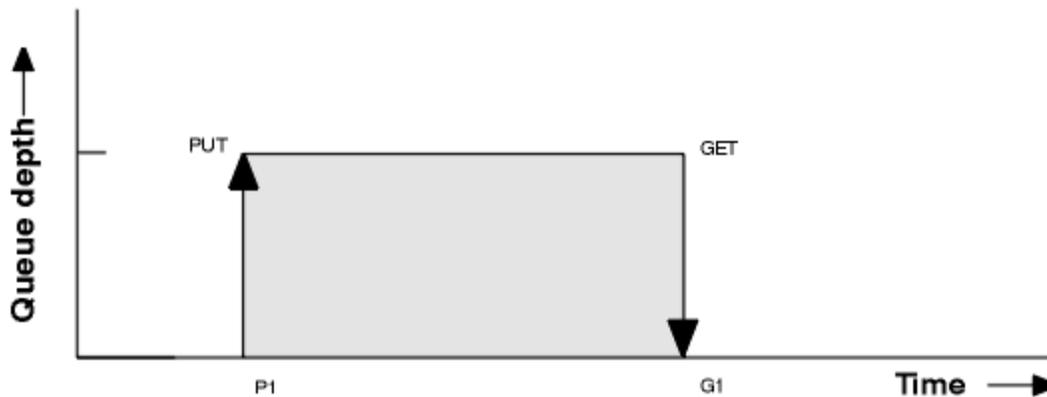


Abbildung 3. Informationen zum Warteschlangenserviceintervall

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 28 beschrieben.

Zugehörige Verweise

[Warteschlangenserviceintervall OK](#)

[Hohes Warteschlangenserviceintervall](#)

[QServiceIntervalEvent \(MQLONG\)](#)

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

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 29

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

Warteschlangenserviceintervallereignisse aktivieren

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

Informationen zu diesem Vorgang

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

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

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

Tabelle 4. Warteschlangenserviceintervallereignisse unter Verwendung von MQSC aktivieren (Forts.)	
Ereignis für Warteschlangenserviceinterv	Warteschlangenattribute
Serviceintervall	QSVCIINT (<i>tt</i>). Hierbei steht <i>tt</i> für den Serviceintervallzeit in Millisekunden.

Führen Sie die folgenden Schritte aus, um Warteschlangenserviceintervallereignisse zu aktivieren

Vorgehensweise

1. Setzen Sie das WS-Managerattribut **PERFMEV** auf ENABLED .
Leistungsereignisse werden auf dem Warteschlangenmanager aktiviert.
2. Setzen Sie das Steuerattribut **QSVCIIEV** für ein Warteschlangenserviceintervall-Hoch-oder OK-Ereignis in der Warteschlange nach Bedarf.
3. Legen Sie das Attribut **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 die Beispiele in diesem Abschnitt, um die Informationen zu verstehen, die Sie aus Warteschlangenserviceintervall-Ereignissen abrufen können.

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

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

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

Die Beispiele veranschaulichen die folgenden Aspekte von Warteschlangenserviceintervall-Ereignissen:

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

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

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

Zugehörige Konzepte

„Ereignis 'Intervall für Warteschlangenwartung'“ auf Seite 26

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 28

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.

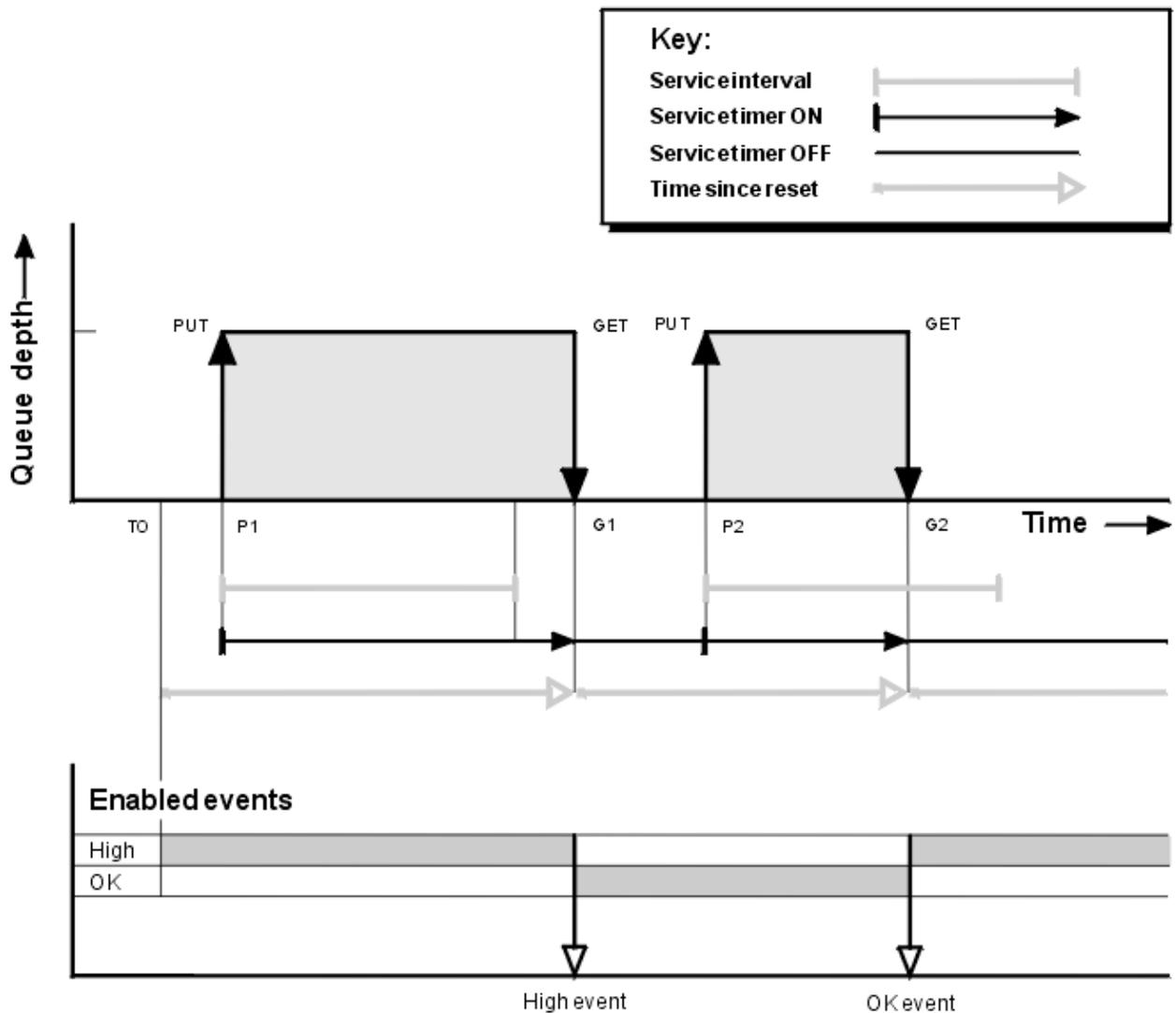


Abbildung 4. Ereignisse des Warteschlangenserviceintervalls-Beispiel 1

Commentary

- 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.
- 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
 - Das OK-Ereignis wird automatisch aktiviert.
 - Das Ereignis "Hoch" ist inaktiviert.
 Da die Warteschlange jetzt leer ist, wird der Servicezeitgeber in den Status OFF (Aus) umgeschaltet.
- Bei P2 wird eine zweite Nachricht in die Warteschlange gestellt. Damit wird der Servicezeitgeber erneut gestartet.
- Bei G2 wird die Nachricht aus der Warteschlange entfernt. Da jedoch die abgelaufene Zeit zwischen P2 und G2 kleiner als das Serviceintervall ist, wird im MQGET-Aufruf von G2 ein Ereignis 'Warteschlangen-

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

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

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

Zusammenfassung der Ereignisstatistik

In [Tabelle 5 auf Seite 33](#) sind die Ereignisstatistikdaten für dieses Beispiel zusammengefasst.

<i>Tabelle 5. Ereignisstatistikzusammenfassung für Beispiel 1</i>		
Attribut	Ereignis 1	Ereignis 2
Zeitpunkt des Ereignisses	T (G1)	T (G2)
Typ des Ereignisses	Hoch	OK
Zurücksetzungs-zeit (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 32](#) 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.

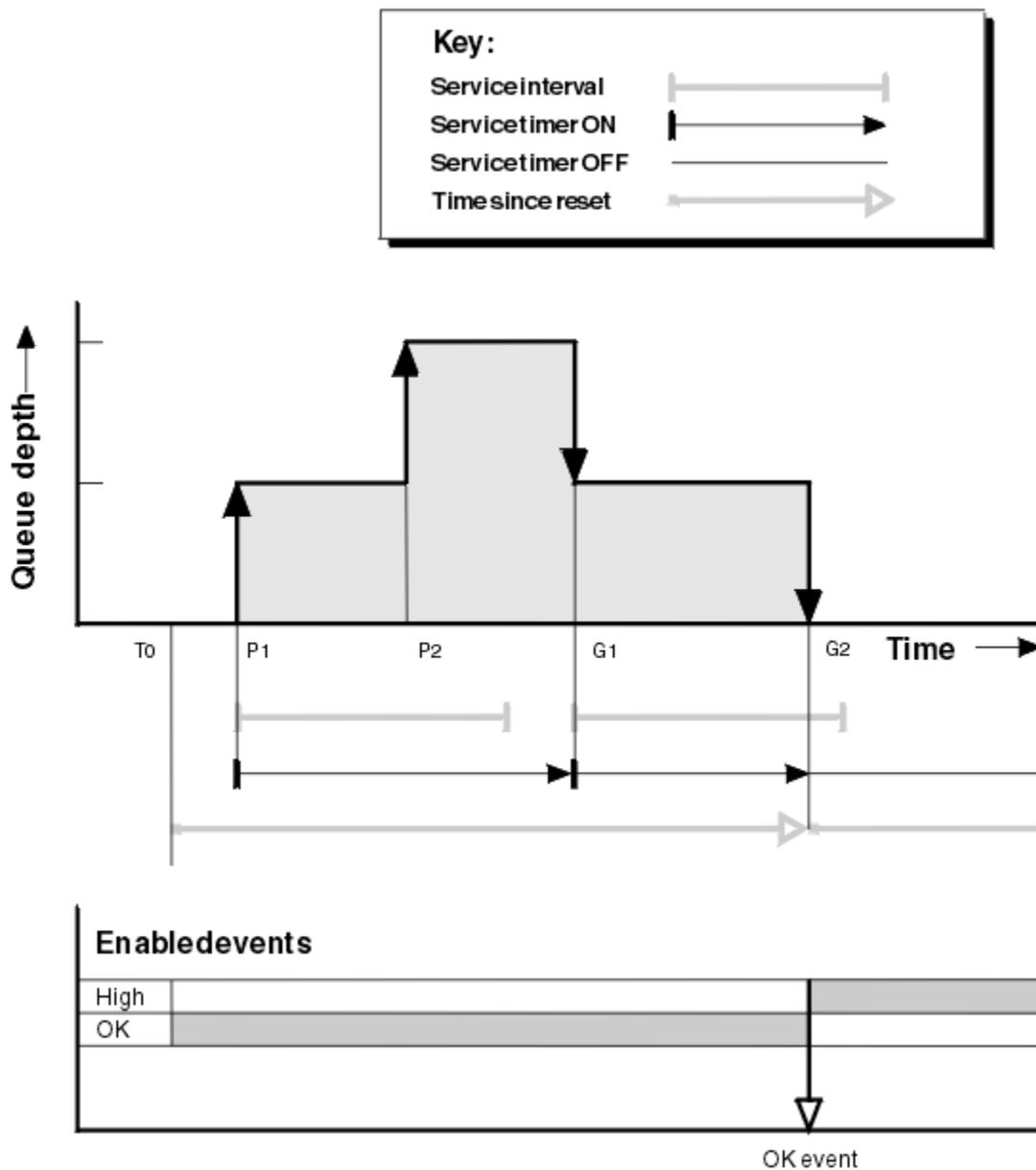


Abbildung 5. Ereignisse des Warteschlangenserviceintervalls-Beispiel 2

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 35](#) sind die Ereignisstatistikdaten für dieses Beispiel zusammengefasst.

Attribut	Ereignis 2
Zeitpunkt des Ereignisses	T (G2)
Typ des Ereignisses	OK
Zurücksetzungs-zeit (TimeS)	T (G2)-T (0)
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.

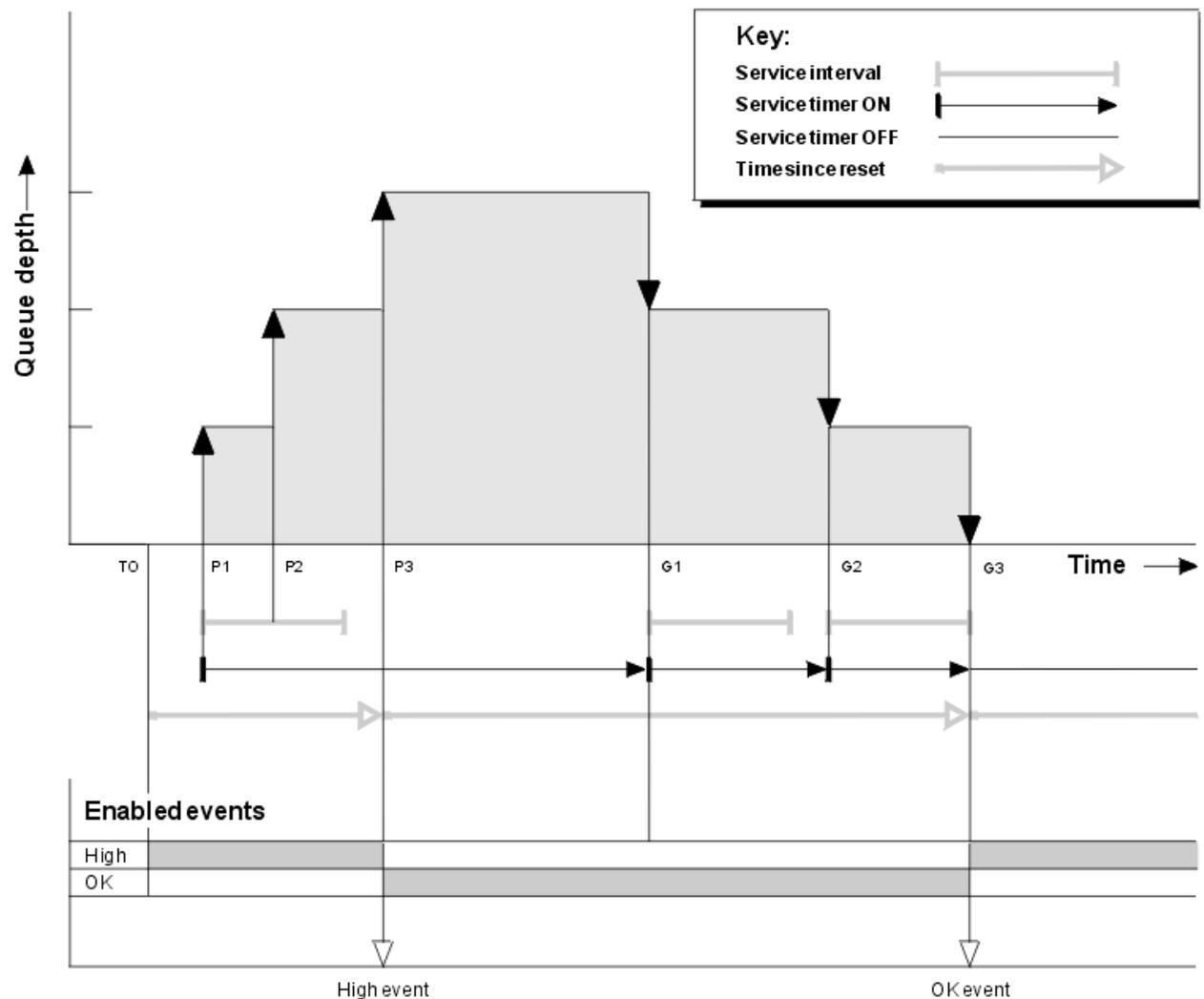


Abbildung 6. Ereignisse des Warteschlangenserviceintervalls-Beispiel 3

Commentary

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

Zusammenfassung der Ereignisstatistik

In Tabelle 7 auf Seite 36 sind die Ereignisstatistikdaten für dieses Beispiel zusammengefasst.

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

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

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

Die Vorabwarnung, dass Probleme auf dem Weg sind, macht es einfacher, vorbeugende Maßnahmen zu ergreifen. Zu diesem Zweck stellt IBM MQ die folgenden Warteschlangenlängereignisse 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 40

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

Zugehörige Verweise

[Warteschlange voll](#)

[Hohe Warteschlangenlänge](#)

[Niedrige Warteschlangenlänge](#)

Warteschlangenlänge aktivieren

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

Informationen zu diesem Vorgang

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

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

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

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

Vorgehensweise

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

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

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

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

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

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

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

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

Ergebnisse

Anmerkung:

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

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

Beispiel

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

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

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

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

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

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

Shared queues and queue depth events on z/OS

On IBM MQ for z/OS, event monitoring is more straightforward for an application that uses shared queues if all the queue managers in the queue sharing group have the same setting for the **PERFMEV** attribute.

When a queue depth event occurs on a shared queue, and the queue manager attribute **PERFMEV** is set to **ENABLED**, the queue managers in the queue sharing group produce an event message. If **PERFMEV** is set to **DISABLED** on some of the queue managers, event messages are not produced by those queue managers, making event monitoring from an application more difficult. For more straightforward monitoring, give each queue manager the same setting for the **PERFMEV** attribute.

This event message that each queue manager generates represents its individual usage of the shared queue. If a queue manager performs no activity on the shared queue, various values in the event message are null or zero. You can use null event messages as follows:

- Ensure that each active queue manager in a queue sharing group generates one event message
- Highlight cases of no activity on a shared queue for the queue manager that produced the event message

Coordinating queue manager

When a queue manager issues a queue depth event, it updates the shared queue object definition to toggle the active performance event attributes. For example, depending on the definition of the queue attributes, a Queue Depth High event enables a Queue Depth Low and a Queue Full event. After updating the shared queue object successfully, the queue manager that detected the performance event initially becomes the *coordinating queue manager*.

If enabled for performance events, the coordinating queue manager performs the following actions:

1. Issues an event message that captures all shared queue performance data it has gathered since the last time an event message was created, or since the queue statistics were last reset. The message descriptor (MQMD) of this message contains a unique correlation identifier (*CorrelId*) created by the coordinating queue manager.
2. Broadcasts to all other *active* queue managers in the same queue sharing group to request the production of an event message for the shared queue. The broadcast contains the correlation identifier created by the coordinating queue manager for the set of event messages.

Having received a request from the coordinating queue manager, if there is an active queue manager in the queue sharing group that is enabled for performance events, that active queue manager issues an

event message for the shared queue. The event message that is issued contains information about all the operations performed by the receiving (active) queue manager since the last time an event message was created, or since the statistics were last reset. The message descriptor (MQMD) of this event message contains the unique correlation identifier (*CorrelId*) specified by the coordinating queue manager.

When performance events occur on a shared queue, n event messages are produced, where n is a number from 1 to the number of active queue managers in the queue sharing group. Each event message contains data that relates to the shared queue activity for the queue manager that generated the event message.

Differences between shared and nonshared queues

Enabling queue depth events on shared queues differs from enabling them on nonshared queues. A key difference is that events are enabled for shared queues even if **PERFMEV** is DISABLED on the queue manager. This is not the case for nonshared queues.

Consider the following example, which illustrates this difference:

- QM1 is a queue manager with *PerformanceEvent* (**PERFMEV** in MQSC) set to DISABLED.
- SQ1 is a shared queue with **QSGDISP** set to (SHARED) QLOCAL in MQSC.
- LQ1 is a nonshared queue with **QSGDISP** set to (QMGR) QLOCAL in MQSC.

Both queues have the following attributes set on their definitions:

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

If messages are placed on both queues so that the depth meets or exceeds the **QDEPTHHI** threshold, the **QDPHIEV** value on SQ1 switches to DISABLED. Also, **QDPLOEV** and **QDPMAXEV** are switched to ENABLED. SQ1's attributes are automatically switched for each performance event at the time the event criteria are met.

In contrast the attributes for LQ1 remain unchanged until **PERFMEV** on the queue manager is ENABLED. This means that if, for example, the queue manager's **PERFMEV** attribute is ENABLED, DISABLED and then set to ENABLED again, the performance event settings on shared queues might not be consistent with those of nonshared queues, even though they might have initially been the same.

Beispiele für Ereignisse der Warteschlangenlänge

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

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

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

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

```
ALTER QMGR PERFMEV(ENABLED)

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

Zugehörige Konzepte

„Ereignisse der Warteschlangenlänge“ auf Seite 36

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

Zugehörige Tasks

„Warteschlangenlänge aktivieren“ auf Seite 37

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

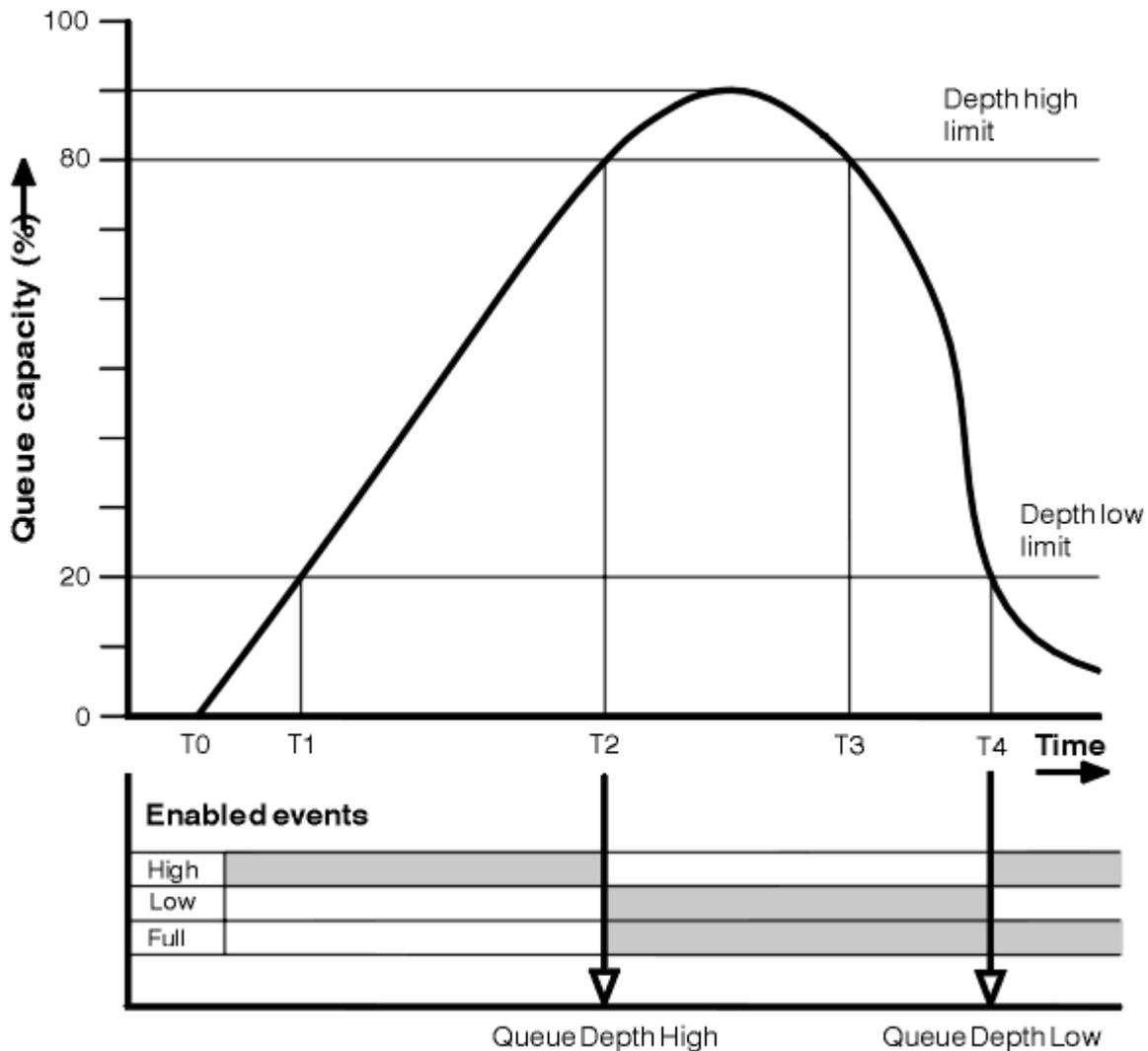


Abbildung 7. Ereignisse der Warteschlangenlänge (1)

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 42](#) sind die Warteschlangenergebnisstatistik zusammengefasst, und [Tabelle 9 auf Seite 42](#) fasst zusammen, welche Ereignisse aktiviert sind.

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

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 43](#) zeigt die Variation der Warteschlangenlänge im Verlauf der Zeit an.

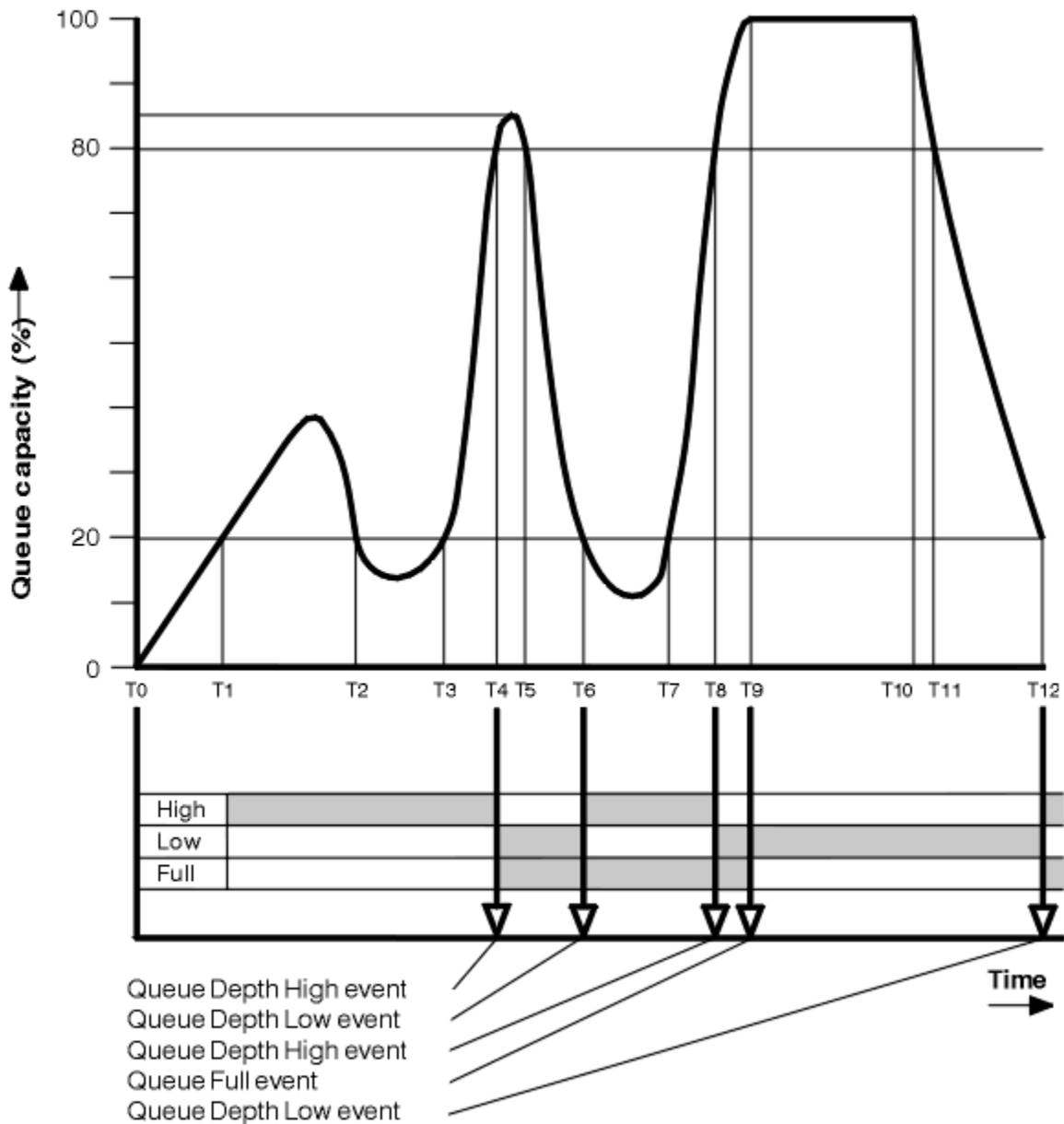


Abbildung 8. Ereignisse der Warteschlangenlänge (2)

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 44 fasst die Warteschlangenereignisstatistik zusammen und Tabelle 11 auf Seite 44 fasst zusammen, welche Ereignisse für dieses Beispiel zu unterschiedlichen Zeiten aktiviert sind.

Tabelle 10. Ereignisstatistikzusammenfassung für Warteschlangentiefenereignisse (Beispiel 2)

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

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

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

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

Konfigurationsereignisse

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

Informationen zum Aktivieren von Ereignissen finden Sie unter „[Steuerung von Konfigurations-, Befehls- und Protokollfunktionsereignissen](#)“ auf Seite 22.

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

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

Die Ereignisdaten enthalten die folgenden Informationen:

Ursprungsinformationen

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

Kontextinformationen

Ein Replikat der Kontextinformationen in den Nachrichtendaten aus der Befehlsnachricht.

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

Objektidentität

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

Objektattribute

enthält die Werte aller Attribute im Objekt.

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

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

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

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

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

Zugehörige Konzepte

[„Konfigurationsereignisse“ auf Seite 17](#)

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 12](#)

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 44

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](#)

[MQSET - Objektattribute festlegen](#)

Konfigurationsereignisverwendung

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

Sie können Konfigurationsereignisse für die folgenden Zwecke verwenden:

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

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

Für einen MQSET-Aufruf oder einen der folgenden Befehle:

- DEFINE, Objekt
- ALTER, Objekt
- Objekt DELETE

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

Auswirkungen von CMDSCOPE

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

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

Zugehörige Konzepte

[Einführung in Programmierbare Befehlsformate](#)

[„Konfigurationsereignisse“ auf Seite 44](#)

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

Zugehörige Verweise

[MQSET - Objektattribute festlegen](#)

[MQSET - Objektattribute festlegen](#)

Ereigniskonfigurationsereignis aktualisieren

Das Konfigurationsereignis für die 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 44

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

Zugehörige Verweise

REFRESH QMGR

WS-Manager aktualisieren

Befehlsereignisse

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

Die Ereignisdaten enthalten die folgenden Informationen:

Ursprungsinformationen

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

Kontextinformationen

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

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

Befehlsinformationen

Der Typ des Befehls, der ausgegeben wurde.

Befehlsdaten

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

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

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

Zugehörige Verweise

Befehl

„Ereignistypen“ auf Seite 12

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

Befehlsereignisgenerierung

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

In den folgenden Situationen wird eine Befehlsereignisnachricht generiert:

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

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

Wenn Befehlsereignisse nicht generiert werden

Eine Befehlsereignisnachricht wird unter den folgenden Umständen nicht generiert:

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

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

Zugehörige Konzepte

„Befehlsereignisse“ auf Seite 48

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

Zugehörige Verweise

[REFRESH QMGR](#)

[START QMGR](#)

[SUSPEND QMGR](#)

[RESUME QMGR](#)

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

Befehlsereignisverwendung

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

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

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

Auswirkungen von CMDSCOPE

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

Zugehörige Konzepte

„Befehlsereignisse“ auf Seite 48

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

„Befehlsereignisgenerierung“ auf Seite 49

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

Zugehörige Verweise

[Die MQSC-Befehle](#)

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

Ereignisse der Protokollfunktion

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

Die Ereignisdaten enthalten die folgenden Informationen:

- Der Name des aktuellen Protokollspeicherbereichs.

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

Ein Protokollfunktionsereignis wird für die Verwaltung des Archivprotokolls generiert, d. h., wenn sich **ARCHLOG** ändert und der Wert **ARCHLOG** in die Ereignisnachricht der Protokollfunktion eingeschlossen wird.

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

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

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

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

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

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

Zugehörige Verweise

[Logger](#)

„Ereignistypen“ auf Seite 12

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

Generierung von Protokollfunktionsereignis

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

In den folgenden Situationen wird eine Protokollfunktionsereignisnachricht generiert:

- Wenn das Warteschlangenmanagerattribut LOGGEREV als ENABLED angegeben ist und der Warteschlangenmanager mit dem Schreiben in einen neuen Protokollspeicherbereich bzw. unter IBM i in einen 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.
- Wenn das WS-Manager-Attribut LOGGEREV benachrichtigt wird, dass der Name des ältesten Protokollspeicherbereichs, für den der Warteschlangenmanager auf die Archivierung wartet (ARCHLOG), archiviert wurde, wird er archiviert.

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

Wenn Protokollfunktionsereignisse nicht generiert werden

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

- Wenn ein WS-Manager für die Verwendung der Umlaufprotokollierung konfiguriert ist.

In diesem Fall wird das WS-Managerattribut LOGGEREV auf DISABLED gesetzt und kann nicht geändert werden.

- Wenn ein Warteschlangenmanager einen Fehler feststellt, der versucht, ein Protokollfunktionsereignis in die Ereigniswarteschlange zu stellen, wird die Aktion, die das Ereignis ausgelöst hat, abgeschlossen, es wird jedoch keine Ereignisnachricht generiert.

Zugehörige Konzepte

„Ereignisse der Protokollfunktion“ auf Seite 50

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

Zugehörige Verweise

[LoggerEvent \(MQLONG\)](#)

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

[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 Protokolldateien entfernen. Durch die regelmäßige Entfernung überflüssiger Protokoll extents wird die Plattenspeicherplatzbelegung auf ein Minimum gehalten.

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

Zugehörige Konzepte

„Ereignisse der Protokollfunktion“ auf Seite 50

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

Zugehörige Verweise

[LoggerEvent \(MQLONG\)](#)

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

„Generierung von Protokollfunktionsereignis“ auf Seite 51

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

Beispielprogramm C zur Überwachung der Ereigniswarteschlange der Protokollfunktion

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

```

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

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

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

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

#define MAX_MESSAGE_LENGTH 8000

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

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

#if defined(MQ_64_BIT)

```

```

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

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

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

static PMQCHAR ParmToString(MQLONG Parameter);

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

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

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

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

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

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

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

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

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

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

```

```

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

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

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

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

return 0;
}

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

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

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

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

        MQGET( hConn,
              hEventQueue,
              &Mqmd,

```

```

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

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

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

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

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

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

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

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

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

pPCFCmd = (PMQCHAR) (pCfh+1);
printf("-----\n");
while(pCfh->ParameterCount-->0)
{
    pCfst = (MQCFST *) pPCFCmd;
    switch(pCfst->Type)
    {
        case MQCFT_STRING:
            Parm = ParmToString(pCfst->Parameter);
            if (Parm != NULL)
            {
                printf("%-32s",Parm);
            }
            else
            {
                printf("%-32" Int32 "d",pCfst->Parameter);
            }
        }
    }
}

```

```

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

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

```

Beispielausgabe

Diese Anwendung erzeugt die folgende Form der Ausgabe:

```

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

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

```

Zugehörige Konzepte

„[Logger-Ereignisverwendung](#)“ auf Seite 52

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

„[Befehlsereignisverwendung](#)“ auf Seite 50

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

Zugehörige Verweise

„[Generierung von Protokollfunktionsereignis](#)“ auf Seite 51

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

Berechtigungskonfigurationsereignisse

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

Die Ereignisdaten enthalten die folgenden Informationen:

Ursprungsinformationen

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

Kontextinformationen

Ein Replikat der Kontextinformationen in den Nachrichtendaten aus der Befehlsnachricht.

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

ID des Berechtigungsdatensatzes

enthält den Profilnamen und den Objekttyp des Berechtigungsdatensatzes.

Objektattribute

enthält die Werte aller Attribute im Berechtigungssatz.

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

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

Zugehörige Verweise

„Ereignistypen“ auf Seite 12

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

Ereignisgenerierung der Berechtigungskonfiguration

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

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

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

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

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

Wenn Berechtigungskonfigurationsereignisse nicht generiert werden

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

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

Zugehörige Konzepte

„Befehlsereignisse“ auf Seite 48

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

Zugehörige Verweise

[REFRESH QMGR](#)

Multi Beispielprogramm zur Überwachung von Instrumentierungsereignissen auf Multiplatforms

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

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

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

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

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

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

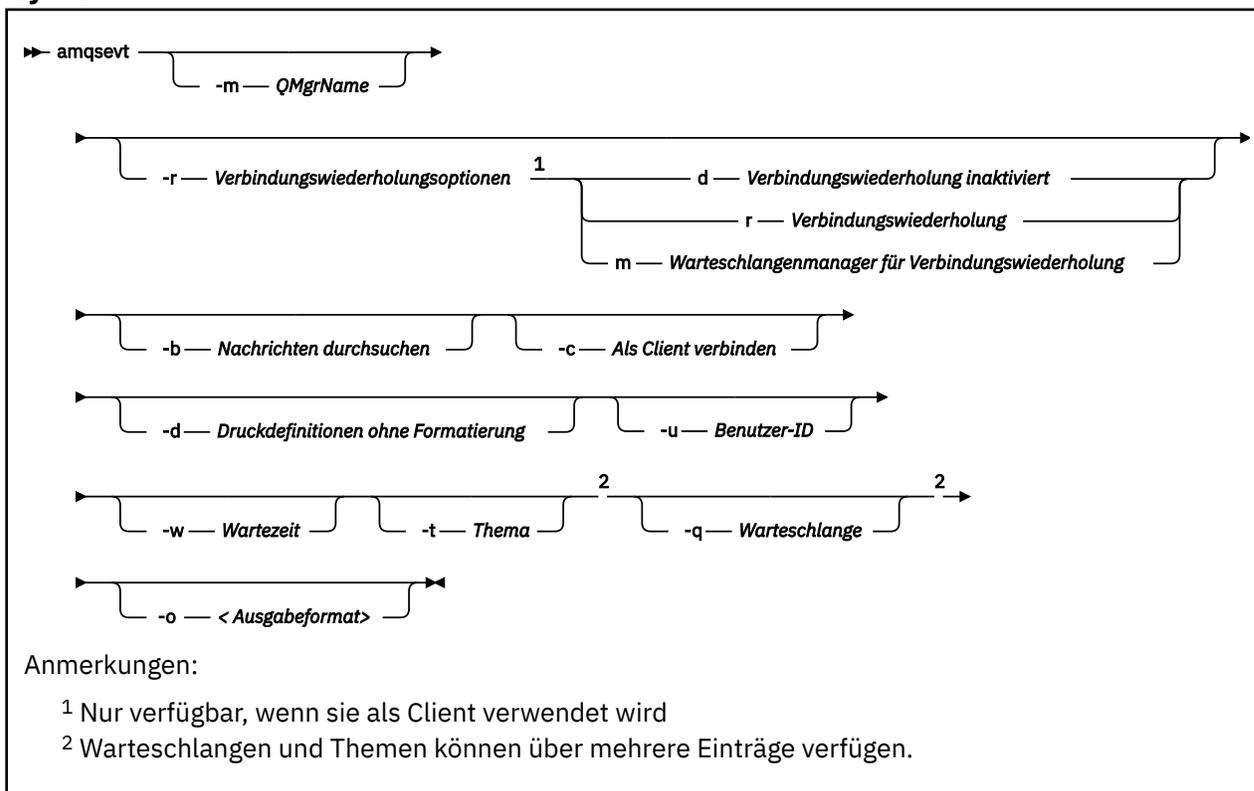


Achtung: Sie können das Programm ohne Angabe von Parametern verwenden. In diesem Fall versucht das Programm, eine Verbindung zum Standardwarteschlangenmanager herzustellen und Nachrichten aus der Standardgruppe von Ereigniswarteschlangen zu lesen (SYSTEM.ADMIN.*.EVENT).

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

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

Syntax



Optionale Parameter

-m *QueueManagerName*

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

-r *Reconnection Options*

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

d

Verbindung zum Client wiederherstellen

r

Verbindung zum Client wiederherstellen

m

Verbindung zum WS-Manager wiederherstellen

-b

Nur Datensätze durchsuchen, statt die Nachrichten destruktiv zu lesen

-c

Wählt die Verbindung als Client aus.

-d

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

-u *User ID*

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

-w *Wait*

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

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

-t *Topic* und

-q *Queue*

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

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

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

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

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

-o <output format>

Das Format der Ausgabe. Mögliche Werte:

Text

Standardtextformat; dies ist der Standardwert.

json

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

.

Beispielausgabe

Die folgenden drei Beispiele zeigen die Ausgabe des Programms.

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

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

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

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

Das dritte Beispiel zeigt die JSON-Ausgabe des Befehls.

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

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

Verwendungsbeispiel

Das folgende Beispiel zeigt, wie mehrere Warteschlangen verwendet werden:

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

Zugehörige Konzepte

„Ereignisüberwachung“ auf Seite 9

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

„Instrumentierungsereignisse“ auf Seite 10

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

Zugehörige Verweise

C-Programmierung

„Beispielprogramm C zur Überwachung der Ereigniswarteschlange der Protokollfunktion“ auf Seite 53

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

Nachrichtenüberwachung

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

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

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

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

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

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

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

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

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

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

Informationen aus Aktivitäten

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

Die letzte bekannte Position einer Nachricht

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

Konfigurationsprobleme mit einem WS-Manager-Netz

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

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

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

Nachrichtenrouten

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

Für eine Trace-Route-Nachricht aufgezeichnete Aktivitätsinformationen verwenden

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

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

Für die ursprüngliche Nachricht aufgezeichnete Aktivitätsinformationen verwenden

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

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

Zugehörige Konzepte

„Nachrichtenüberwachung“ auf Seite 62

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 64

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 71

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

Zugehörige Tasks

Schreiben eigener Nachrichtenkanalagenten

Nachrichtenroutetechniken

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

Aktivitätenaufzeichnung

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

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

Trace-Route-Messaging

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

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

Vergleich von Aktivitätsaufzeichnung und Trace-Route-Messaging

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

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

Vollständigkeit der Nachrichtenroute

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

MCAs und benutzerdefinierte Anwendungen, die mit Warteschlangenmanagern verbunden sind, können Informationen zu den Aktivitäten aufzeichnen, die im Namen einer Nachricht ausgeführt werden. Die Aufzeichnung von Aktivitätsinformationen wird von den WS-Managerattributen ACTIVREC und ROUTEREC gesteuert. Ein Warteschlangenmanager-Netz kann vollständige Nachrichtenrouten ermitteln.

Wie Aktivitätsinformationen gespeichert werden

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

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

Zugehörige Konzepte

„Aktivitätenaufzeichnung“ auf Seite 66

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 71

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 77

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

„Aktivitätsbericht-Nachrichtendaten“ auf Seite 114

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

Aktivitätenaufzeichnung

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

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

Verwendung des Aktivitätsberichts

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

Die letzte bekannte Position einer Nachricht ermitteln

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

Konfigurationsprobleme mit einem WS-Manager-Netz bestimmen

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

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

Aktivitätsberichtsformat

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

Der Nachrichtendeskriptor

- Eine MQMD-Struktur

Nachrichtendaten

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

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

Zugehörige Verweise

[MQMD - Nachrichtendeskriptor](#)

[MQEPH - Eingebetteter PCF-Header](#)

Aktivitätsaufzeichnung steuern

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

Informationen zu diesem Vorgang

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

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

Vorgehensweise

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

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

2. Aktivieren oder Inaktivieren des Warteschlangenmanagers für die Aktivitätsaufzeichnung.

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

MSG

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

WARTESCHLANGE

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

INAKTIVIERT

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

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

```
ALTER QMGR ACTIVREC(QUEUE)
```

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

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

Einrichten einer allgemeinen Warteschlange für Aktivitätsberichte

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

Vorbereitende Schritte

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

Informationen zu diesem Vorgang

Wenn eine Reihe von Warteschlangenmanagern in einem Warteschlangenmanagernetzwerk so eingestellt sind, dass sie Aktivitätsberichte an die lokale Systemwarteschlange zustellen, kann es zeitaufwendig sein, die Positionen der Aktivitätsberichte zu einer bestimmten Nachricht zu ermitteln. Alternativ können Sie auch einen einzigen Knoten verwenden, der ein Warteschlangenmanager ist, der eine gemeinsame Warteschlange hostet. Alle Warteschlangenmanager in einem WS-Manager-Netz können Aktivitätsberichte an diese allgemeine Warteschlange zustellen. Der Vorteil der Verwendung einer allgemeinen Warteschlange besteht darin, dass WS-Manager keine Aktivitätsberichte an die in einer Nachricht angegebene 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 anfordern, sofern sie nicht für eine Trace-Route-Nachricht generiert wurden. In diesem Fall können Sie Aktivitätsberichte mit der IBM MQ-Anwendung zur Routenanzeige anfordern.

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

Vorgehensweise

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

OperationType

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

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

OperationDate und OperationTime

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

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

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

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

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

Weitere Aktivitätsberichte abrufen

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

Informationen zu diesem Vorgang

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

Vorgehensweise

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

Ein Aktivitätsbericht kann nur dann an eine Warteschlange für nicht zustellbare Nachrichten zugestellt werden, wenn die Berichtsoption MQRO_DEAD_LETTER_Q festgelegt ist.

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

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

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

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

Umstände, in denen Aktivitätsinformationen nicht erfasst werden

Um die vollständige Abfolge der Aktivitäten zu ermitteln, die im Namen einer Nachricht ausgeführt wurden, müssen Informationen zu jeder Aktivität angefordert werden. Wenn die Informationen zu einer Aktivität nicht aufgezeichnet wurden oder nicht erfasst wurden, können Sie nur eine Teilsequenz von Aktivitäten bestimmen.

Aktivitätsinformationen werden unter den folgenden Umständen nicht erfasst:

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

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

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

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

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

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

Trace-Route-Messaging

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

Da die Trace-Route-Nachricht über das WS-Manager-Netz weitergeleitet wird, werden die Aktivitätsinformationen aufgezeichnet. Diese Aktivitätsinformationen enthalten Informationen zu den Anwendungen, die die Aktivitäten ausgeführt haben, wann sie ausgeführt wurden, und die Operationen, die als Teil der

Aktivitäten ausgeführt wurden. Sie können die Informationen, die im Trace-Route-Messaging aufgezeichnet wurden, für die folgenden Zwecke verwenden:

So ermitteln Sie die letzte bekannte Position einer Nachricht:

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

Gehen Sie wie folgt vor, um Konfigurationsprobleme mit einem Warteschlangenmanagernetz

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

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

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

Zugehörige Konzepte

„Trace-Route-Nachrichtenreferenz“ auf Seite 131

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

Wie Aktivitätsinformationen aufgezeichnet werden

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

Akkumulierende Aktivitätsinformationen in den Nachrichtendaten der Trace-Route-Nachricht

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

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

Aktivitätsinformationen mithilfe von Aktivitätsberichten aufzeichnen

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

enthält. Die Aktivitätsaufzeichnung für Trace-Route-Nachrichten funktioniert genauso wie für jede andere Nachricht.

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

Informationen zu aufgezeichneten Aktivitäten werden abgefragt

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

Vorbereitende Schritte

Wenn Sie mit den Aktivitätsinformationen nicht vertraut sind, lesen Sie die Informationen in [„Wie Aktivitätsinformationen aufgezeichnet werden“](#) auf Seite 72.

Informationen zu diesem Vorgang

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

Prozedur

- Rufen Sie die Trace-Route-Nachricht ab.

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

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

- Verwenden Sie eine Trace-Route-Antwortnachricht.

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

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

- Verwenden Sie Aktivitätsberichte.

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

Trace für Trace-Route-Messaging steuern

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

Vorbereitende Schritte

Wenn Sie Aktivitätsberichte zum Aufzeichnen von Aktivitätsinformationen für eine Trace-Route-Nachricht verwenden, lesen Sie die Informationen in [„Aktivitätsaufzeichnung steuern“](#) auf Seite 67.

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 76.
- 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 76

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

Zugehörige Tasks

[„Aktivitätsaufzeichnung steuern“](#) auf Seite 67

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

WS-Manager für Trace-Route-Messaging aktivieren

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

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

MSG

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

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

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

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

WARTESCHLANGE

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

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

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

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

INAKTIVIERT

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

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

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

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

Anwendungen für Trace-Route-Messaging aktivieren

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

Vorbereitende Schritte

Wenn Sie mit dem Format einer Trace-Route-Nachricht nicht vertraut sind, lesen Sie die Informationen in „[Trace-Route-Nachrichtenreferenz](#)“ auf Seite 131.

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 Trace-Route-Nachricht ausgeführt werden, den Wert des Parameters *MaxActivities* überschreitet.
Die Gesamtzahl der Aktivitäten ist die Summe aus *RecordedActivities*, *UnrecordedActivities* und *DiscontinuityCount*.
Wenn die Gesamtzahl der Aktivitäten *MaxActivities* überschreitet, weisen Sie die Nachricht mit dem Feedback MQFB_MAX_ACTIVITIES zurück.
 4. Wenn der Wert von *Accumulate* als MQROUTE_ACCUMULATE_IN_MSG oder MQROUTE_ACCUMULATE_AND_REPLY festgelegt ist und der Warteschlangenmanager für Trace-Route-Messaging aktiviert ist, schreiben Sie eine Aktivität-PCF-Gruppe an das Ende des PCF-Blocks in den Nachrichtendaten einer Trace-Route-Nachricht.
 5. Liefern Sie die Trace-Route-Nachricht an eine lokale Warteschlange.
 - Wenn der Parameter, *Deliver*, als MQROUTE_DELIVER_NO angegeben wird, wird die Trace-Route-Nachricht mit dem Feedback MQFB_NOT_DELIVERED zurückgewiesen.
 - Wenn der Parameter *Deliver* als MQROUTE_DELIVER_YES angegeben ist, übergeben Sie die Trace-Route-Nachricht an die lokale Warteschlange.
 6. Eine Trace-Route-Antwortnachricht generieren, wenn alle folgenden Bedingungen erfüllt sind:
 - Die Trace-Route-Nachricht wurde an eine lokale Warteschlange gesendet oder zurückgewiesen.
 - Der Wert des Parameters (*Accumulate*) lautet MQROUTE_ACCUMULATE_AND_REPLY.
 - Der Warteschlangenmanager ist für Trace-Route-Messaging aktiviert.
 Die Trace-Route-Antwortnachricht wird in die Warteschlange gestellt, die durch das Attribut ROUTE-REC des Warteschlangenmanagers festgelegt wird.
 7. Wenn die Trace-Route-Nachricht einen Aktivitätsbericht angefordert hat und der WS-Manager für die Aktivitätsaufzeichnung aktiviert ist, generieren Sie einen Aktivitätsbericht.
Der Aktivitätsbericht wird in die Warteschlange gestellt, die durch das Warteschlangenmanagerattribut ACTIVREC bestimmt wird.

Trace-Route-Nachricht generieren und konfigurieren

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

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

Nachrichtendeskriptor

Eine MQMD-Struktur, wobei das Feld *Format* auf MQFMT_ADMIN oder MQFMT_EMBEDDED_PCF gesetzt ist.

Nachrichtendaten

Eine der folgenden Kombinationen:

- Ein PCF-Header (MQCFH) und Trace-Route-Nachrichtendaten, wenn *Format* auf MQFMT_ADMIN gesetzt ist
- Ein eingebetteter PCF-Header (MQEPH), Trace-Route-Nachrichtendaten und zusätzliche benutzerdefinierte Nachrichtendaten, wenn *Format* auf MQFMT_EMBEDDED_PCF gesetzt ist.

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

Manuelle Generierung

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

Die IBM MQ-Anwendung für die Routenanzeige

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

Einschränkung: dspmqrte kann nicht auf IBM MQ for 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 mit dem optionalen Parameter *-ceine* Verbindung zum Warteschlangenmanager als Client 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.

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

Nachrichtendaten

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

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

PCF-Gruppe "TraceRoute"

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

In der folgenden Tabelle sind die Parameter in der Gruppe *TraceRoute* aufgelistet, die von einem MCA erkannt werden. Weitere Parameter können hinzugefügt werden, falls benutzerdefinierte Anwendungen für ihre Erkennung geschrieben werden (siehe hierzu „Zusätzliche Aktivitätsinformationen“ auf Seite 83).

Tabelle 12. PCF-Gruppe "TraceRoute"

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

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

Detail

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

MQROUTE_DETAIL_LOW

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

MQROUTE_DETAIL_MEDIUM

Die in *MQROUTE_DETAIL_LOW* angegebenen Aktivitäten sollten aufgezeichnet werden. Darüber hinaus werden Aktivitäten, die von MCAs ausgeführt werden, aufgezeichnet.

MQROUTE_DETAIL_HIGH

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

RecordedActivities

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

UnrecordedActivities

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

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

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

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

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

DiscontinuityCount

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

MaxActivities

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

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

Der Wert von *MaxActivities* kann wie folgt sein:

Eine positive Ganzzahl

Die maximale Anzahl der Aktivitäten.

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

MQROUTE_UNLIMITED_ACTIVITIES

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

Accumulate

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

MQROUTE_ACCUMULATE_IN_MSG

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

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

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

MQROUTE_ACCUMULATE_AND_REPLY

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

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

MQRROUTE_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:

MQRROUTE_FORWARD_IF_SUPPORTED

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

MQRROUTE_FORWARD_ALL

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

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

1. Stellen Sie fest, ob der ferne Warteschlangenmanager in der Lage ist, Trace-Route-Messaging zu unterstützen.
 - Wenn der ferne Warteschlangenmanager in der Lage ist, Trace-Route-Messaging zu unterstützen, fährt der Algorithmus mit Schritt „4“ auf Seite 80 fort.
 - Wenn der ferne Warteschlangenmanager nicht in der Lage ist, Tracing-Route-Messaging zu unterstützen, fährt der Algorithmus mit Schritt „2“ auf Seite 80 fort.
2. Stellen Sie fest, ob der Parameter *Deliver* aus der Gruppe *TraceRoute* alle nicht erkannten Zustelloptionen in der Bitmaske MQRROUTE_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 80 fort.
3. Ermitteln Sie den Wert des Parameters *Deliver* aus der PCF-Gruppe *TraceRoute* in der Trace-Route-Nachricht.
 - Wenn *Deliver* als MQRROUTE_DELIVER_YES angegeben wird, wird die Trace-Route-Nachricht an den fernen WS-Manager weitergeleitet.
 - Wenn *Deliver* als MQRROUTE_DELIVER_NO angegeben wird, wird der Algorithmus mit Schritt „4“ auf Seite 80 fortgesetzt.
4. Stellen Sie fest, ob der Parameter *Forward* aus der Gruppe *TraceRoute* alle nicht erkannten Weiterleitungsoptionen in der Bitmaske MQRROUTE_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 80 fort.
5. Ermitteln Sie den Wert des Parameters *Forward* aus der PCF-Gruppe *TraceRoute* in der Trace-Route-Nachricht.
 - Wenn *Forward* als MQRROUTE_FORWARD_IF_SUPPORTED angegeben wird, wird die Trace-Route-Nachricht mit dem Feedback MQFB_NOT_FORWARDED zurückgewiesen.
 - Wenn *Forward* als MQRROUTE_FORWARD_ALL angegeben wird, kann die Trace-Route-Nachricht an den fernen WS-Manager weitergeleitet werden.

Deliver

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

MQROUTE_DELIVER_YES

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

MQROUTE_DELIVER_NO

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

Eine allgemeine Warteschlange für Antwortnachrichten für Tracetrouten konfigurieren

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

Vorbereitende Schritte

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

Informationen zu diesem Vorgang

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

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

Vorgehensweise

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

Erfassung und Verwendung von aufgezeichneten Informationen

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

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

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

Informationen aus Trace-Route-Antwortnachrichten werden abgefragt.

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

Informationen zu diesem Vorgang

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

Vorgehensweise

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

Informationen aus Trace-Route-Nachrichten werden abgefragt.

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

Informationen zu diesem Vorgang

Sie können Aktivitätsinformationen aus einer Trace-Route-Nachricht nur dann anfordern, wenn Sie die Position der Trace-Route-Nachricht kennen und den Parameter *Accumulate* in der *TraceRoute* -PCF-Gruppe entweder als *MQROUTE_ACCUMULATE_IN_MSG* oder als *MQROUTE_ACCUMULATE_AND_REPLY* angegeben hat.

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

Vorgehensweise

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

Informationen aus Aktivitätsberichten werden abgefragt

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

Informationen zu diesem Vorgang

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

Vorgehensweise

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

Zusätzliche Aktivitätsinformationen

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

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

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

GroupName

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

ParameterName

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

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

ParameterValue

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

Zusätzliche Aktivitätsinformationen aufzeichnen: Beispiel 1

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

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

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

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

ColorValue

Kennung
65536

Datentyp
MQCFST

Wert
'Rot'

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

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

```
65536: 'Red'
```

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

Ein Beispiel für die Situation, in der die IBM MQ-Anwendung für die Routenanzeige die Parameter-ID des PCF-Parameters nicht erkennt, finden Sie im Abschnitt „Zusätzliche Aktivitätsinformationen aufzeichnen: Beispiel 2“ auf Seite 85.

Zusätzliche Aktivitätsinformationen aufzeichnen: Beispiel 2

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

- Mit der IBM MQ-Anwendung für die Routenanzeige wird eine Trace-Route-Nachricht generiert und auf die gleiche Weise in ein Warteschlangenmanagernetz gestellt, wie in „Zusätzliche Aktivitätsinformationen aufzeichnen: Beispiel 1“ auf Seite 84 beschrieben wird.
- Da die Trace-Route-Nachricht über das WS-Manager-Netz weitergeleitet wird, führt eine Benutzeranwendung, die für die Trace-Route-Nachrichtenübertragung aktiviert ist, eine niedrige Detailaktivität im Namen der Nachricht durch. Zusätzlich zum Schreiben der Standardaktivitätsinformationen in die Trace-Route-Nachricht schreibt die Benutzeranwendung die folgenden PCF-Parameter an das Ende der Aktivitätsgruppe:

ColorInfo

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

ColorName

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

ColorValue

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

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

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

```
Color: 'Red'
```

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

IBM MQ-Anwendung für die Routenanzeige

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

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

- Zum Konfigurieren, Generieren und Einlegen einer Trace-Route-Nachricht in ein WS-Manager-Netz.
Wenn Sie eine Trace-Route-Nachricht in ein WS-Manager-Netz einreihen, können Aktivitätsinformationen erfasst und verwendet werden, um die Route zu bestimmen, die die Trace-Route-Nachricht genommen hat. Sie können die Merkmale der Trace-Route-Nachrichten wie folgt angeben:
 - Die Zieladresse der Trace-Route-Nachricht.
 - Gibt an, wie die Trace-Route-Nachricht eine andere Nachricht imitiert.
 - Wie die Trace-Route-Nachricht behandelt werden soll, wenn sie über ein WS-Manager-Netz weitergeleitet wird.
 - Gibt an, ob die Aktivitäten-Aufzeichnung oder Trace-Route-Messaging zum Aufzeichnen von Aktivitätsinformationen verwendet wird.
- Gehen Sie wie folgt vor, um Aktivitätsinformationen für eine Trace-Route-Nachricht zu sortieren und anzuzeigen.

Wenn die IBM MQ-Anwendung für die Routenanzeige eine Trace-Route-Nachricht in ein Warteschlangenmanagernetz gestellt hat, nachdem die zugehörigen Aktivitätsinformationen zurückgegeben wurden, können die Informationen sofort sortiert und angezeigt werden. Alternativ können mit der IBM

MQ-Anwendung zur Routenanzeige Aktivitätsinformationen zu einer Trace-Route-Nachricht, die zuvor generiert wurde, sortiert und angezeigt werden.

Zugehörige Verweise

[dspmqrte](#)

Parameter für Trace-Route-Nachrichten

Verwenden Sie diese Seite, um eine Übersicht über die Parameter zu erhalten, die von der IBM MQ-Anwendung für die Routenanzeige **dspmqrte** bereitgestellt werden, um die Merkmale einer Trace-Route-Nachricht zu ermitteln. Dazu gehört auch die Verarbeitung bei der Weiterleitung durch ein Warteschlangenmanagernetz.

Zugehörige Verweise

[dspmqrte](#)

Warteschlangenmanagerverbindung

Geben Sie auf dieser Seite den Warteschlangenmanager an, zu dem die IBM MQ-Anwendung für die Routenanzeige eine Verbindung herstellt.

-c

Gibt an, dass die IBM MQ-Anwendung für die Routenanzeige eine Verbindung als Clientanwendung herstellt.

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

-m QMgrName

Der Name des Warteschlangenmanagers, zu dem die IBM MQ-Anwendung für die Routenanzeige eine Verbindung herstellt. Der Name kann bis zu 48 Zeichen enthalten.

Wenn Sie diesen Parameter nicht angeben, wird der Standardwarteschlangenmanager verwendet.

Zieladresse

Verwenden Sie diese Seite, um die Zieladresse einer Trace-Route-Nachricht anzugeben.

-q TargetQName

Wenn die IBM MQ-Anwendung zur Routenanzeige verwendet wird, um eine Trace-Route-Nachricht in ein Warteschlangenmanagernetz zu senden, gibt *TargetQName* den Namen der Zielwarteschlange an.

-ts TargetTopicString

Gibt die Themenzeichenfolge an.

-qm TargetQMgr

Qualifiziert die Ziel-Destination; die normale Auflösung des Warteschlangenmanagers wird dann angewendet. Die Ziel-Destination wird mit *-q TargetQName* oder *-ts TargetTopicString* angegeben.

Wenn Sie diesen Parameter nicht angeben, wird der Warteschlangenmanager, mit dem die IBM MQ-Anwendung für die Routenanzeige verbunden ist, als Zielwarteschlangenmanager verwendet.

-o

Gibt an, dass die Ziel-Destination nicht an ein bestimmtes Ziel gebunden ist. In der Regel wird dieser Parameter verwendet, wenn die Trace-Route-Nachricht in einen Cluster gestellt werden soll. Die Ziel-Destination wird mit der Option `MQOO_BIND_NOT_FIXED` geöffnet.

Wenn Sie diesen Parameter nicht angeben, wird die Ziel-Destination an ein bestimmtes Ziel gebunden.

Das Veröffentlichungsthema

Verwenden Sie für Publish/Subscribe-Anwendungen diese Seite, um die Themenzeichenfolge einer Trace-Route-Nachricht für die IBM MQ-Anwendung für die Routenanzeige anzugeben, die veröffentlicht werden soll.

-ts TopicName

Gibt eine Themenzeichenfolge an, für die die IBM MQ-Anwendung für die Routenanzeige eine Trace-Route-Nachricht veröffentlichen soll, und stellt diese Anwendung in den Themenmodus. In diesem Modus verfolgt die Anwendung alle Nachrichten, die sich aus der Veröffentlichungsanforderung ergeben.

Mit der IBM MQ-Anwendung für die Routenanzeige können Sie die Ergebnisse aus einem Aktivitätsbericht anzeigen, der für die Veröffentlichung von Nachrichten generiert wurde.

Nachrichtenmimierung

Verwenden Sie diese Seite, um eine Trace-Route-Nachricht zu konfigurieren, um eine Nachricht zu imitieren, z. B., wenn die ursprüngliche Nachricht nicht das geplante Ziel erreicht hat.

Eine Verwendung des Trace-Route-Messaging ist die Bestimmung der letzten bekannten Position einer Nachricht, die nicht an das geplante Ziel gelangt ist. In der IBM MQ-Anwendung für die Routenanzeige werden Parameter bereitgestellt, mit denen eine Trace-Route-Nachricht konfiguriert werden kann, die die ursprüngliche Nachricht imitiert. Wenn Sie eine Nachricht imitieren, können Sie die folgenden Parameter verwenden:

-l Persistence

Gibt die Persistenz der generierten Trace-Route-Nachricht an. Gültige Werte für *Persistence* sind:

ja

Die generierte Trace-Route-Nachricht ist persistent. (MQPER_PERSISTENT).

nein

Die generierte Trace-Route-Nachricht ist **nicht** persistent. (MQPER_NOT_PERSISTENT).

q

Die generierte Trace-Route-Nachricht übernimmt ihren Persistenzwert von der Destination, die durch *-q TargetQName* oder *-ts TargetTopicString* angegeben wird. (MQPER_PERSISTENCE_AS_Q_DEF).

Eine Trace-Route-Antwortnachricht oder die zurückgegebenen Berichtsnachrichten verwenden denselben Persistenzwert wie die ursprüngliche Trace-Route-Nachricht.

Wenn *Persistence* als **yes** angegeben ist, müssen Sie den Parameter *-rq ReplyToQ* angeben. Die Empfangswarteschlange für Antworten darf nicht in eine temporäre dynamische Warteschlange aufgelöst werden.

Wenn Sie diesen Parameter nicht angeben, wird die generierte Trace-Route-Nachricht **nicht** permanent angezeigt.

-p Priority

Gibt die Priorität der Trace-Route-Nachricht an. Der Wert von *Priority* ist größer-gleich 0, oder MQPRI_PRIORITY_AS_Q_DEF. MQPRI_PRIORITY_AS_Q_DEF gibt an, dass der Prioritätswert von dem Ziel übernommen wird, das durch *-q TargetQName* oder *-ts TargetTopicString* angegeben wird.

Wenn Sie diesen Parameter nicht angeben, wird der Prioritätswert von dem Ziel übernommen, das durch *-q TargetQName* oder *-ts TargetTopicString* angegeben wird.

-xs Expiry

Gibt die Ablaufzeit für die Trace-Route-Nachricht in Sekunden an.

Wenn Sie diesen Parameter nicht angeben, wird die Ablaufzeit als 60 Sekunden angegeben.

-ro none | ReportOption

none

Gibt an, dass keine Berichtsoptionen festgelegt sind.

ReportOption

Gibt Berichtsoptionen für die Trace-Route-Nachricht an. Mehrere Berichtsoptionen können mit einem Komma als Trennzeichen angegeben werden. Gültige Werte für *ReportOption* sind:

Aktivität

Die Berichtsoption MQRO_ACTIVITY wird festgelegt.

- Die Berichtsoption MQRO_COA_WITH_FULL_DATA ist festgelegt.

Kabeljau

Die Berichtsoption MQRO_COD_WITH_FULL_DATA ist festgelegt.

ausnahmebedingung

Die Berichtsoption 'MQRO_EXCEPTION_WITH_FULL_DATA' wird festgelegt.

Ablauf

Die Berichtsoption 'MQRO_EXPIRATION_WITH_FULL_DATA' wird festgelegt.

Verwerfen

Die Berichtsoption MQRO_DISCARD_MSG wird festgelegt.

Wenn weder *-ro ReportOption* noch *-ro none* angegeben sind, werden die Berichtsoptionen MQRO_ACTIVITY und MQRO_DISCARD_MSG angegeben.

In der IBM MQ-Anwendung zur Routenanzeige ist es nicht zulässig, Benutzerdaten zur Trace-Route-Nachricht hinzuzufügen. Wenn Benutzerdaten zur Trace-Route-Nachricht hinzugefügt werden müssen, müssen Sie die Trace-Route-Nachricht manuell generieren.

Aufgezeichnete Aktivitätsinformationen

Verwenden Sie diese Seite, um die Methode anzugeben, mit der aufgezeichnete Aktivitätsinformationen zurückgegeben werden, die Sie dann verwenden können, um die Route zu bestimmen, die eine Trace-Route-Nachricht eingenommen hat.

Aufgezeichnete Aktivitätsinformationen können wie folgt zurückgegeben werden:

- In Aktivitätsberichten
- In einer Trace-Route-Antwortnachricht
- In der Trace-Route-Nachricht selbst (die in die Zielwarteschlange gestellt wurde)

Bei Verwendung von **dspmqrte** wird die Methode zur Rückgabe aufgezeichneter Aktivitätsinformationen mithilfe der folgenden Parameter bestimmt:

-ro activity

Gibt an, dass Aktivitätsinformationen mithilfe von Aktivitätsberichten zurückgegeben werden. Standardmäßig ist die Aktivitätsaufzeichnung aktiviert.

-ac -ar

Gibt an, dass die Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden und dass eine Trace-Route-Antwortnachricht generiert werden soll.

-ac

Gibt an, dass die Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden sollen.

Wenn Sie diesen Parameter nicht angeben, werden die Aktivitätsinformationen in der Trace-Route-Nachricht nicht kumuliert.

-ar

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

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

-ac -d ja

Gibt an, dass die Aktivitätsinformationen in der Trace-Route-Nachricht kumuliert werden und dass bei der Ankunft die Trace-Route-Nachricht in die Zielwarteschlange gestellt werden soll.

-ac

Gibt an, dass die Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden sollen.

Wenn Sie diesen Parameter nicht angeben, werden die Aktivitätsinformationen in der Trace-Route-Nachricht nicht kumuliert.

-d Ja

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

Wenn Sie diesen Parameter nicht angeben, wird die Trace-Route-Nachricht nicht in die Zielwarteschlange gestellt.

Die Trace-Route-Nachricht kann dann aus der Zielwarteschlange abgerufen werden und die erfassten Aktivitätsinformationen werden erfasst.

Sie können diese Methoden nach Bedarf kombinieren.

Darüber hinaus kann die Detailstufe der aufgezeichneten Aktivitätsinformationen mit dem folgenden Parameter angegeben werden:

-t Detail

Gibt die Aktivitäten an, die aufgezeichnet werden. Die gültigen Werte für *Detail* sind:

Niedrig

Aktivitäten, die von einer vom Benutzer definierten Anwendung ausgeführt werden, werden nur aufgezeichnet.

Medium

Die in "low" angegebenen Aktivitäten werden aufgezeichnet. Darüber hinaus werden Veröffentlichungsaktivitäten und Aktivitäten, die von MCAs ausgeführt werden, aufgezeichnet.

hoch

Aktivitäten, die in "low" und "medium" angegeben sind, werden aufgezeichnet. MCAs stellen keine weiteren Aktivitätsinformationen auf dieser Detaillierungsebene bereit. Diese Option ist für benutzerdefinierte Anwendungen verfügbar, die nur weitere Aktivitätsinformationen zugänglich machen sollen. Wenn z. B. eine benutzerdefinierte Anwendung die Route einer Nachricht durch Berücksichtigung bestimmter Nachrichtenmerkmale bestimmt, könnte die Routing-Logik in diese Detaillierungsebene eingeschlossen werden.

Wenn Sie diesen Parameter nicht angeben, werden die Aktivitäten auf mittlerer Ebene aufgezeichnet.

Die IBM MQ-Anwendung für die Routenanzeige speichert die zurückgegebenen Nachrichten standardmäßig in einer temporären dynamischen Warteschlange. Wenn die IBM MQ-Anwendung für die Routenanzeige beendet wird, wird die temporäre dynamische Warteschlange geschlossen und alle Nachrichten werden bereinigt. Wenn die zurückgegebenen Nachrichten über die aktuelle Ausführung der IBM MQ-Anwendung für die Routenanzeige hinaus erforderlich sind, muss mit den folgenden Parametern eine permanente Warteschlange angegeben werden:

-rq ReplyToQ

Gibt den Namen der Warteschlange für Antwortnachrichten an, an die alle Antworten auf die Trace-Route-Nachricht gesendet werden. Wenn die Trace-Route-Nachricht persistent ist oder wenn der Parameter *-n* angegeben ist, muss eine Empfangswarteschlange für Antworten angegeben werden, die keine temporäre dynamische Warteschlange ist.

Wenn Sie diesen Parameter nicht angeben, wird eine dynamische Empfangswarteschlange für Antworten unter Verwendung der Systemstandardmodellwarteschlange SYSTEM.DEFAULT.MODEL.QUEUE erstellt.

-rqm ReplyToQMGr

Gibt den Namen des Warteschlangenmanagers an, in dem sich die Antwortwarteschlange befindet. Der Name kann bis zu 48 Zeichen enthalten.

Wenn Sie diesen Parameter nicht angeben, wird der Warteschlangenmanager, mit dem die IBM MQ-Anwendung für die Routenanzeige verbunden ist, als Warteschlangenmanager für Antwortnachrichten verwendet.

Handhabung der Trace-Route-Nachricht

Verwenden Sie diese Seite, um zu steuern, wie eine Trace-Route-Nachricht verarbeitet wird, wenn sie über ein Warteschlangenmanagernetzwerk weitergeleitet wird.

Die folgenden Parameter können einschränken, wo die Trace-Route-Nachricht im WS-Manager-Netz weitergeleitet werden kann:

-d Deliver

Gibt an, ob die Trace-Route-Nachricht bei der Ankunft an die Zielwarteschlange zugestellt werden soll. Gültige Werte für *Deliver* sind:

ja

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

nein

Bei der Ankunft wird die Trace-Route-Nachricht nicht in die Zielwarteschlange gestellt.

Wenn Sie diesen Parameter nicht angeben, wird die Trace-Route-Nachricht nicht in die Zielwarteschlange gestellt.

-f Forward

Gibt den Typ des Warteschlangenmanagers an, an den die Trace-Route-Nachricht weitergeleitet werden kann. Ausführliche Informationen zu dem Algorithmus, den Warteschlangenmanager verwenden, um zu bestimmen, ob eine Nachricht an einen fernen Warteschlangenmanager weitergeleitet werden soll, finden Sie in „PCF-Gruppe "TraceRoute"“ auf Seite 77. Die gültigen Werte für *Forward* sind:

Alle

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

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

unterstützt

Die Trace-Route-Nachricht wird nur an einen Warteschlangenmanager weitergeleitet, der den Parameter *Deliver* von der *TraceRoute* -PCF-Gruppe berücksichtigt.

Wenn Sie diesen Parameter nicht angeben, wird die Trace-Route-Nachricht nur an einen WS-Manager weitergeleitet, der den Parameter *Deliver* berücksichtigt.

Die folgenden Parameter können verhindern, dass eine Trace-Route-Nachricht in einem Warteschlangenmanagernetzwerk unendlich bleibt:

-s Activities

Gibt die maximale Anzahl aufgezeichneter Aktivitäten an, die im Namen der Trace-Route-Nachricht ausgeführt werden können, bevor sie gelöscht wird. Dadurch wird verhindert, dass die Trace-Route-Nachricht unbegrenzt weitergeleitet wird, wenn sie in einer Endlosschleife abgefangen wird. Der Wert von *Activities* ist größer-gleich 1, oder MQROUTE_UNLIMITED_ACTIVITIES. MQROUTE_UNLIMITED_ACTIVITIES gibt an, dass eine unbegrenzte Anzahl von Aktivitäten im Namen der Trace-Route-Nachricht ausgeführt werden kann.

Wenn Sie diesen Parameter nicht angeben, kann im Namen der Trace-Route-Nachricht eine unbegrenzte Anzahl von Aktivitäten ausgeführt werden.

-xs Expiry

Gibt die Ablaufzeit für die Trace-Route-Nachricht in Sekunden an.

Wenn Sie diesen Parameter nicht angeben, wird die Ablaufzeit als 60 Sekunden angegeben.

-xp PassExpiry

Gibt an, ob die Ablaufzeit von der Trace-Route-Nachricht an eine Trace-Route-Antwortnachricht übergeben wird. Gültige Werte für *PassExpiry* sind:

ja

Die Berichtsoption 'MQRO_PASS_DISCARD_AND_EXPIRY' wird im Nachrichtendeskriptor der Trace-Route-Nachricht angegeben.

Wenn eine Trace-Route-Antwortnachricht oder Aktivitätsberichte für die Trace-Route-Nachricht generiert werden, wird die MQRO_DISCARD-Berichtsoption (falls angegeben) und die verbleibende Ablaufzeit an übergeben.

Dies ist der Standardwert.

nein

Die Berichtsoption 'MQRO_PASS_DISCARD_AND_EXPIRY' wurde nicht angegeben.

Wenn eine Trace-Route-Antwortnachricht für die Trace-Route-Nachricht generiert wird, werden die Löschoption und die Ablaufzeit aus der Trace-Route-Nachricht **nicht** weitergegeben.

Wenn Sie diesen Parameter nicht angeben, wird MQRO_PASS_DISCARD_AND_EXPIRY nicht angegeben.

-ro discard

Gibt die Berichtsoption MQRO_DISCARD_MSG an. Dadurch kann verhindert werden, dass die im WS-Manager-Netz verbleibende Trace-Route-Nachricht unbegrenzt bleibt.

Anzeige von Aktivitätsinformationen

Die IBM MQ-Anwendung für die Routenanzeige kann Aktivitätsinformationen für eine Trace-Route-Nachricht anzeigen, die sie gerade in ein Warteschlangenmanagernetz gestellt hat, oder sie kann Aktivitätsinformationen für eine zuvor generierte Trace-Route-Nachricht anzeigen. Es kann auch zusätzliche Informationen anzeigen, die von benutzerschriftlichen Anwendungen aufgezeichnet wurden.

Geben Sie den folgenden Parameter an, um anzugeben, ob Aktivitätsinformationen, die für eine Trace-Route-Nachricht zurückgegeben werden, angezeigt werden sollen:

-n

Gibt an, dass Aktivitätsinformationen, die für die Trace-Route-Nachricht zurückgegeben werden, nicht angezeigt werden sollen.

Wenn dieser Parameter von einer Anforderung für eine Trace-Route-Antwortnachricht (*-ar*) oder einen der Berichtsgenerierungsoptionen aus (*-ro ReportOption*) begleitet wird, muss eine bestimmte (nicht-Modell-) Antwortwarteschlange angegeben werden, die mit *-rq ReplyToQ* angegeben wird. Standardmäßig werden nur Aktivitätsberichtsrichten angefordert.

Nachdem die Trace-Route-Nachricht in die angegebene Zielwarteschlange gestellt wurde, wird eine 48 Zeichen lange hexadezimale Zeichenfolge angezeigt, die die Nachrichten-ID der Trace-Route-Nachricht enthält. Die Nachrichten-ID kann von der IBM MQ-Anwendung für die Routenanzeige verwendet werden, um die Aktivitätsinformationen für die Trace-Route-Nachricht mit dem Parameter *-i CorrelId* zu einem späteren Zeitpunkt anzuzeigen.

Wenn Sie diesen Parameter nicht angeben, werden die Aktivitätsinformationen, die für die Trace-Route-Nachricht zurückgegeben werden, in dem durch den Parameter *-v* angegebenen Format angezeigt.

Beim Anzeigen von Aktivitätsinformationen für eine Trace-Route-Nachricht, die gerade in ein WS-Manager-Netz gestellt wurde, kann der folgende Parameter angegeben werden:

-w WaitTime

Gibt die Dauer in Sekunden an, die die IBM MQ-Anwendung für die Routenanzeige auf Aktivitätsberichte oder auf eine Trace-Route-Antwortnachricht wartet, um diese an die angegebene Empfangswarteschlange für Antworten zurückzugeben.

Wenn Sie diesen Parameter nicht angeben, wird die Wartezeit als Verfallszeit der Trace-Route-Nachricht plus 60 Sekunden angegeben.

Bei der Anzeige der zuvor aufgelaufenen Aktivitätsinformationen müssen die folgenden Parameter festgelegt werden:

-q TargetQName

Wenn die IBM MQ-Anwendung für die Routenanzeige verwendet wird, um vorher zusammengestellte Aktivitätsinformationen anzuzeigen, gibt *TargetQName* den Namen der Warteschlange an, in der die Aktivitätsinformationen gespeichert sind.

-i CorrelId

Dieser Parameter wird verwendet, wenn mit der IBM MQ-Anwendung für die Routenanzeige nur vorher gesammelte Aktivitätsinformationen angezeigt werden. Es können viele Aktivitätsberichte und Trace-Route-Antwortnachrichten in der Warteschlange vorhanden sein, die durch *-q Zielwarteschlangennamenname* angegeben wird. *CorrelId* wird verwendet, um die Aktivitätsberichte oder eine Trace-Route-Antwortnachricht in Bezug auf eine Trace-Route-Nachricht zu identifizieren. Geben Sie die Nachrichten-ID der ursprünglichen Trace-Route-Nachricht in *CorrelId* an.

Das Format von *CorrelId* ist eine hexadezimale Zeichenfolge mit 48 Zeichen.

Die folgenden Parameter können verwendet werden, wenn zuvor aufgelaufene Aktivitätsinformationen angezeigt werden, oder wenn die aktuellen Aktivitätsinformationen für eine Trace-Route-Nachricht angezeigt werden:

-b

Gibt an, dass die IBM MQ-Anwendung für die Routenanzeige nur Aktivitätsberichte oder eine Trace-Route-Antwortnachricht in Bezug auf eine Nachricht durchsucht. Dadurch können die Aktivitätsinformationen zu einem späteren Zeitpunkt erneut angezeigt werden.

Wenn Sie diesen Parameter nicht angeben, führt die IBM MQ-Anwendung für die Routenanzeige für Aktivitätsberichte oder für eine Trace-Route-Antwortnachricht, die sich auf eine Nachricht bezieht, einen Abruf mit Löschen aus.

-v summary | all | none | outline DisplayOption

Zusammenfassung

Die Warteschlangen, an die die Trace-Route-Nachricht weitergeleitet wurde, werden angezeigt.

Alle

Alle verfügbaren Informationen werden angezeigt.

none

Es werden keine Informationen angezeigt.

Gliederung DisplayOption

Gibt die Anzeigeeoptionen für die Trace-Route-Nachricht an. Mehrere Anzeigeeoptionen können mit einem Komma als Trennzeichen angegeben werden.

Wenn keine Werte angegeben werden, wird Folgendes angezeigt:

- Der Anwendungsname
- Der Typ jeder Operation.
- Alle betriebspezifischen Parameter

Gültige Werte für *DisplayOption* sind:

Aktivität

Alle Nicht-PCF-Gruppenparameter in *Activity* -PCF-Gruppen werden angezeigt.

Kennungen

Es werden Werte mit den Parameter-IDs MQBACF_MSG_ID oder MQBACF_CORREL_ID angezeigt. Dieser Wert überschreibt *msgdelta*.

das Kundenstamms

Alle Nicht-PCF-Gruppenparameter in *Message* -PCF-Gruppen werden angezeigt. Wenn dieser Wert angegeben wird, können Sie *msgdelta* nicht angeben.

msgdelta

Alle Nicht-PCF-Gruppenparameter in *Message* -PCF-Gruppen, die seit der letzten Operation geändert wurden, werden angezeigt. Wenn dieser Wert angegeben wird, können Sie *message* nicht angeben.

Betrieb

Alle Nicht-PCF-Gruppenparameter in *Operation* -PCF-Gruppen werden angezeigt.

Traceroute

Alle Nicht-PCF-Gruppenparameter in *TraceRoute* -PCF-Gruppen werden angezeigt.

Wenn Sie diesen Parameter nicht angeben, wird eine Zusammenfassung der Nachrichtenroute angezeigt.

Anzeige zusätzlicher Informationen

Da eine Trace-Route-Nachricht über ein WS-Manager-Netz weitergeleitet wird, können benutzerdefinierte Anwendungen zusätzliche Informationen aufzeichnen, indem sie einen oder mehrere zusätzliche PCF-Parameter in die Nachrichtendaten der Trace-Route-Nachricht oder in die Nachrichtendaten eines Aktivitätsberichts schreiben. Damit die IBM MQ-Anwendung für die Routenanzeige zusätzliche Informationen in einem lesbaren Format anzeigen kann, muss sie in einem bestimmten Format aufgezeichnet werden, wie im Abschnitt „Zusätzliche Aktivitätsinformationen“ auf Seite 83 beschrieben wird.

Beispiele für die IBM MQ-Anwendung für die Routenanzeige

In den folgenden Beispielen wird gezeigt, wie Sie die IBM MQ-Anwendung für die Routenanzeige verwenden können. In jedem Beispiel sind zwei WS-Manager (QM1 und QM2) über zwei Kanäle miteinander verbunden (QM2.TO.QM1 und QM1.TO.QM2).

Beispiel 1-Aktivitätenberichte anfordern

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

In diesem Beispiel stellt die IBM MQ-Anwendung für die Routenanzeige eine Verbindung zu Warteschlangenmanager QM1 her und wird dazu verwendet, eine Trace-Route-Nachricht zu generieren und an die Zielwarteschlange TARGET.Q auf dem fernen Warteschlangenmanager QM2 zu übergeben. Die erforderliche Berichtsoption wird so angegeben, dass Aktivitätsberichte angefordert werden, wenn die Antwortnachricht für die Trace-Route weitergeleitet wird. Bei der Ankunft in der Zielwarteschlange wird die Trace-Route-Nachricht verworfen. Aktivitätsinformationen, die mithilfe von Aktivitätsberichten an die IBM MQ-Anwendung für die Routenanzeige zurückgegeben werden, werden sortiert und angezeigt.

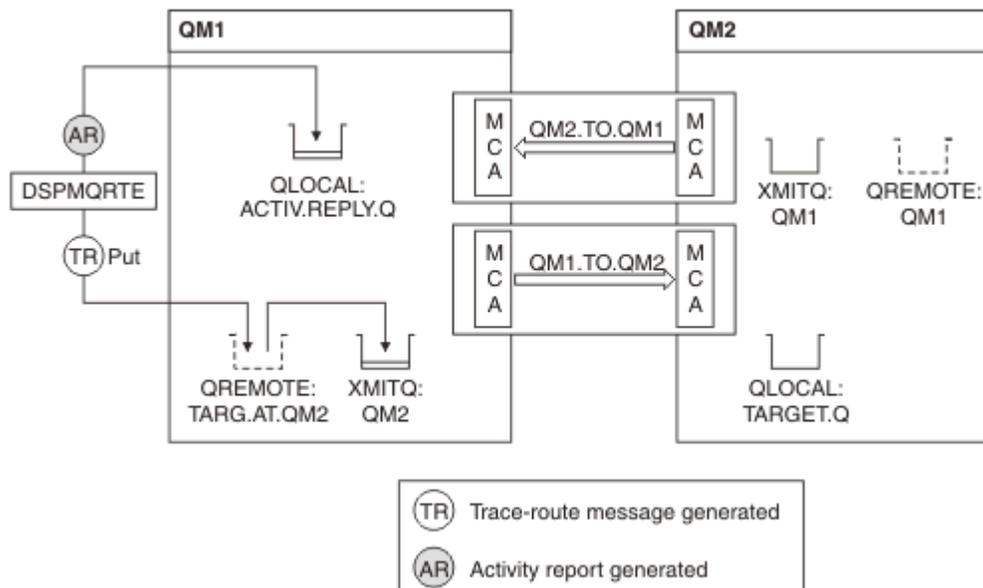


Abbildung 9. Aktivitätenberichte anfordern, Diagramm 1

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

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq ACTIV.REPLY.Q
```

QM1 ist der Name des Warteschlangenmanagers, mit dem die IBM MQ-Anwendung für die Routenanzeige eine Verbindung herstellt, TARG.AT.QM2 ist der Name der Zielwarteschlange und ACTIV.REPLY.Q ist der Name der Warteschlange, in die alle Antworten auf die Trace-Route-Nachricht gesendet werden soll.

Für alle nicht angegebenen Optionen werden Standardwerte angenommen, aber insbesondere die Option -f (die Trace-Route-Nachricht wird nur an einen Warteschlangenmanager weitergeleitet, der den Parameter "Deliver" der PCF-Gruppe "TraceRoute" berücksichtigt), die Option -d (bei der Ankunft, die Trace-Route-Nachricht wird nicht in die Zielwarteschlange gestellt), die Option -ro (MQRO_ACTIVITY und MQRO_DISCARD_MSG Berichtsoptionen angegeben) und die Option -t (mittlere Aktivität auf Detailebene wird aufgezeichnet).

- DSPMQRTE generiert die Trace-Route-Nachricht und stellt sie in die ferne Warteschlange TARG.AT.QM2.
- DSPMQRTE sieht dann den Wert des Attributs ACTIVREC des Warteschlangenmanagers QM1 aus. Der Wert ist MSG, daher generiert DSPMQRTE einen Aktivitätsbericht und stellt ihn in die Antwortwarteschlange ACTIV.REPLY.Q.

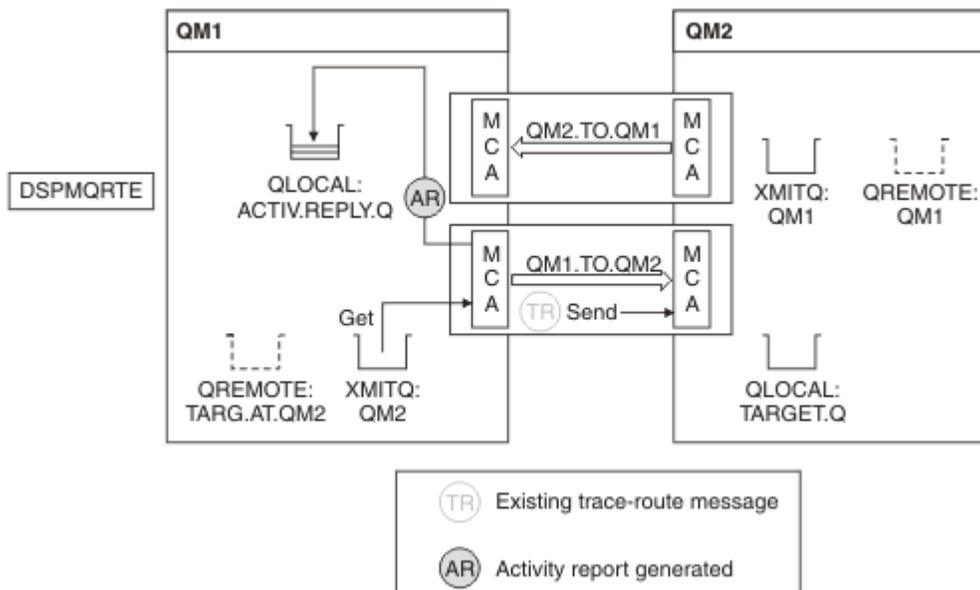


Abbildung 10. Aktivitätenberichte anfordern, Diagramm 2

- Der sendende Nachrichtenkanalagent (MCA) ruft die Trace-Route-Nachricht aus der Übertragungswarteschlange ab. Die Nachricht ist eine Trace-Route-Nachricht, daher beginnt der MCA mit der Aufzeichnung der Aktivitätsinformationen.
- Das Attribut ACTIVREC des Warteschlangenmanagers (QM1) lautet MSG, und die Option MQRO_ACTIVITY wird im Feld "Bericht" des Nachrichtendeskriptors angegeben. Daher generiert der Nachrichtenkanalmanager später einen Aktivitätsbericht. Der Wert des Parameters "RecordedActivities" in der PCF-Gruppe "TraceRoute" wird um 1 erhöht.
- Der MCA überprüft, ob der Wert für MaxActivities in der PCF-Gruppe "TraceRoute" nicht überschritten wurde.
- Bevor die Nachricht an QM2 weitergeleitet wird, folgt der MCA dem Algorithmus, der in [Weiterleiten](#) beschrieben ist (Schritte „1“ auf Seite 80, „4“ auf Seite 80 und „5“ auf Seite 80), und der Nachrichtenkanalmanager wählt die Nachricht aus.
- Der MCA generiert dann einen Aktivitätsbericht und stellt ihn in die Antwortwarteschlange (ACTIV.REPLY.Q).

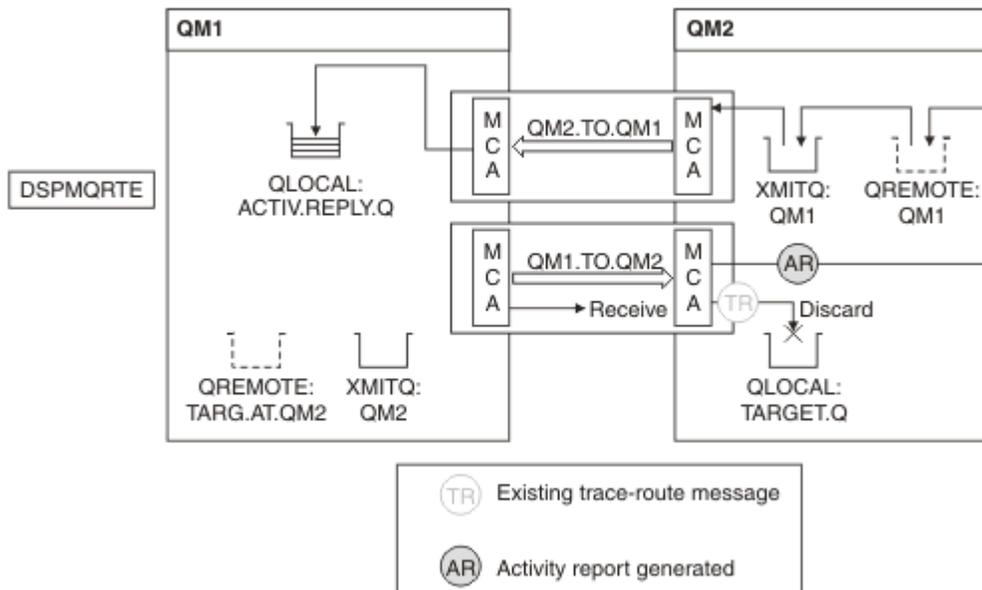


Abbildung 11. Aktivitätenberichte anfordern, Diagramm 3

- Der empfangende MCA empfängt die Trace-Route-Nachricht aus dem Kanal. Die Nachricht ist eine Trace-Route-Nachricht, daher beginnt der MCA mit der Aufzeichnung der Informationen über die Aktivität.
- Wenn der Warteschlangenmanager, von dem die Trace-Route-Nachricht stammt, von Version IBM WebSphere MQ 5.3.1 oder einer früheren Version stammt, erhöht der MCA den Parameter Discontinuity-Count des PCF-Trace-Leitwegs um 1. Dies ist hier nicht der Fall.
- Das Attribut ACTIVREC des Warteschlangenmanagers (QM2) ist MSG, und die Option MQRO_ACTIVITY wird angegeben, daher generiert der Nachrichtenkanalmanager einen Aktivitätsbericht. Der Wert des Parameters RecordedActivities wird um 1 erhöht.
- Die Zielwarteschlange ist eine lokale Warteschlange, daher wird die Nachricht mit dem Feedback MQFB_NOT_DELIVERED in Übereinstimmung mit dem Wert des Parameters Deliver in der PCF-Gruppe TraceRoute gelöscht.
- Der MCA generiert dann den endgültigen Aktivitätsbericht und stellt ihn in die Antwortwarteschlange. Dies wird in die Übertragungswarteschlange aufgelöst, die dem WS-Manager QM1 zugeordnet ist, und der Aktivitätsbericht wird an Warteschlangenmanager QM1 (ACTIV.REPLY.Q) zurückgegeben.

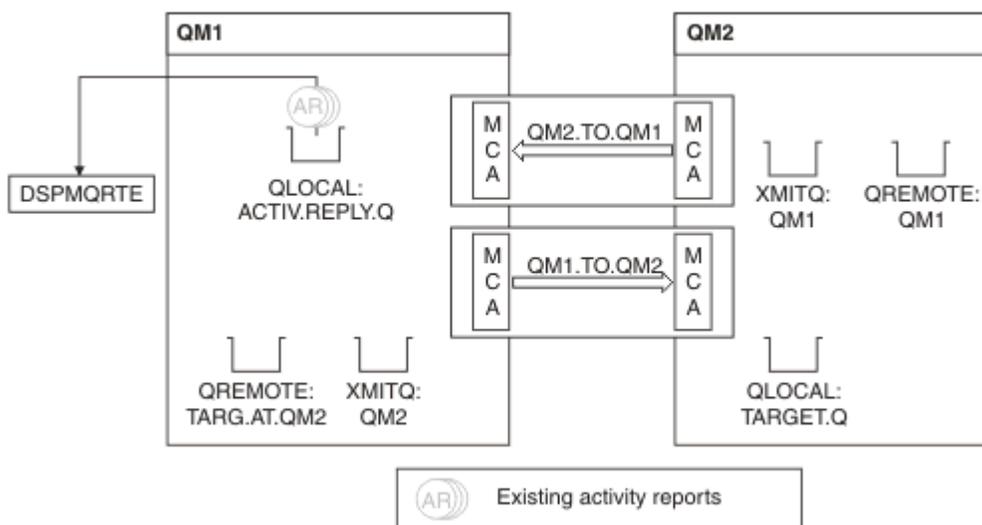


Abbildung 12. Aktivitätenberichte anfordern, Diagramm 4

- In der Zwischenzeit hat DSPMQRTE MQGETs in der Antwortwarteschlange (ACTIV.REPLY.Q) ausgeführt, die auf Aktivitätsberichte wartet. Es wartet bis zu 120 Sekunden (60 Sekunden länger als die Ablaufzeit der Trace-Route-Nachricht), da -w nicht angegeben wurde, als DSPMQRTE gestartet wurde.
- DSPMQRTE ruft die 3-Aktivität ab, die von der Antwortwarteschlange abgemeldet wird.
- Die Aktivitätsberichte werden unter Verwendung der Parameter "RecordedActivities", "UnrecordedActivities" und "DiscontinuityCount" in der PCF-Gruppe "TraceRoute" für jede der Aktivitäten geordnet. Der einzige Wert, der in diesem Beispiel ungleich null ist, ist RecordedActivities. Daher ist dieser Parameter der einzige tatsächlich verwendete Parameter.
- Das Programm wird beendet, sobald die Löschoption angezeigt wird. Obwohl die letzte Operation eine Löschoption war, wird sie so behandelt, als sei eine Nachricht in der Form, dass die Rückmeldung MQFB_NOT_DELIVERED ist.

Die Ausgabe, die angezeigt wird, folgt:

```

AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.

```

Beispiel 2-Anforderung einer Trace-Route-Antwortnachricht anfordern

Generieren und Bereitstellung einer Trace-Route-Nachricht an die Zielwarteschlange

In diesem Beispiel stellt die IBM MQ-Anwendung für die Routenanzeige eine Verbindung zu Warteschlangenmanager QM1 her und wird dazu verwendet, eine Trace-Route-Nachricht zu generieren und an die Zielwarteschlange TARGET.Q auf dem fernen Warteschlangenmanager QM2 zu übergeben. Die erforderliche Option wird so angegeben, dass die Aktivitätsinformationen in der Trace-Route-Nachricht akkumuliert werden. Bei der Ankunft in der Zielwarteschlange wird eine Trace-Route-Antwortnachricht angefordert, und die Trace-Route-Nachricht wird gelöscht.

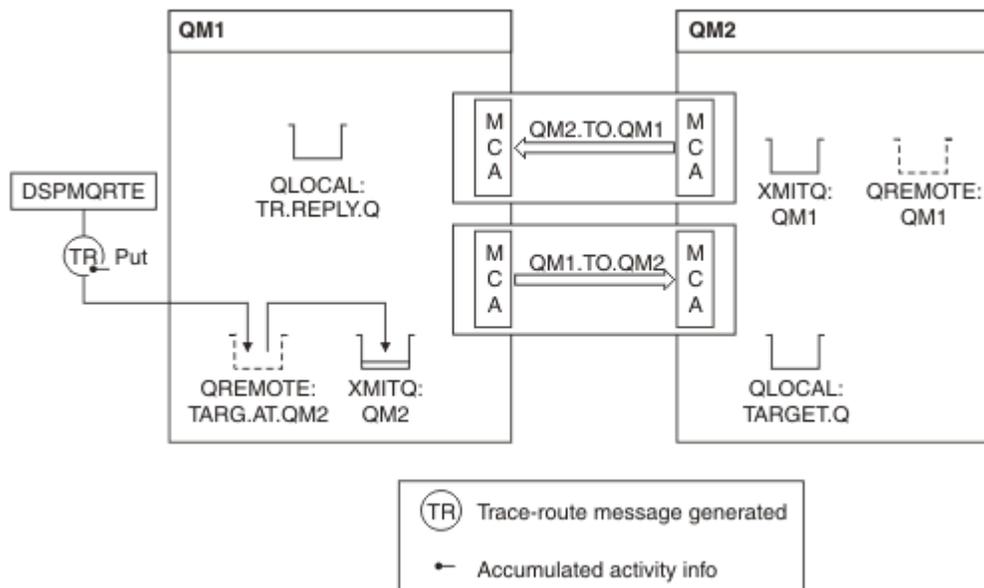


Abbildung 13. Anforderung einer Trace-Route-Antwortnachricht, Diagramm 1

- 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, mit dem die IBM MQ-Anwendung für die Routenanzeige eine Verbindung herstellt, TARG.AT.QM2 ist der Name der Zielwarteschlange und ACTIV.REPLY.Q ist der Name der Warteschlange, in die alle Antworten auf die Trace-Route-Nachricht gesendet werden soll. 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.

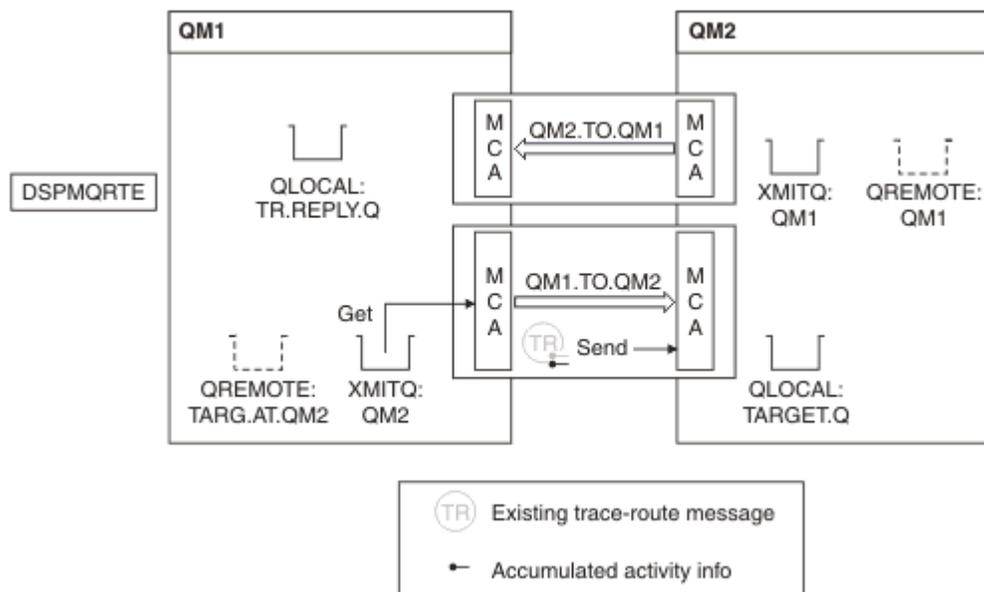


Abbildung 14. Antwort auf Trace-Route-Antwortnachricht anfordern, Diagramm 2

- 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.

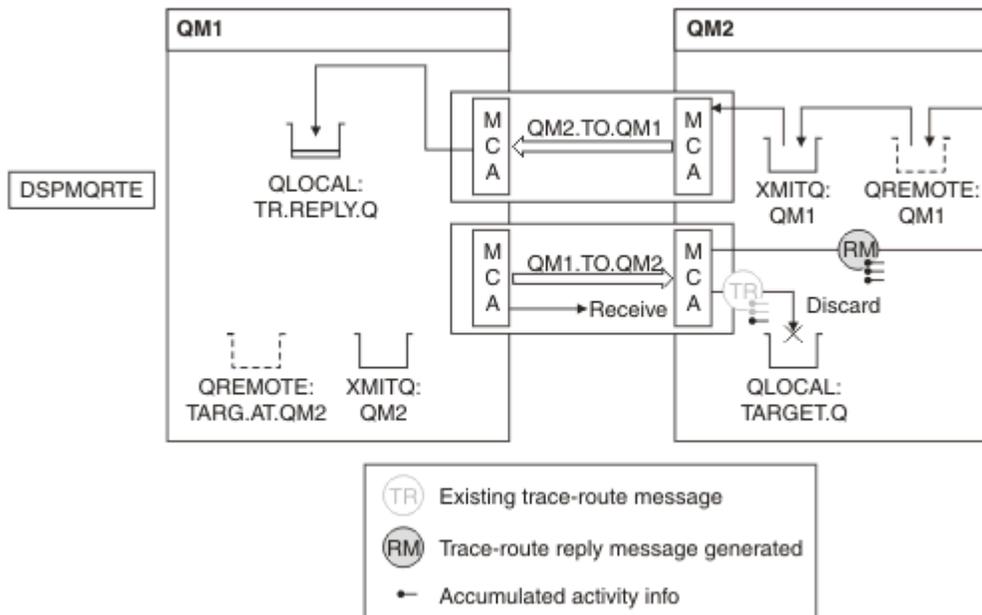


Abbildung 15. Antwort auf Trace-Route-Antwortnachricht anfordern, Diagramm 3

- 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.

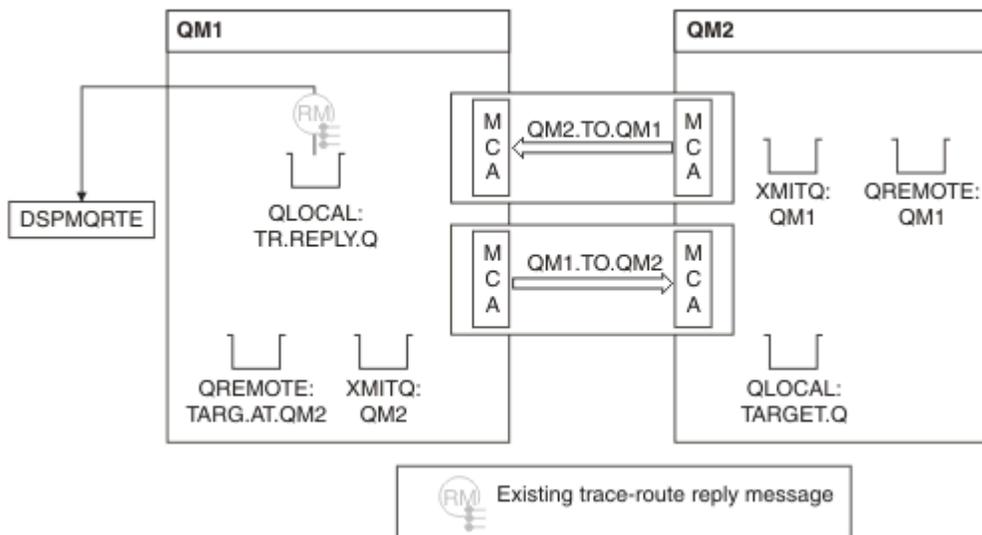


Abbildung 16. Antwort auf Trace-Route-Antwortnachricht anfordern, Diagramm 4

- In der Zwischenzeit wartet DSPMQRTE auf die Antwort auf die Trace-Route-Antwortnachricht, um in die Antwortwarteschlange zurückzukehren. Wenn er zurückkehrt, analysiert DSPMQRTE jede Aktivität, die er enthält, und druckt sie aus. Die letzte Operation ist eine Löschoperation. DSPMQRTE beendet, nachdem er gedruckt wurde.

Die Ausgabe, die angezeigt wird, folgt:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
TR.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

Beispiel 3-Übergabe von Aktivitätsberichten an die Systemwarteschlange

Es wird erkannt, wenn Aktivitätsberichte an andere Warteschlangen als die Warteschlange für zu beantwortende Nachrichten übergeben werden, und mit der IBM MQ-Anwendung für die Routenanzeige können Aktivitätsberichte aus der anderen Warteschlange gelesen werden.

Dieses Beispiel ist identisch mit dem „Beispiel 1-Aktivitätenberichte anfordern“ auf Seite 94, mit der Ausnahme, dass WSM2 jetzt den Wert des Attributs ACTIVREC queue management auf QUEUE gesetzt hat. Der Kanal QM1.TO.QM2 muss neu gestartet worden sein, damit diese Funktion wirksam wird.

In diesem Beispiel wird gezeigt, wie erkannt wird, wenn Aktivitätsberichte an andere Warteschlangen als die Warteschlange für Antwortantworten zugestellt werden. Nach dem Erkennen werden Aktivitätsberichte aus der anderen Warteschlange mit der IBM MQ-Anwendung zur Routenanzeige gelesen.

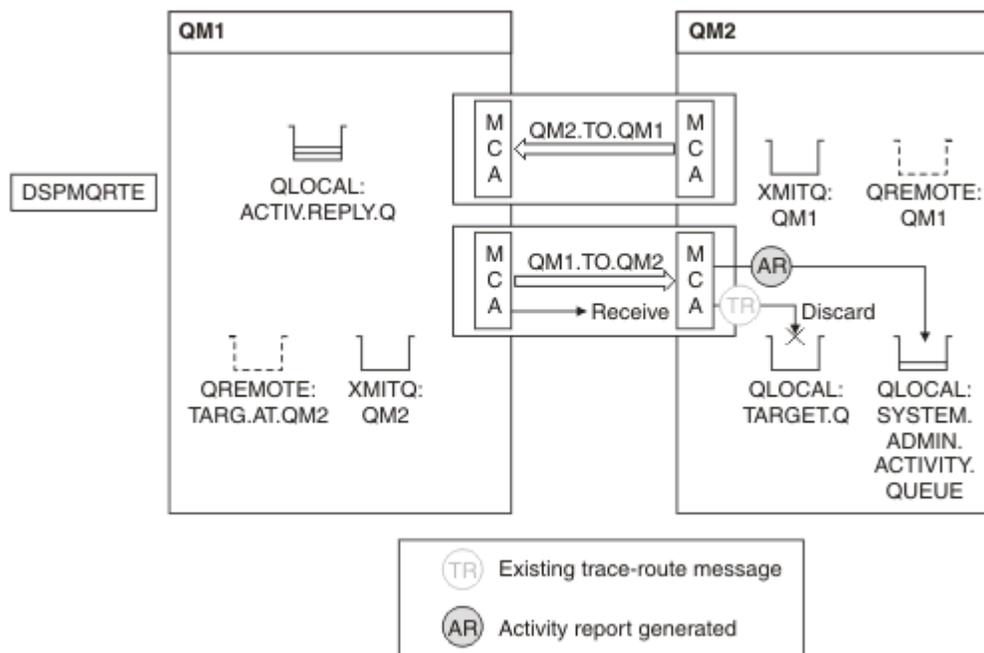


Abbildung 17. Aktivitätsberichte an die Systemwarteschlange übergeben, Diagramm 1

- Die Nachricht ist eine Trace-Route-Nachricht, daher beginnt der empfangende MCA mit dem Aufzeichnen von Informationen über die Aktivität.
- Der Wert des Warteschlangenmanagerattributs ACTIVREC auf QM2 ist jetzt QUEUE, daher generiert der Nachrichtenkanalmanager einen Aktivitätsbericht, stellt ihn jedoch in die Systemwarteschlange (SYSTEM.ADMIN.ACTIVITY.QUEUE) und nicht in die Antwortwarteschlange (ACTIV.REPLY.Q).

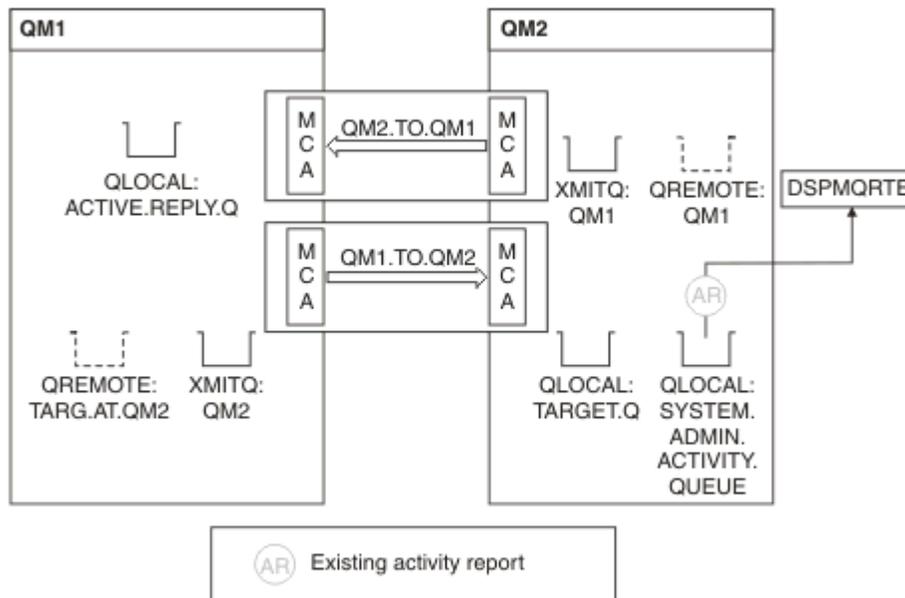


Abbildung 18. Aktivitätsberichte an die Systemwarteschlange übergeben, Diagramm 2

- In der Zwischenzeit hat DSPMQRTE auf Aktivitätsberichte gewartet, um in ACTIV.REPLY.Q. einzutreffen. Nur zwei kommen. DSPMQRTE wartet weiterhin 120 Sekunden, da es den Anschein hat, dass die Route noch nicht abgeschlossen ist.

Die Ausgabe, die angezeigt wird, folgt:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
ACTIV.REPLY.Q -v outline identifiers'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
```

```
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'
```

```
Operation:
  OperationType: Put
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'TARG.AT.QM2'
  ResolvedQName: 'QM2'
  RemoteQName: 'TARGET.Q'
  RemoteQMgrName: 'QM2'
```

```
-----
Activity:
  ApplName: 'cann\output\bin\runmqchl.EXE'
```

```
Operation:
  OperationType: Get
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001505'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001502'
```

```
EmbeddedMQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'QM2'
  ResolvedQName: 'QM2'
```

```
Operation:
  OperationType: Send
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  RemoteQMgrName: 'QM2'
  ChannelName: 'QM1.TO.QM2'
  ChannelType: Sender
  XmitQName: 'QM2'
```

```
-----
AMQ8652: DSPMQRTE command has finished.
```

- Die letzte Operation, die DSPMQRTE beobachtet hat, war eine Sendung, daher wird der Kanal ausgeführt. Jetzt müssen wir herausfinden, warum wir keine weiteren Aktivitätsberichte vom WS-Manager QM2 erhalten haben (wie in RemoteQMgrName angegeben).
- Um zu überprüfen, ob Aktivitätsinformationen in der Systemwarteschlange vorhanden sind, starten Sie DSPMQRTE auf QM2, um zu versuchen, weitere Aktivitätsberichte zu erfassen. Verwenden Sie den folgenden Befehl, um DSPMQRTE zu starten:

```
dspmqrte -m QM2 -q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C9154220001502 -v outline
```

Dabei gilt: 414D51204C41524745512020202020A3C9154220001502 ist die MsgId der Trace-Route-Nachricht, die in die Datei gestellt wurde.

- DSPMQRTE führt dann erneut eine Folge von MQGETs aus und wartet auf Antworten in der Systemaktivitätswarteschlange, die sich auf die Trace-Route-Nachricht mit der angegebenen ID bezieht.

- DSPMQRTE ruft einen weiteren Aktivitätsbericht ab, den er anzeigt. DSPMQRTE stellt fest, dass die vorangegangenen Aktivitätsberichte fehlen, und zeigt eine Nachricht an, die dies angibt. Wir wissen aber schon über diesen Teil der Strecke.

Die Ausgabe, die angezeigt wird, folgt:

```

AMQ8653: DSPMQRTE command started with options '-m QM2
-q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C915420001502 -v outline'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----

Activity:
  Activity information unavailable.

-----

Activity:
  ApplName: 'cann\output\bin\AMQRMPPA.EXE'

  Operation:
    OperationType: Receive
    QMgrName: 'QM2'
    RemoteQMgrName: 'QM1'
    ChannelName: 'QM1.TO.QM2'
    ChannelType: Receiver

  Operation:
    OperationType: Discard
    QMgrName: 'QM2'
    QName: 'TARGET.Q'
    Feedback: NotDelivered

-----

AMQ8652: DSPMQRTE command has finished.

```

- Dieser Aktivitätsbericht zeigt an, dass die Routeninformationen jetzt abgeschlossen sind. Es ist kein Fehler aufgetreten.
- Nur weil Leitweginformationen nicht verfügbar sind oder weil DSPMQRTE nicht die gesamte Route anzeigen kann, bedeutet dies nicht, dass die Nachricht nicht zugestellt wurde. Beispielsweise können die WS-Manager-Attribute unterschiedlicher WS-Manager unterschiedlich sein, oder es ist möglicherweise keine Antwortwarteschlange definiert, um die Antwort zurückerhalten zu können.

Beispiel 4-Kanalfehler diagnostizieren

Diagnostizieren eines Problems, bei dem die Trace-Route-Nachricht die Zielwarteschlange nicht erreicht

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

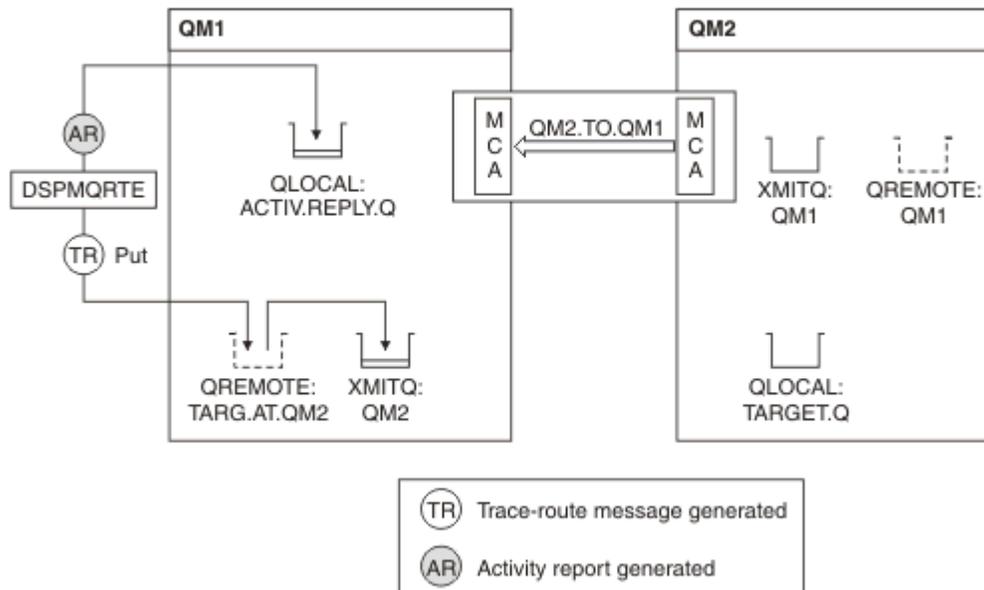


Abbildung 19. Kanalfehler diagnostizieren

- In diesem Beispiel wird der Kanal QM1.TO.QM2 nicht ausgeführt.
- Mit dem Befehl DSPMQRTE wird eine Trace-Route-Nachricht (wie in Beispiel 1) in die Zielwarteschlange gestellt und ein Aktivitätsbericht generiert.
- Es gibt keinen MCA zum Abrufen der Nachricht aus der Übertragungswarteschlange (WSM2). Daher ist dies der einzige Aktivitätsbericht, den DSPMQRTE aus der Antwortwarteschlange zurückerhält. Diesmal deutet die Tatsache, dass die Route nicht vollständig ist, auf ein Problem hin. Der Administrator kann die in ResolvedQName gefundene Übertragungswarteschlange verwenden, um zu ermitteln, warum die Übertragungswarteschlange nicht gewartet wird.

Die Ausgabe, die angezeigt wird, folgt:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q -v outline'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'

  Operation:
    OperationType: Put
    QMgrName: 'QM1'
    QName: 'TARG.AT.QM2'
    ResolvedQName: 'QM2'
    RemoteQName: 'TARGET.Q'
    RemoteQMgrName: 'QM2'
-----
AMQ8652: DSPMQRTE command has finished.
```

Aktivitätsberichtsreferenz

Verwenden Sie diese Seite, um eine Übersicht über das Nachrichtenformat des Aktivitätsberichts zu erhalten. Die Nachrichtendaten des Aktivitätsberichts enthalten die Parameter, die die Aktivität beschreiben.

Aktivitätsberichtsformat

Aktivitätsberichte sind standardmäßige IBM MQ-Berichtsnachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Aktivitätsberichte sind PCF-Nachrichten, die von Anwendungen gene-

riert werden, die eine Aktivität im Namen einer Nachricht ausgeführt haben, da sie über ein Warteschlangenmanagernetzwerk weitergeleitet wurde.

Aktivitätsberichte enthalten die folgenden Informationen:

Ein Nachrichtendeskriptor

Eine MQMD-Struktur

Nachrichtendaten

Befasst sich wie folgt:

- Ein eingebetteter PCF-Header (MQEPH).
- Nachrichtendaten des Aktivitätsberichts.

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

Tabelle 19 auf Seite 106 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 Flaggen 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 Ferner Warteschlangenmanagername ² ³ ⁴ ⁵ ⁷ Subskriptionsebene ⁹ Subskriptionskennung ⁹ Feedback ² ¹⁰ Kanalname ⁴ ⁵ Kanaltyp ⁴ ⁵ Name der Übertragungswarteschlange ⁵ TraceRoute ⁶ Detail Aufgezeichnete Aktivitäten Nicht aufgezeichnete Aktivitäten Diskontinuitätszähler Max. Aktivitäten Kumulieren Deliver

Anmerkungen:

1. Wird für Get- und Browse-Operationen zurückgegeben.
2. Wird für Discard-Operationen zurückgegeben.
3. Zurückgegeben für Put-, Put-Reply- und Put-Report-Operationen.

4. Zurückgegeben für Empfangsoperationen.
5. Wird für Sendeoperationen zurückgegeben.
6. Wird für Trace-Route-Nachrichten zurückgegeben.
7. Nicht zurückgegeben für 'Put' -Operationen in ein Thema, das in Publish-Aktivitäten enthalten ist.
8. Für ausgeschlossene Publish-Operationen nicht zurückgegeben. Für Publish- und Discarded Publish-Operationen, die eine Untergruppe von Parametern enthalten.
9. Zurückgegeben für Publish, Discarded Publish und Excluded Publish-Operationen.
10. Zurückgegeben bei Veröffentlichen von Veröffentlichen und Ausgeschlossenen Publish-Operationen.

MQMD des Aktivitätsberichts (Nachrichtendeskriptor)

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

StrucId

Struktur-ID:

Datentyp

MQCHAR4

Wert

MQMD_STRUC_ID.

Version

Strukturversionsnummer

Datentyp

MQLONG

Werte

Kopiert aus dem ursprünglichen Nachrichtendeskriptor. Mögliche Werte:

MQMD_VERSION_1

Version-1 Nachrichtendeskriptorstruktur, die in allen Umgebungen unterstützt wird.

MQMD_VERSION_2

Version 2 der Nachrichtendeskriptorstruktur, die in den folgenden Umgebungen unterstützt wird:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

und auf allen IBM MQ MQI clients, die mit diesen Systemen verbunden sind.

Report

Optionen für weitere Berichtsnachrichten

Datentyp

MQLONG

Wert

Wenn MQRO_PASS_DISCARD_AND_EXPIRY oder MQRO_DISCARD_MSG im Feld *Report* des ursprünglichen Nachrichtendeskriptors angegeben wurden:

MQRO_DISCARD

Der Bericht wird gelöscht, wenn er nicht an die Zielwarteschlange übergeben werden kann.

Andernfalls:

MQRO_NONE

Kein Bericht erforderlich

MsgType

Gibt den Typ der Nachricht an.

Datentyp

MQLONG

Wert

MQMT_REPORT

Expiry

Lebensdauer der Berichtsnachricht

Datentyp

MQLONG

Wert

Wenn das Feld *Report* im ursprünglichen Nachrichtendeskriptor als MQRO_PASS_DISCARD_AND_EXPIRY angegeben wird, wird die verbleibende Ablaufzeit von der ursprünglichen Nachricht verwendet.

Andernfalls:

MQEI_UNLIMITED

Der Bericht hat keine Ablaufzeit.

Feedback

Beschreibung: Rückmeldungs- oder Ursachencode.

Datentyp: MQLONG.

Wert: **MQFB_ACTIVITY**
Tätigkeitsbericht.**Encoding**

Beschreibung: Numerische Codierung von Berichtsnachrichtendaten.

Datentyp: MQLONG.

Wert: MQENC_NATIVE.

CodedCharSetId

Beschreibung: Zeichensatzes-ID der Berichtsnachrichtendaten.

Datentyp: MQLONG.

Wert: Legen Sie die entsprechende Gruppe fest.

Format

Beschreibung: Formatname der Berichtsnachrichtendaten

Datentyp: MQCHAR8.

Wert: **MQFMT_EMBEDDED_PCF**
Eingebettete PCF-Nachricht**Priority**

Beschreibung: Nachrichtenpriorität melden.

Datentyp: MQLONG.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

Persistence

Beschreibung: Nachrichtenpersistenz berichten.

Datentyp: MQLONG.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

MsgId

Beschreibung: Nachrichten-ID.

Datentyp: MQBYTE24.

Werte: Wenn das Feld *Report* im ursprünglichen Nachrichtendeskriptor als 'MQRO_PASS_MSG_ID' angegeben ist, wird die Nachrichten-ID aus der ursprünglichen Nachricht verwendet.

Andernfalls wird vom WS-Manager ein eindeutiger Wert generiert.

CorrelId

Beschreibung: Korrelations-ID.

Datentyp: MQBYTE24.

Wert: Wenn das Feld *Report* im ursprünglichen Nachrichtendeskriptor als 'MQRO_PASS_CORREL_ID' angegeben ist, wird die Korrelations-ID aus der ursprünglichen Nachricht verwendet.

Andernfalls wird die Nachrichten-ID aus der ursprünglichen Nachricht kopiert.

BackoutCount

Beschreibung: Zurücksetzungszähler.

Datentyp: MQLONG.

Wert: 0.

ReplyToQ

Beschreibung: Name der Antwortwarteschlange.

Datentyp: MQCHAR48.

Werte: Leer.

ReplyToQMgr

Beschreibung: Antwort-Warteschlangenmanagername

Datentyp: MQCHAR48.

Wert: Der Name des Warteschlangenmanagers, der die Berichtsnachricht generiert hat.

UserIdentifier

Beschreibung: Die Benutzer-ID der Anwendung, die die Berichtsnachricht generiert hat.

Datentyp: MQCHAR12.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

AccountingToken

Beschreibung: Abrechnungstoken, das es einer Anwendung ermöglicht, die Arbeit zu berechnen, die als Ergebnis der Nachricht ausgeführt wurde.
Datentyp: MQBYTE32.
Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

ApplIdentityData

Beschreibung: Identitätsbezogene Anwendungsdaten.
Datentyp: MQCHAR32.
Werte: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

PutApplType

Beschreibung: Typ der Anwendung, die die Berichtsnachricht eingibt.
Datentyp: MQLONG.
Wert: **MQAT_QMGR**
Nachricht des vom WS-Manager generierten Nachrichten.

PutApplName

Beschreibung: Name der Anwendung, die die Berichtsnachricht eingibt.
Datentyp: MQCHAR28.
Wert: Entweder die ersten 28 Byte des Warteschlangenmanagernamens oder der Name des MCA, der die Berichtsnachricht generiert hat.

PutDate

Beschreibung: Datum, an dem die Nachricht eingereicht wurde.
Datentyp: MQCHAR8.
Wert: Wie vom WS-Manager generiert.

PutTime

Beschreibung: Zeitpunkt, zu dem die Nachricht eingereicht wurde.
Datentyp: MQCHAR8.
Wert: Wie vom WS-Manager generiert.

ApplOriginData

Beschreibung: Anwendungsdaten zum Ursprung
Datentyp: MQCHAR4.
Wert: Leer.

Wenn *Version* MQMD_VERSION_2 ist, sind die folgenden zusätzlichen Felder vorhanden:

GroupId

Beschreibung: Gibt an, zu welcher Nachrichtengruppe oder logischen Nachricht die physische Nachricht gehört.
Datentyp: MQBYTE24.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

MsgSeqNumber

Beschreibung: Folgenummer der logischen Nachricht in einer Gruppe

Datentyp: MQLONG.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

Offset

Beschreibung: Relative Adresse von Daten in einer physischen Nachricht ab dem Anfang der logischen Nachricht.

Datentyp: MQLONG.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

MsgFlags

Beschreibung: Nachrichtenmarkierungen, die Attribute der Nachricht angeben oder ihre Verarbeitung steuern.

Datentyp: MQLONG.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

OriginalLength

Beschreibung: Länge der ursprünglichen Nachricht

Datentyp: MQLONG.

Wert: Kopiert aus dem ursprünglichen Nachrichtendeskriptor.

Aktivitätenbericht MQEPH (eingebetteter PCF-Header)

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

Die MQEPH-Struktur enthält eine Beschreibung der PCF-Informationen, die die Nachrichtendaten eines Aktivitätsberichts enthalten, sowie die Anwendungsnachrichtendaten, die auf sie folgen.

Für einen Aktivitätsbericht enthält die MQEPH-Struktur die folgenden Werte:

StrucId

Beschreibung: Struktur-ID.

Datentyp: MQCHAR4.

Wert: MQEPH_STRUC_ID.

Version

Beschreibung: Strukturversionsnummer.

Datentyp: MQLONG.

Werte: MQEPH_VERSION_1.

StrucLength

Beschreibung: Strukturlänge.

Datentyp: MQLONG.

Wert: Gesamtlänge der Struktur einschließlich der PCF-Parameterstrukturen, die auf diese Struktur folgen.

Encoding

Beschreibung: Numerische Codierung der Nachrichtendaten, die auf die letzte PCF-Parameterstruktur folgen.

Datentyp: MQLONG.

Wert: Wenn Daten aus den ursprünglichen Anwendungsnachrichtendaten in der Berichtsnachricht enthalten sind, wird der Wert aus dem Feld *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 112.

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: **MQCFT_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 Trace-Route-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:	<i>Activity</i> .
Zurückgegeben:	Immer.

ActivityDescription

Beschreibung:	Beschreibung der Aktivität, die von der Anwendung ausgeführt wird.
ID:	MQCACF_ACTIVITY_DESCRIPTION.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Activity</i> .
Maximale Länge:	64
Zurückgegeben:	Immer.

Operation

Beschreibung:	Gruppierte Parameter, die eine Operation der Aktivität beschreiben.
ID:	MQGACF_OPERATION.
Datentyp:	MQCFGR.
In PCF-Gruppe enthalten:	<i>Activity</i> .
Parameter in PCF- Gruppe:	<i>OperationType</i> <i>OperationDate</i> <i>OperationTime</i> <i>Message</i> <i>QMgrName</i> <i>QSGName</i>

Anmerkung: Je nach Operationstyp werden in dieser Gruppe zusätzliche Parameter zurückgegeben. Diese zusätzlichen Parameter werden als Operation-spezifische Aktivitätsberichtsnachrichtendaten beschrieben.

Zurückgegeben: Eine *Operation* -PCF-Gruppe pro Operation in der Aktivität.

OperationType

Beschreibung:	Typ der ausgeführten Operation.
ID:	MQIACF_OPERATION_TYPE.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation</i> .
Werte:	MQOPER_ *.
Zurückgegeben:	Immer.

OperationDate

Beschreibung:	Datum, an dem die Operation ausgeführt wurde.
ID:	MQCACF_OPERATION_DATE.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_DATE_LENGTH.
Zurückgegeben:	Immer.

OperationTime

Beschreibung:	Zeit, zu der die Operation ausgeführt wurde.
ID:	MQCACF_OPERATION_TIME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_TIME_LENGTH.
Zurückgegeben:	Immer.

Message

Beschreibung:	Gruppierte Parameter, die die Nachricht beschreiben, die die Aktivität verursacht hat.
ID:	MQGACF_MESSAGE.
Datentyp:	MQCFGR.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Parameter in Gruppe:	<i>MsgLength</i> <i>MQMD</i> <i>EmbeddedMQMD</i>
Zurückgegeben:	Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

MsgLength

Beschreibung:	Länge der Nachricht, die die Aktivität verursacht hat, bevor die Aktivität auftrat.
ID:	MQIACF_MSG_LENGTH.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Message.</i>
Zurückgegeben:	Immer.

MQMD

Beschreibung:	Gruppierte Parameter, die sich auf den Nachrichtendeskriptor der Nachricht beziehen, die die Aktivität verursacht hat.
ID:	MQGACF_MQMD.

Datentyp: MQCFGR.

In PCF-Gruppe
enthalten: *Message.*

Parameter in
Gruppe: *StrucId*
Version
Report
MsgType
Expiry
Feedback
Encoding
CodedCharSetId
Format
Priority
Persistence
MsgId
CorrelId
BackoutCount
ReplyToQ
ReplyToQMgr
UserIdentifier
AccountingToken
ApplIdentityData
PutApplType
PutApplName
PutDate
PutTime
ApplOriginData
GroupId
MsgSeqNumber
Offset
MsgFlags
OriginalLength

Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

EmbeddedMQMD

Beschreibung: Gruppierete Parameter, die den Nachrichtendeskriptor beschreiben, der in eine Nachricht in einer Übertragungswarteschlange eingebettet ist.

ID: MQGACF_EMBEDDED_MQMD.

Datentyp: MQCFGR.

In PCF-Gruppe
enthalten: *Message.*

Parameter in Gruppe:	<i>StrucId</i> <i>Version</i> <i>Report</i> <i>MsgType</i> <i>Expiry</i> <i>Feedback</i> <i>Encoding</i> <i>CodedCharSetId</i> <i>Format</i> <i>Priority</i> <i>Persistence</i> <i>MsgId</i> <i>CorrelId</i> <i>BackoutCount</i> <i>ReplyToQ</i> <i>ReplyToQMgr</i> <i>UserIdentifier</i> <i>AccountingToken</i> <i>ApplIdentityData</i> <i>PutApplType</i> <i>PutApplName</i> <i>PutDate</i> <i>PutTime</i> <i>ApplOriginData</i> <i>GroupId</i> <i>MsgSeqNumber</i> <i>Offset</i> <i>MsgFlags</i> <i>OriginalLength</i>
----------------------	---

Zurückgegeben: Für Get-Operationen, bei denen die Warteschlange in eine Übertragungswarteschlange aufgelöst wird.

StrucId

Beschreibung:	Struktur-ID
ID:	MQCACF_STRUC_ID.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Maximale Länge:	4.
Zurückgegeben:	Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

Version

Beschreibung:	Strukturversionsnummer.
ID:	MQIACF_VERSION.
Datentyp:	MQCFIN.

In PCF-Gruppe
enthalten: *MQMD* oder *EmbeddedMQMD*.

Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

Report

Beschreibung: Optionen für Berichtsnachrichten.

ID: MQIACF_REPORT.

Datentyp: MQCFIN.

In PCF-Gruppe
enthalten: *MQMD* oder *EmbeddedMQMD*.

Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

MsgType

Beschreibung: Gibt den Typ der Nachricht an.

ID: MQIACF_MSG_TYPE.

Datentyp: MQCFIN.

In PCF-Gruppe
enthalten: *MQMD* oder *EmbeddedMQMD*.

Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

Expiry

Beschreibung: Lebensdauer der Nachricht

ID: MQIACF_EXPIRY.

Datentyp: MQCFIN.

In PCF-Gruppe
enthalten: *MQMD* oder *EmbeddedMQMD*.

Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

Feedback

Beschreibung: Rückmeldungs- oder Ursachencode.

ID: MQIACF_FEEDBACK.

Datentyp: MQCFIN.

In PCF-Gruppe
enthalten: *MQMD* oder *EmbeddedMQMD*.

Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

Encoding

Beschreibung: Numerische Codierung von Nachrichtendaten.

ID: MQIACF_ENCODING.

Datentyp: MQCFIN.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

CodedCharSetId

Beschreibung: Zeichensatz-ID der Nachrichtendaten.
ID: MQIA_CODED_CHAR_SET_ID.
Datentyp: MQCFIN.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

Format

Beschreibung: Name des Formats von Nachrichtendaten
ID: MQCACH_FORMAT_NAME.
Datentyp: MQCFST.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Maximale Länge: MQ_FORMAT_LENGTH.
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

Priority

Beschreibung: Nachrichtenpriorität
ID: MQIACF_PRIORITY.
Datentyp: MQCFIN.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

Persistence

Beschreibung: Nachrichtenpersistenz
ID: MQIACF_PERSISTENCE.
Datentyp: MQCFIN.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

MsgId

Beschreibung: Nachrichten-ID.
ID: MQBACF_MSG_ID.

Datentyp: MQCFBS.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Maximale Länge: MQ_MSG_ID_LENGTH.
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

CorrelId

Beschreibung: Korrelations-ID.
ID: MQBACF_CORREL_ID.
Datentyp: MQCFBS.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Maximale Länge: MQ_CORREL_ID_LENGTH.
Zurückgegeben: Immer mit Ausnahme von Ausgeschlossenen Publish-Operationen.

BackoutCount

Beschreibung: Zurücksetzungszähler.
ID: MQIACF_BACKOUT_COUNT.
Datentyp: MQCFIN.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Zurückgegeben: Immer mit Ausnahme der Operationen 'Veröffentlichen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

ReplyToQ

Beschreibung: Name der Antwortwarteschlange.
ID: MQCACF_REPLY_TO_Q.
Datentyp: MQCFST.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Maximale Länge: MQ_Q_NAME_LENGTH.
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish- und Discarded Publish-Operationen.

ReplyToQMGr

Beschreibung: Antwort-Warteschlangenmanagername
ID: MQCACF_REPLY_TO_Q_MGR.
Datentyp: MQCFST.
In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.
Maximale Länge: MQ_Q_MGR_NAME_LENGTH.
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

UserIdentifier

Beschreibung:	Die Benutzer-ID der Anwendung, die die Nachricht erstellt hat.
ID:	MQCACF_USER_IDENTIFIER.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Maximale Länge:	MQ_USER_ID_LENGTH.
Zurückgegeben:	Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen'.

AccountingToken

Beschreibung:	Abrechnungstoken, das es einer Anwendung ermöglicht, die Arbeit zu berechnen, die als Ergebnis der Nachricht ausgeführt wurde.
ID:	MQBACF_ACCOUNTING_TOKEN.
Datentyp:	MQCFBS.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Maximale Länge:	MQ_ACCOUNTING_TOKEN_LENGTH.
Zurückgegeben:	Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen'.

ApplIdentityData

Beschreibung:	Identitätsbezogene Anwendungsdaten.
ID:	MQCACF_APPL_IDENTITY_DATA.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Maximale Länge:	MQ_APPL_IDENTITY_DATA_LENGTH.
Zurückgegeben:	Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen'.

PutApplType

Beschreibung:	Typ der Anwendung, die die Nachricht eingereicht hat.
ID:	MQIA_APPL_TYPE.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .
Zurückgegeben:	Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

PutApplName

Beschreibung:	Name der Anwendung, die die Nachricht einreicht.
ID:	MQCACF_APPL_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>MQMD</i> oder <i>EmbeddedMQMD</i> .

Maximale Länge: MQ_APPL_NAME_LENGTH.
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

PutDate

Beschreibung: Datum, an dem die Nachricht eingereicht wurde.
ID: MQCACF_PUT_DATE.
Datentyp: MQCFST.
In PCF-Gruppe enthalten: MQMD oder EmbeddedMQMD.
Maximale Länge: MQ_PUT_DATE_LENGTH.
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

PutTime

Beschreibung: Zeitpunkt, zu dem die Nachricht eingereicht wurde.
ID: MQCACF_PUT_TIME.
Datentyp: MQCFST.
In PCF-Gruppe enthalten: MQMD oder EmbeddedMQMD.
Maximale Länge: MQ_PUT_TIME_LENGTH.
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

ApplOriginData

Beschreibung: Anwendungsdaten zum Ursprung
ID: MQCACF_APPL_ORIGIN_DATA.
Datentyp: MQCFST.
In PCF-Gruppe enthalten: MQMD oder EmbeddedMQMD.
Maximale Länge: MQ_APPL_ORIGIN_DATA_LENGTH.
Zurückgegeben: Immer, mit Ausnahme von 'Ausgeschlossene Bereitstellungsoperationen' und 'In MQMD' für Publish/Discarded Publish Operations.

GroupId

Beschreibung: Gibt an, zu welcher Nachrichtengruppe oder logischen Nachricht die physische Nachricht gehört.
ID: MQBACF_GROUP_ID.
Datentyp: MQCFBS.
In PCF-Gruppe enthalten: MQMD oder EmbeddedMQMD.
Maximale Länge: MQ_GROUP_ID_LENGTH.

Zurückgegeben: Wenn der *Version* als MQMD_VERSION_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

MsgSeqNumber

Beschreibung: Folgenummer der logischen Nachricht in einer Gruppe

ID: MQIACH_MSG_SEQUENCE_NUMBER.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.

Zurückgegeben: Wenn *Version* als MQMD_VERSION_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

Offset

Beschreibung: Relative Adresse von Daten in einer physischen Nachricht ab dem Anfang der logischen Nachricht.

ID: MQIACF_OFFSET.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.

Zurückgegeben: Wenn *Version* als MQMD_VERSION_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

MsgFlags

Beschreibung: Nachrichtenmarkierungen, die Attribute der Nachricht angeben oder ihre Verarbeitung steuern.

ID: MQIACF_MSG_FLAGS.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.

Zurückgegeben: Wenn *Version* als MQMD_VERSION_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

OriginalLength

Beschreibung: Länge der ursprünglichen Nachricht

ID: MQIACF_ORIGINAL_LENGTH.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten: MQMD oder *EmbeddedMQMD*.

Zurückgegeben: Wenn *Version* als MQMD_VERSION_2 angegeben ist. Nicht in 'Ausgeschlossene Bereitstellungsoperationen' und 'MQMD' für Publish/Discarded Publish Operations zurückgegeben.

QMgrName

Beschreibung:	Der Name des Warteschlangenmanagers, in dem die Aktivität ausgeführt wurde.
ID:	MQCA_Q_MGR_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<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 in einem IBM MQ for z/OS-Warteschlangenmanager ausgeführt wurde.

TraceRoute

Beschreibung:	Gruppierte Parameter, die Attribute der Trace-Route-Nachricht angeben.
ID:	MQGACF_TRACE_ROUTE.
Datentyp:	MQCFGR.
Enthalten in PCF- Gruppe:	<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 Trace-Route-Nachricht zu dem Zeitpunkt, zu dem der Aktivitätsbericht generiert wurde.

Nachrichtendaten für operationsspezifische Aktivitätsberichte

Verwenden Sie diese Seite, um die zusätzlichen PCF-Parameter anzuzeigen, die in einem Aktivitätsbericht in der PCF-Gruppe *Operation* zurückgegeben werden können, abhängig vom Wert des Parameters *OperationType*.

Die zusätzlichen Parameter variieren in Abhängigkeit von den folgenden Operationstypen:

Abrufen/Durchsuchen (MQOPER_GET/MQOPER_BROWSE)

Die zusätzlichen Aktivitätsberichtsnachrichtendatenparameter, die in der PCF-Gruppe *Operation* für den Operationstyp Get/Browse (MQOPER_GET/MQOPER_BROWSE) zurückgegeben werden (eine Nachricht in einer Warteschlange wurde erhalten oder durchsucht).

QName

Beschreibung:	Der Name der Warteschlange, die geöffnet wurde.
ID:	MQCA_Q_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Immer.

ResolvedQName

Beschreibung:	Der Name, in den die geöffnete Warteschlange aufgelöst wird.
ID:	MQCACF_RESOLVED_Q_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Immer.

Discard (MQOPER_DISCARD)

Die zusätzlichen Nachrichtendatenparameter für Aktivitäten, die in der PCF-Gruppe *Operation* für den Operationstyp 'Discard (MQOPER_DISCARD)' zurückgegeben werden (eine Nachricht wurde gelöscht).

Feedback

Beschreibung:	Die Ursache für die Nachricht, die gelöscht wird.
ID:	MQIACF_FEEDBACK.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Immer.

QName

Beschreibung:	Der Name der Warteschlange, die geöffnet wurde.
ID:	MQCA_Q_NAME.
Datentyp:	MQCFST.
Maximale Länge:	MQ_Q_NAME_LENGTH
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Wenn die Nachricht gelöscht wurde, weil sie nicht erfolgreich in eine Warteschlange gestellt wurde.

RemoteQMgrName

Beschreibung:	Der Name des Warteschlangenmanagers, für den die Nachricht bestimmt war.
ID:	MQCA_REMOTE_Q_MGR_NAME.
Datentyp:	MQCFST.
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Wenn der Wert von <i>Feedback</i> MQFB_NOT_FORWARDED ist.

Publish/Discarded Publish/Excluded Publish/Excluded Publish (MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/MQOPER_EXCLUDED_PUBLISH)

Die zusätzlichen Aktivitätsberichtsdatenparameter, die in der PCF-Gruppe *Operation* für den Operationstyp *Publish/Discarded Publish/Excluded Publish (MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/MQOPER_EXCLUDED_PUBLISH)* zurückgegeben werden (eine *Publish/Subscribe*-Nachricht wurde zugestellt, verworfen oder ausgeschlossen).

SubId

Beschreibung:	Die Subskriptionskennung.
ID:	MQBACF_SUB_ID.
Datentyp:	MQCFBS.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Immer.

SubLevel

Beschreibung:	Die Subskriptionsebene.
ID:	MQIACF_SUB_LEVEL.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Immer.

Feedback

Beschreibung:	Der Grund für das Löschen der Nachricht.
ID:	MQIACF_FEEDBACK.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Wenn die Nachricht gelöscht wurde, weil sie nicht an einen Subskribenten zugestellt wurde, oder die Nachricht nicht zugestellt wurde, weil der Subskribent ausgeschlossen wurde.

Die Veröffentlichungsoperation 'MQOPER_PUBLISH' enthält Informationen zu einer Nachricht, die an einen bestimmten Subskribenten zugestellt wird. Diese Operation beschreibt die Elemente der einseitigen Nachricht, die sich möglicherweise von der in der zugeordneten Put-Operation beschriebenen Nachricht geändert haben. Ähnlich wie bei einer Put-Operation enthält sie eine Nachrichtengruppe MQGACF_MES-

SAGE und innerhalb dieser eine MQMD-Gruppe MQGACF_MQMD. Diese MQMD-Gruppe enthält jedoch die folgenden Felder, die von einem Subskribenten überschrieben werden können: *Format, Priority, Persistence, MsgId, CorrelId, UserIdentifier, AccountingToken, ApplIdentityData*.

Die *SubId* und die *SubLevel* des Subskribenten sind in den Operationsinformationen enthalten. Sie können die *Sub-ID* mit dem PCF-Befehl MQCMD_INQUIRE_SUBSCRIBER verwenden, um alle anderen Attribute für einen Subskribenten abzurufen.

Die Operation MQOPER_DISCARDED_PUBLISH der Veröffentlichungsfunktion ist analog zu der Operation 'Discard', die verwendet wird, wenn eine Nachricht nicht in Punkt-zu-Punkt-Messaging zugestellt wird. Eine Nachricht wird nicht an einen Subskribenten zugestellt, wenn die Nachricht explizit angefordert wurde, dass sie nicht an eine lokale Destination zugestellt werden soll, und dieser Subskribent gibt ein lokales Ziel an. Eine Nachricht wird auch als nicht zugestellt betrachtet, wenn ein Problem beim Abrufen der Nachricht in die Zielwarteschlange besteht, z. B. weil die Warteschlange voll ist.

Die Informationen in einer Discarded Publish-Operation sind dieselben wie bei einer Publizierungsoperation mit dem Hinzufügen eines *Feedback* -Felds, das die Gründe angibt, warum die Nachricht nicht zugestellt wurde. Dieses Feedback-Feld enthält die Werte MQFB_* oder MQRC_*, die mit der Operation MQOPER_DISCARD gemeinsam sind. Der Grund für das Verwerfen einer Veröffentlichung im Gegensatz zu Ausschließen ist mit den Gründen für das Löschen eines Abschlusses identisch.

Die Operation 'Ausgeschlossene Publizierung' MQOPER_EXCLUDED_PUBLISH stellt Informationen zu einem Subskribenten bereit, der für die Zustellung der Nachricht in Betracht gezogen wurde, da das Thema, für das der Subskribent subskribiert, mit der zugeordneten Put-Operation übereinstimmt, die Nachricht jedoch nicht an den Subskribenten zugestellt wurde, weil andere Auswahlkriterien nicht mit der Nachricht übereinstimmen, die in das Thema gestellt wird. Wie bei einer Discarded Publish-Operation enthält das Feld *Feedback* Informationen zu der Ursache, warum diese Subskription ausgeschlossen wurde. Anders als bei der Veröffentlichung von Veröffentlichungsoperationen werden jedoch keine nachrichtenbezogenen Informationen bereitgestellt, da für diesen Subskribenten keine Nachricht generiert wurde.

Bericht 'Put/Put Reply/Put Report' (MQOPER_PUT/MQOPER_PUT_REPLY/MQOPER_PUT_REPORT)

Die zusätzlichen Aktivitätsberichtsdatenparameter, die in der PCF-Gruppe *Operation* für den Operationstyp 'Put/Put Reply/Put Report' (MQOPER_PUT/MQOPER_PUT_REPLY/MQOPER_PUT_REPORT) zurückgegeben werden, (eine Nachricht, eine Antwortnachricht oder eine Berichtsnachricht wurde in eine Warteschlange gestellt).

QName

Beschreibung:	Der Name der Warteschlange, die geöffnet wurde.
ID:	MQCA_Q_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation</i> .
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Es wird immer, abgesehen von einer Ausnahme, nicht zurückgegeben, wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

ResolvedQName

Beschreibung:	Der Name, in den die geöffnete Warteschlange aufgelöst wird.
ID:	MQCACF_RESOLVED_Q_NAME.
Datentyp:	MQCFST.

In PCF-Gruppe enthalten: *Operation.*

Maximale Länge: MQ_Q_NAME_LENGTH

Zurückgegeben: Wenn die geöffnete Warteschlange aufgelöst werden konnte. Wird nicht zurückgegeben, wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

RemoteQName

Beschreibung: Der Name der geöffneten Warteschlange, wie sie auf dem fernen Warteschlangenmanager bekannt ist.

ID: MQCA_REMOTE_Q_NAME.

Datentyp: MQCFST.

In PCF-Gruppe enthalten: *Operation.*

Maximale Länge: MQ_Q_NAME_LENGTH

Zurückgegeben: Wenn es sich bei der geöffneten Warteschlange um eine ferne Warteschlange handelt. Wird nicht zurückgegeben, wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

RemoteQMgrName

Beschreibung: Der Name des fernen Warteschlangenmanagers, auf dem die ferne Warteschlange definiert ist.

ID: MQCA_REMOTE_Q_MGR_NAME.

Datentyp: MQCFST.

In PCF-Gruppe enthalten: *Operation.*

Maximale Länge: MQ_Q_MGR_NAME_LENGTH

Zurückgegeben: Wenn es sich bei der geöffneten Warteschlange um eine ferne Warteschlange handelt. Wird nicht zurückgegeben, wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

TopicString

Beschreibung: Die vollständige Themenzeichenfolge, in die die Nachricht gestellt wird.

ID: MQCA_TOPIC_STRING.

Datentyp: MQCFST.

In PCF-Gruppe enthalten: *Operation.*

Zurückgegeben: Wenn die Put-Operation zu einem Thema gehört, das in einer Veröffentlichungsaktivität enthalten ist.

Feedback

Beschreibung: Der Grund für die Nachricht, die in die Warteschlange für dead-letter gestellt wird.

ID: MQIACF_FEEDBACK.

Datentyp: MQCFIN.

In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Wenn die Nachricht in die Warteschlange für nicht zustellbare Nachrichten gestellt wurde.

Empfangen (MQOPER_RECEIVE)

Die zusätzlichen Aktivitätsberichtsdatenparameter, die in der PCF-Gruppe *Operation* für den Operationstyp 'Empfangen (MQOPER_RECEIVE)' zurückgegeben werden (eine Nachricht wurde auf einem Kanal empfangen).

ChannelName

Beschreibung:	Der Name des Kanals, auf dem die Nachricht empfangen wurde.
ID:	MQCACH_CHANNEL_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_CHANNEL_NAME_LENGTH
Zurückgegeben:	Immer.

ChannelType

Beschreibung:	Der Typ des Kanals, auf dem die Nachricht empfangen wurde.
ID:	MQIACH_CHANNEL_TYPE.
Datentyp:	MQCFIN.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Zurückgegeben:	Immer.

RemoteQMgrName

Beschreibung:	Der Name des Warteschlangenmanagers, von dem die Nachricht empfangen wurde.
ID:	MQCA_REMOTE_Q_MGR_NAME.
Datentyp:	MQCFST.
In PCF-Gruppe enthalten:	<i>Operation.</i>
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH
Zurückgegeben:	Immer.

Senden (MQOPER_SEND)

Die zusätzlichen Nachrichtendatenparameter für Aktivitäten, die in der PCF-Gruppe *Operation* für den Operationstyp 'Senden (MQOPER_SEND)' zurückgegeben werden (eine Nachricht wurde auf einem Kanal gesendet).

ChannelName

Beschreibung:	Der Name des Kanals, an den die Nachricht gesendet wurde.
ID:	MQCACH_CHANNEL_NAME.
Datentyp:	MQCFST.

In PCF-Gruppe
enthalten: *Operation.*

Maximale Länge: MQ_CHANNEL_NAME_LENGTH.

Zurückgegeben: Immer.

ChannelType

Beschreibung: Der Typ des Kanals, an den die Nachricht gesendet wurde.

ID: MQIACH_CHANNEL_TYPE.

Datentyp: MQCFIN.

In PCF-Gruppe
enthalten: *Operation.*

Zurückgegeben: Immer.

XmitQName

Beschreibung: Die Übertragungswarteschlange, aus der die Nachricht abgerufen wurde.

ID: MQCACH_XMIT_Q_NAME.

Datentyp: MQCFST.

In PCF-Gruppe
enthalten: *Operation.*

Maximale Länge: MQ_Q_NAME_LENGTH.

Zurückgegeben: Immer.

RemoteQMgrName

Beschreibung: Der Name des fernen Warteschlangenmanagers, an den die Nachricht gesendet wurde.

ID: MQCA_REMOTE_Q_MGR_NAME.

Datentyp: MQCFST.

In PCF-Gruppe
enthalten: *Operation.*

Maximale Länge: MQ_Q_MGR_NAME_LENGTH

Zurückgegeben: Immer.

Trace-Route-Nachrichtenreferenz

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

Nachrichtenformat für Trace-Route

Trace-Route-Nachrichten sind IBM MQ-Standardnachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Die Nachrichtendaten enthalten Informationen zu den Aktivitäten, die in einer Trace-Route-Nachricht ausgeführt wurden, da sie über ein WS-Manager-Netz weitergeleitet wurde.

Trace-Route-Nachrichten enthalten die folgenden Informationen:

Ein Nachrichtendeskriptor

Eine MQMD-Struktur, wobei das Feld *Format* auf MQFMT_ADMIN oder MQFMT_EMBEDDED_PCF gesetzt ist.

Nachrichtendaten

Besteht aus:

- Ein PCF-Header (MQCFH) und Trace-Route-Nachrichtendaten, wenn *Format* auf MQFMT_ADMIN gesetzt ist, oder
- Ein eingebetteter PCF-Header (MQEPH), Trace-Route-Nachrichtendaten und zusätzliche benutzerdefinierte Nachrichtendaten, wenn *Format* auf MQFMT_EMBEDDED_PCF gesetzt ist.

Wenn Sie mit der IBM MQ-Anwendung zur Routenanzeige eine Trace-Route-Nachricht generieren, wird *Format* auf MQFMT_ADMIN gesetzt.

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

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

Tabelle 20 auf Seite 133 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 Flaggen PCF-Header (MQCFH) Strukturtyp Strukturlänge Strukturversion Befehls-ID Nachrichtenfolgennummer Steuerungsoptionen Beendigungscode Ursachencode Parameteranzahl	TraceRoute Detail Aufgezeichnete Aktivitäten Nicht aufgezeichnete Aktivitäten Diskontinuitätszähler Max. Aktivitäten Kumulieren Deliver

Trace-Route-Nachricht MQMD (Nachrichtendeskriptor)

Verwenden Sie diese Seite, um die Werte anzuzeigen, die in der MQMD-Struktur für eine Trace-Route-Nachricht enthalten sind.

StrucId

Beschreibung: Struktur-ID.
Datentyp: MQCHAR4.
Wert: MQMD_STRUC_ID.

Version

Beschreibung: Strukturversionsnummer.
Datentyp: MQLONG.
Werte: **MQMD_VERSION_1.**

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:	2.

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 138.

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:	Nachrichtenfolgennummer. Dies ist die Folgennummer der Nachricht in einer Gruppe zusammengehöriger Nachrichten.
Datentyp:	MQLONG.
Werte:	1.

Control

Beschreibung:	Steuroptionen.
Datentyp:	MQLONG.
Werte:	MQCFC_LAST.

CompCode

Beschreibung:	Beendigungscode.
Datentyp:	MQLONG.
Werte:	MQCC_OK.

Reason

Beschreibung:	Beendigungscode für Ursachencode.
Datentyp:	MQLONG.
Werte:	MQRC_NONE.

ParameterCount

Beschreibung:	Anzahl der Parameterstrukturen. Dies ist die Anzahl der Parameterstrukturen, die auf die MQCFH-Struktur folgen. Eine Gruppenstruktur (MQCFGR) und die darin enthaltenen Parameterstrukturen werden nur als eine einzige Struktur gezählt.
Datentyp:	MQLONG.
Werte:	1 oder größer.

Trace-Route-Nachrichtendaten

Verwenden Sie diese Seite, um die Parameter anzuzeigen, aus denen die *TraceRoute* -PCF-Gruppe Teil der Trace-Route-Nachrichtendaten ist.

Der Inhalt der Trace-Route-Nachrichtendaten hängt vom Parameter *Accumulate* in der *TraceRoute* -PCF-Gruppe ab. Trace-Route-Nachrichtendaten bestehen aus der *TraceRoute* -PCF-Gruppe und null oder mehr *Activity* -PCF-Gruppen. Die *TraceRoute* -PCF-Gruppe ist in diesem Abschnitt ausführlich beschrieben. Weitere Informationen zu der *Activity* -PCF-Gruppe finden Sie in den zugehörigen Informationen.

Trace-Route-Nachrichtendaten enthalten die folgenden Parameter:

TraceRoute

Beschreibung:	Gruppierte Parameter, die Attribute der Trace-Route-Nachricht angeben. Für eine Trace-Route-Nachricht können einige dieser Parameter geändert werden, um zu steuern, wie die Route verarbeitet wird.
ID:	MQGACF_TRACE_ROUTE.
Datentyp:	MQCFGR.
Enthalten in PCF-Gruppe:	Keine.

Parameter in Gruppe:	<i>Detail</i> <i>RecordedActivities</i> <i>UnrecordedActivities</i> <i>DiscontinuityCount</i> <i>MaxActivities</i> <i>Accumulate</i> <i>Forward</i> <i>Deliver</i>
----------------------	---

Detail

Beschreibung:	Die Detailstufe, die für die Aktivität aufgezeichnet wird.
ID:	MQIACF_ROUTE_DETAIL.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .
Werte:	<p>MQROUTE_DETAIL_LOW Aktivitäten, die von einer vom Benutzer geschriebenen Anwendung ausgeführt werden, werden aufgezeichnet.</p> <p>MQROUTE_DETAIL_MEDIUM Die in MQROUTE_DETAIL_LOW angegebenen Aktivitäten werden aufgezeichnet. Darüber hinaus werden Aktivitäten, die von MCAs ausgeführt werden, aufgezeichnet.</p> <p>MQROUTE_DETAIL_HIGH Die in 'MQROUTE_DETAIL_LOW' und 'MQROUTE_DETAIL_MEDIUM' angegebenen Aktivitäten werden aufgezeichnet. MCAs zeichnen keine weiteren Aktivitätsinformationen auf dieser Detaillierungsebene auf. Diese Option ist nur für benutzerdefinierte Anwendungen verfügbar, die weitere Aktivitätsinformationen aufzeichnen sollen.</p>

RecordedActivities

Beschreibung:	Die Anzahl der Aktivitäten, die von der Trace-Route-Nachricht verursacht wurden, wo Informationen aufgezeichnet wurden.
ID:	MQIACF_RECORDED_ACTIVITIES.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .

UnrecordedActivities

Beschreibung:	Die Anzahl der Aktivitäten, die von der Trace-Route-Nachricht verursacht wurden, wo keine Informationen aufgezeichnet wurden.
ID:	MQIACF_UNRECORDED_ACTIVITIES.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .

DiscontinuityCount

Beschreibung:	Die Häufigkeit, mit der eine Trace-Route-Nachricht von einem WS-Manager empfangen wurde, der keine Trace-Route-Messaging unterstützt.
ID:	MQIACF_DISCONTINUITY_COUNT.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .

MaxActivities

Beschreibung:	Die maximale Anzahl an Aktivitäten, an der die Trace-Route-Nachricht beteiligt sein kann, bevor sie nicht mehr verarbeitet wird.
ID:	MQIACF_MAX_ACTIVITIES.
Datentyp:	MQCFIN.
Enthalten in PCF-Gruppe:	<i>TraceRoute</i> .
Wert:	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: *TraceRoute*.

Wert: **MQROUTE_FORWARD_IF_SUPPORTED**

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

MQROUTE_FORWARD_ALL

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

Deliver

Beschreibung: Gibt die Aktion an, die ausgeführt werden soll, wenn die Trace-Route-Nachricht erfolgreich an der Zielwarteschlange ankommt.

ID: MQIACF_ROUTE_DELIVERY.

Datentyp: MQCFIN.

Enthalten in PCF-Gruppe: *TraceRoute*.

Wert: **MQROUTE_DELIVER_YES**

Bei der Ankunft wird die Trace-Route-Nachricht in die Zielwarteschlange gestellt. Jede Anwendung, die eine zerstörerische Verbindung in der Zielwarteschlange ausführt, kann die Trace-Route-Nachricht empfangen.

MQROUTE_DELIVER_NO

Bei der Ankunft wird die Trace-Route-Nachricht gelöscht.

Trace-Route-Antwortnachrichtenreferenz

Verwenden Sie diese Seite, um eine Übersicht über das Antwortnachrichtenformat für Trace-Routen zu erhalten. Die Trace-Route-Antwortnachrichtendaten sind ein Duplikat der Trace-Route-Nachrichtendaten aus der Trace-Route-Nachricht, für die sie generiert wurde.

Antwortnachrichtenformat für Trace-Route

Trace-Route-Antwortnachrichten sind IBM MQ-Standardnachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Die Nachrichtendaten enthalten Informationen zu den Aktivitäten, die in einer Trace-Route-Nachricht ausgeführt wurden, da sie über ein WS-Manager-Netz weitergeleitet wurde.

Tracing-Route-Antwortnachrichten enthalten die folgenden Informationen:

Ein Nachrichtendeskriptor

Eine MQMD-Struktur

Nachrichtendaten

Ein PCF-Header (MQCFH) und Trace-Route-Antwortnachrichtendaten

Die Trace-Route-Antwortnachrichtendaten bestehen aus einer oder mehreren *Activity* -PCF-Gruppen.

Wenn eine Trace-Route-Nachricht ihre Zielwarteschlange erreicht, kann eine Trace-Route-Antwortnachricht generiert werden, die eine Kopie der Aktivitätsinformationen aus der Trace-Route-Nachricht enthält. Die Antwortnachricht für die Trace-Route wird an eine Empfangswarteschlange für Antworten oder an eine Systemwarteschlange zugestellt.

Tabelle 21 auf Seite 143 zeigt die Struktur einer Antwortnachricht für die Trace-Route, einschließlich der Parameter, die nur unter bestimmten Bedingungen zurückgegeben werden.

Tabelle 21. Antwortnachrichtenformat für Trace-Route

MQMD-Struktur	PCF-Header MQCFH-Struktur	Trace-Route-Antwortnachrichtendaten
Struktur-ID Strukturversion Berichtsoptionen Nachrichtentyp Verfallszeit Feedback Encoding ID des codierten Zeichensatzes (CCSID). Nachrichtenformat Priority Permanenz Nachrichten-ID Korrelations-ID Rücksetzungszähler Empfangswarteschlange für Antworten Warteschlangenmanager für Antwortwarteschlange Benutzer-ID Abrechnung Anwendungsidentitätsdaten Anwendungstyp Anwendungsname Aufnahmezeit Ursprungsdaten der Anwendung Gruppen-ID Nachrichtenfolgennummer Offset Nachrichtenmarkierungen Ursprüngliche Länge	PCF-Header (MQCFH) Strukturtyp 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 Detail Aufgezeichnete Aktivitäten Nicht aufgezeichnete Aktivitäten Diskontinuitätszähler Max. Aktivitäten Kumulieren Deliver

Anmerkung:

1. Wird für Get- und Browse-Operationen zurückgegeben.
2. Wird für Discard-Operationen zurückgegeben.
3. Zurückgegeben für Put-, Put-Reply- und Put-Report-Operationen.
4. Zurückgegeben für Empfangsoperationen.
5. Wird für Sendeoperationen zurückgegeben.

Trace-Route-Antwortnachricht MQMD (Nachrichtendeskriptor)

Verwenden Sie diese Seite, um die Werte anzuzeigen, die in der MQMD-Struktur für eine Trace-Route-Antwortnachricht enthalten sind.

Für eine Antwortnachricht auf Trace-Route enthält die MQMD-Struktur die Parameter, die im Nachrichtendeskriptor für Aktivitätsberichte beschrieben sind. Einige der Parameterwerte in einem Antwortnach-

richtendeskriptor einer Trace-Route unterscheiden sich von denen in einem Nachrichtendeskriptor für Aktivitätsberichte wie folgt:

MsgType

Beschreibung: Typ der Nachricht.
Datentyp: MQLONG.
Wert: **MQMT_REPLY**

Feedback

Beschreibung: Rückmeldungs- oder Ursachencode.
Datentyp: MQLONG.
Wert: **MQFB_NONE**

Encoding

Beschreibung: Numerische Codierung von Nachrichtendaten.
Datentyp: MQLONG.
Wert: Kopiert aus Trace-Route-Nachrichtendeskriptor.

CodedCharSetId

Beschreibung: Zeichensatz-ID der Nachrichtendaten.
Datentyp: MQLONG.
Wert: Kopiert aus Trace-Route-Nachrichtendeskriptor.

Format

Beschreibung: Name des Formats von Nachrichtendaten
Datentyp: MQCHAR8.
Wert: **MQFMT_ADMIN**
Admin-Nachricht.

Trace-Route-Antwortnachricht MQCFH (PCF-Header)

Verwenden Sie diese Seite, um die PCF-Werte anzuzeigen, die in der MQCFH-Struktur für eine Trace-Route-Antwortnachricht enthalten sind.

Der PCF-Header (MQCFH) für eine Trace-Route-Antwortnachricht ist die gleiche wie bei einer Trace-Route-Nachricht.

Trace-Route-Antwortnachrichtendaten

Die Trace-Route-Antwortnachrichtendaten sind ein Duplikat der Trace-Route-Nachrichtendaten aus der Trace-Route-Nachricht, für die sie generiert wurde.

Die Antwortnachrichtendaten der Traceroute enthalten mindestens eine *Activity* -Gruppe. Die Parameter werden in „[Aktivitätsbericht-Nachrichtendaten](#)“ auf Seite 114 beschrieben.

Abrechnungs- und Statistiknachrichten

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

Abrechnungsnachrichten

Mit Abrechnungsnachrichten werden Informationen zu den MQI-Operationen aufgezeichnet, die von IBM MQ-Anwendungen ausgeführt werden (siehe „Abrechnungsnachrichten“ auf Seite 145).

Statistiknachrichten

Mit Statistiknachrichten werden Informationen zu den Aktivitäten aufgezeichnet, die in einem IBM MQ-System auftreten (siehe „Statistiknachrichten“ auf Seite 149). Einige in Statistiknachrichten aufgezeichnete Aktivitäten beziehen sich auf interne Warteschlangenmanageroperationen.

 Die hier beschriebenen Abrechnungsnachrichten und Statistiknachrichten sind nicht unter IBM MQ for z/OS verfügbar, aber über SMF (System Management Facility, Systemverwaltungsfunktion) ist eine ähnliche Funktion verfügbar.

Abrechnungs- und Statistiknachrichten werden an eine von zwei Systemwarteschlangen zugestellt. Benutzeranwendungen können die Nachrichten aus diesen Systemwarteschlangen abrufen und die aufgezeichneten Informationen zu verschiedenen Zwecken verwenden:

- Konto für Anwendungsressourcenverwendung.
- Die Anwendungsaktivität erfassen.
- Kapazitätsplanung.
- Erkennen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Unterstützung bei der Bestimmung der Ursachen von Problemen in Ihrem Warteschlangenmanagernetzwerk.
- Verbessern Sie die Effizienz Ihres Warteschlangenmanagernetzes.
- Machen Sie sich mit der Ausführung Ihres WS-Manager-Netzes vertraut.
- Stellen Sie sicher, dass Ihr Warteschlangenmanagernetzwerk ordnungsgemäß ausgeführt wird.

Zugehörige Konzepte

„Using System Management Facility“ auf Seite 335

You can use SMF to collect statistics and accounting information. To use SMF, certain parameters must be set in z/OS and in IBM MQ.

Abrechnungsnachrichten

Abrechnungsnachrichten zeichnen Informationen zu den MQI-Operationen auf, die von IBM MQ-Anwendungen ausgeführt werden. Eine Abrechnungsnachricht ist eine PCF-Nachricht, die eine Reihe von PCF-Strukturen enthält.

Wenn eine Anwendung die Verbindung zu einem WS-Manager trennt, wird eine Abrechnungsnachricht generiert und an die Systemabrechnungswarteschlange (SYSTEM.ADMIN.ACCOUNTING.QUEUE) übergeben. Für IBM MQ-Anwendungen mit langer Laufzeit werden temporäre Abrechnungsnachrichten folgendermaßen generiert:

- Wenn die Zeit seit dem Zeitpunkt der Verbindungseinbindung überschritten wurde, überschreitet das konfigurierte Intervall.
- Wenn die Zeit seit der letzten Zwischenabrechnungsnachricht das konfigurierte Intervall überschreitet.

Abrechnungsnachrichten befinden sich in den folgenden Kategorien:

MQI-Abrechnungsnachrichten

MQI-Abrechnungsnachrichten enthalten Informationen, die sich auf die Anzahl der MQI-Aufrufe beziehen, die über eine Verbindung zu einem Warteschlangenmanager hergestellt wurden.

Nachrichten zur Warteschlangenabrechnung

Warteschlangenabrechnungsnachrichten enthalten Informationen, die sich auf die Anzahl der MQI-Aufrufe beziehen, die unter Verwendung von Verbindungen zu einem Warteschlangenmanager (gruppiert nach Warteschlange) hergestellt wurden.

Jede Warteschlangenabrechnungsnachricht kann bis zu 100 Datensätze enthalten, wobei jeder Datensatz sich auf eine Aktivität bezieht, die von der Anwendung in Bezug auf eine bestimmte Warteschlange ausgeführt wird.

Abrechnungsnachrichten werden nur für lokale Warteschlangen aufgezeichnet. Wenn eine Anwendung einen MQI-Aufruf für eine Aliaswarteschlange aufruft, werden die Abrechnungsdaten für die Basiswarteschlange erfasst und die Abrechnungsdaten für eine ferne Warteschlange werden für die Übertragungswarteschlange aufgezeichnet.

Anmerkung: Da sich die Informationen auf MQI-Operationen beziehen, die von IBM MQ -Anwendungen ausgeführt wurden, enthalten die Informationen keine Operationen, die sich auf eine Datenstromwarteschlange beziehen, es sei denn, die Anwendung befindet sich direkt in dieser Warteschlange.

Zugehörige Verweise

„MQI-Abrechnungsnachrichtendaten“ auf Seite 165

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

„Nachrichtendaten der Warteschlangenabrechnung“ auf Seite 176

Auf dieser Seite können Sie die Struktur einer Warteschlangenabrechnungsnachricht anzeigen.

Abrechnungsnachrichtenformat

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

Nachrichtendeskriptor

- Eine Abrechnungsnachricht MQMD (Nachrichtendeskriptor)

Abrechnungsnachrichtendaten

- Eine Abrechnungsnachricht MQCFH (PCF-Header)
- Accounting-Nachrichtendaten, die immer zurückgegeben werden
- Abrechnungs-Nachrichtendaten, die zurückgegeben werden, falls

Die Abrechnungsnachricht MQCFH (PCF-Header) enthält Informationen zu der Anwendung und das Intervall, in dem die Abrechnungsdaten aufgezeichnet wurden.

Abrechnungsnachrichtendaten enthalten PCF-Parameter, in denen die Abrechnungsdaten gespeichert werden. Der Inhalt der Abrechnungsnachrichten hängt wie folgt von der Nachrichtenkategorie ab:

MQI-Abrechnungsnachricht

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

Warteschlangenabrechnungsnachricht

Die Daten der Warteschlangenabrechnungsnachricht bestehen aus einer Reihe von PCF-Parametern und im Bereich von 1 bis 100 *QAccountingData* -PCF-Gruppen.

Es gibt eine *QAccountingData* -PCF-Gruppe für jede Warteschlange, für die Abrechnungsdaten erfasst wurden. Wenn eine Anwendung auf mehr als 100 Warteschlangen zugreift, werden mehrere Abrechnungsnachrichten generiert. Jede Nachricht hat die *SeqNumber* im MQCFH (PCF-Header) entsprechend aktualisiert, und die letzte Nachricht in der Sequenz hat den Parameter *Control* in der MQCFH-Datei als MQCFC_LAST angegeben.

Erfassung von Abrechnungsdaten

Verwenden Sie die Attribute der Warteschlange und des Warteschlangenmanagers, um die Erfassung von Abrechnungsdaten zu steuern. Sie können MQCONNX-Optionen auch verwenden, um die Erfassung auf Verbindungsebene zu steuern.

Erfassung von MQI-Abrechnungsdaten steuern

Verwenden Sie das WS-Manager-Attribut ACCTMQI, um die Erfassung von MQI-Abrechnungsdaten zu steuern.

Wenn Sie den Wert dieses Attributs ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter ACCTMQI an. Abrechnungsnachrichten werden nur für Verbindungen generiert, die nach der Aktivierung der Abrechnung beginnen. Der Parameter **ACCTMQI** kann die folgenden Werte haben:

ON

Die MQI-Abrechnungsdaten werden für jede Verbindung zum WS-Manager erfasst.

OFF

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

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

```
ALTER QMGR ACCTMQI(ON)
```

Informationen zur Warteschlangenabrechnung

Verwenden Sie das Warteschlangenattribut ACCTQ und das WS-Manager-Attribut ACCTQ, um die Erfassung von Warteschlangenabrechnungsdaten zu steuern.

Wenn Sie den Wert des Warteschlangenattributs ändern möchten, verwenden Sie den MQSC-Befehl ALTER QLOCAL und geben Sie den Parameter ACCTQ an. Abrechnungsnachrichten werden nur für Verbindungen generiert, die nach der Aktivierung der Abrechnung beginnen. Beachten Sie, dass Änderungen an diesem Wert nur für Verbindungen zum WS-Manager wirksam sind, die nach der Änderung des Attributs auftreten.

Das Warteschlangenattribut ACCTQ kann die folgenden Werte haben:

ON

Die Warteschlangenabrechnungsdaten für diese Warteschlange werden für jede Verbindung zu dem Warteschlangenmanager erfasst, der die Warteschlange öffnet.

OFF

Die Warteschlangenabrechnungsdaten für diese Warteschlange werden nicht erfasst.

QMGR

Die Erfassung der Warteschlangenabrechnungsdaten für diese Warteschlange wird entsprechend dem Wert des Warteschlangenmanagerattributs ACCTQ gesteuert. Dies ist der Standardwert.

Wenn Sie den Wert des Warteschlangenmanagerattributs ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter ACCTQ an. Das WS-Manager-Attribut ACCTQ kann die folgenden Werte haben:

ON

Die Warteschlangenabrechnungsdaten werden für Warteschlangen erfasst, die das Warteschlangenattribut ACCTQ als QMGR festgelegt haben.

OFF

Es werden keine Warteschlangenabrechnungsdaten für Warteschlangen erfasst, für die das Warteschlangenattribut ACCTQ als QMGR festgelegt ist. Dies ist der Standardwert.

KEINE

Die Erfassung von Warteschlangenabrechnungsdaten ist für alle Warteschlangen inaktiviert, unabhängig vom Warteschlangenattribut ACCTQ.

Wenn das WS-Managerattribut ACCTQ auf NONE gesetzt ist, ist die Erfassung von Warteschlangenabrechnungsinformationen für alle Warteschlangen inaktiviert, unabhängig vom Warteschlangenattribut ACCTQ.

Wenn Sie beispielsweise die Erfassung von Abrechnungsdaten für die Warteschlange Q1 aktivieren möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QLOCAL(Q1) ACCTQ(ON)
```

Verwenden Sie den folgenden MQSC-Befehl, um die Erfassung von Abrechnungsdaten für alle Warteschlangen zu aktivieren, in denen das Warteschlangenattribut ACCTQ als QMGR angegeben ist:

```
ALTER QMGR ACCTQ(ON)
```

MQCONN-Optionen für die Steuerung der Erfassung von Abrechnungsdaten

Verwenden Sie den Parameter **ConnectOpts** im Aufruf MQCONN, um die Erfassung von MQI- und Warteschlangenabrechnungsdaten auf Verbindungsebene zu ändern, indem die effektiven Werte der WS-Manager-Attribute ACCTMQI und ACCTQ überschrieben werden.

Der Parameter **ConnectOpts** kann die folgenden Werte haben:

MQCNO_ACCOUNTING_MQI_ENABLED

Wenn der Wert des WS-Managerattributs ACCTMQI als OFF angegeben ist, wird die MQI-Abrechnung für diese Verbindung aktiviert. Dies entspricht dem WS-Managerattribut ACCTMQI, das als ON angegeben wird.

Wenn der Wert des WS-Managerattributs ACCTMQI nicht als OFF angegeben ist, hat dieses Attribut keine Auswirkungen.

MQCNO_ACCOUNTING_MQI_DISABLED

Wenn der Wert des WS-Managerattributs ACCTMQI als ON angegeben wird, ist die MQI-Abrechnung für diese Verbindung inaktiviert. Dies entspricht dem WS-Managerattribut ACCTMQI, das als OFF angegeben wird.

Wenn der Wert des WS-Managerattributs ACCTMQI nicht als ON angegeben ist, hat dieses Attribut keine Auswirkungen.

MQCNO_ACCOUNTING_Q_ENABLED

Wenn der Wert des Warteschlangenmanagerattributs ACCTQ als OFF angegeben ist, wird die Warteschlangenabrechnung für diese Verbindung aktiviert. Alle Warteschlangen mit ACCTQ, die als QMGR angegeben sind, werden für die Warteschlangenabrechnung aktiviert. Dies entspricht dem WS-Managerattribut ACCTQ, das als ON angegeben wird.

Wenn der Wert des WS-Managerattributs ACCTQ nicht als OFF angegeben ist, hat dieses Attribut keine Auswirkungen.

MQCNO_ACCOUNTING_Q_DISABLED

Wenn der Wert des Warteschlangenmanagerattributs ACCTQ als ON angegeben ist, ist die Warteschlangenabrechnung für diese Verbindung inaktiviert. Dies entspricht dem WS-Managerattribut ACCTQ, das als OFF angegeben wird.

Wenn der Wert des Warteschlangenmanagerattributs ACCTQ nicht als ON angegeben ist, hat dieses Attribut keine Auswirkungen.

Diese Überschreibungen sind standardmäßig inaktiviert. Wenn Sie sie aktivieren möchten, setzen Sie das WS-Managerattribut ACCTCONO auf ENABLED. Verwenden Sie den folgenden MQSC-Befehl, um Abrechnungsüberschreibungen für einzelne Verbindungen zu aktivieren:

```
ALTER QMGR ACCTCONO(ENABLED)
```

Generierung von Accounting-Nachrichten bei Verbindung zu Anwendungen

Abrechnungsnachrichten werden generiert, wenn eine Anwendung die Verbindung zum WS-Manager trennt. Temporäre Abrechnungsnachrichten werden auch für lang aktive IBM MQ-Anwendungen geschrieben.

Abrechnungsnachrichten werden auf eine der folgenden Arten generiert, wenn eine Anwendung die Verbindung trennt:

- Die Anwendung gibt einen MQDISC-Aufruf aus.
- Der WS-Manager erkennt, dass die Anwendung beendet wurde.

Temporäre Abrechnungsnachrichten werden für lange aktive IBM MQ-Anwendungen geschrieben, wenn das Intervall seit dem Zeitpunkt der Verbindungsherstellung oder seit der letzten temporären Abrechnungsnachricht, die geschrieben wurde, das konfigurierte Intervall überschreitet. Das WS-Manager-Attribut (ACCTINT) gibt die Zeit in Sekunden an, nach der temporäre Abrechnungsnachrichten automatisch geschrieben werden können. Abrechnungsnachrichten werden nur generiert, wenn die Anwendung mit dem Warteschlangenmanager interagiert, so dass Anwendungen, die lange Zeiträume ohne Ausführung von MQI-Anforderungen mit dem Warteschlangenmanager verbunden bleiben, keine Abrechnungsnachrichten generieren, bis die Ausführung der ersten MQI-Anforderung nach Abschluss des Abrechnungsintervalls erfolgt ist.

Das Standardabrechnungsintervall beträgt 1800 Sekunden (30 Minuten). Wenn Sie z. B. das Accounting-Intervall auf 900 Sekunden (15 Minuten) ändern möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QMGR ACCTINT(900)
```

Statistiknachrichten

Informationen zu den Datensätzen für Statistiknachrichten über die Aktivitäten in einem IBM MQ-System. Bei einer Statistiknachricht handelt es sich um eine PCF-Nachricht, die eine Reihe von PCF-Strukturen enthält.

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

Statistiknachrichten befinden sich in den folgenden Kategorien:

MQI-Statistiknachrichten

MQI-Statistiknachrichten enthalten Informationen, die sich auf die Anzahl der MQI-Aufrufe während eines konfigurierten Intervalls beziehen. Die Informationen können z. B. die Anzahl der MQI-Aufrufe enthalten, die von einem Warteschlangenmanager ausgegeben werden.

Warteschlangenstatistiknachrichten

Warteschlangenstatistiknachrichten enthalten Informationen, die sich auf die Aktivität einer Warteschlange während eines konfigurierten Intervalls beziehen. Zu den Informationen gehören die Anzahl der Nachrichten, die in die Warteschlange gestellt und aus der Warteschlange abgerufen wurden, sowie die Gesamtzahl der Byte, die von einer Warteschlange verarbeitet wurden.

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

Statistiknachrichten werden nur für lokale Warteschlangen aufgezeichnet. Wenn eine Anwendung einen MQI-Aufruf für eine Aliaswarteschlange aufruft, werden die Statistikdaten für die Basiswarteschlange aufgezeichnet, und für eine ferne Warteschlange werden die Statistikdaten für die Übertragungswarteschlange aufgezeichnet.

Kanalstatistiknachrichten

Kanalstatistiknachrichten enthalten Informationen, die sich auf die Aktivität eines Kanals während eines konfigurierten Intervalls beziehen. Beispielsweise kann es sich bei den Informationen um die Anzahl der vom Kanal übertragenen Nachrichten oder um die Anzahl der Byte handeln, die vom Kanal übertragen wurden.

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

Anmerkung: Da sich die Informationen auf Aktivitäten beziehen, die in einem IBM MQ -System auftreten, umfassen die Informationen Operationen, die sich auf Streaming-Warteschlangen beziehen.

Zugehörige Verweise

[„Steuerung der MQI-Statistikerfassung“ auf Seite 150](#)

Verwenden Sie das WS-Manager-Attribut STATMQI, um die Erfassung von MQI-Statistikinformationen zu steuern.

[„Erfassung von Warteschlangenstatistikdaten steuern“ auf Seite 151](#)

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

„Erfassung von Kanalstatistikdaten steuern“ auf Seite 152

Verwenden Sie das Kanalattribut STATCHL, um die Erfassung von Kanalstatistikdaten zu steuern. Sie können auch WS-Manager-Attribute festlegen, um die Informationssammlung zu steuern.

Format der 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)
- Statistikenachrichtendaten, die immer zurückgegeben werden
- Statistikenachrichtendaten, die zurückgegeben werden, falls verfügbar

Die Statistikenachricht 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 Statistikenachrichten hängt wie folgt von der Nachrichtenkategorie ab:

MQI-Statistikenachricht

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

Warteschlangenstatistikenachricht

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

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 Statistikenachrichten 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.

Kanalstatistikenachricht

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 Statistikenachrichten generiert. Jede Nachricht hat die *SeqNumber* im MQCFH (PCF-Header) entsprechend aktualisiert, und die letzte Nachricht in der Sequenz hat den Parameter *Control* in der MQCFH-Datei als MQCFC_LAST angegeben.

Erfassung statistischer Daten

Verwenden Sie die Attribute der Warteschlange, des Warteschlangenmanagers und des Kanals, um die Erfassung statistischer Daten zu steuern.

Steuerung der MQI-Statistikerfassung

Verwenden Sie das WS-Manager-Attribut STATMQI, um die Erfassung von MQI-Statistikinformationen zu steuern.

Wenn Sie den Wert dieses Attributs ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter **STATMQI** an. Statistikenachrichten 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)
```

Erfassung von Warteschlangenstatistikdaten steuern

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

Sie können die Erfassung von Warteschlangenstatistikdaten für einzelne Warteschlangen oder für mehrere Warteschlangen aktivieren oder inaktivieren. Wenn Sie einzelne Warteschlangen steuern möchten, legen Sie das Warteschlangenattribut STATQ fest. Sie aktivieren oder inaktivieren die Erfassung von Warteschlangenstatistikdaten auf WS-Managerebene unter Verwendung des Warteschlangenmanagerattributs STATQ. Für alle Warteschlangen, für die das Warteschlangenattribut STATQ mit dem Wert QMGR angegeben wurde, wird die Erfassung von Warteschlangenstatistikdaten auf WS-Managerebene gesteuert.

Warteschlangenstatistikdaten werden nur für Operationen mit IBM MQ MQI-Objektkennungen erhöht, die nach dem Aktivieren der Statistikerfassung geöffnet wurden.

Warteschlangenstatistiknachrichten werden nur für Warteschlangen generiert, für die die Statistikdaten im vorherigen Zeitraum erfasst wurden.

Dieselbe Warteschlange kann über mehrere Operationen zum Einlegen von Objekten und zum Abrufen von Operationen über mehrere Objektkennungen verfügen. Einige Objektkennungen wurden möglicherweise geöffnet, bevor die Statistikerfassung aktiviert wurde, andere jedoch später geöffnet wurden. Aus diesem Grund ist es möglich, dass die Warteschlangenstatistik die Aktivität einiger Operationen zum Einlegen und Abrufen von Operationen erfasst und nicht alle.

Um sicherzustellen, dass die Warteschlangenstatistik die Aktivität aller Anwendungen erfasst, müssen Sie neue Objektkennungen in der Warteschlange oder in den Warteschlangen schließen und erneut öffnen, die Sie überwachen. Der beste Weg, um dies zu erreichen, ist es, alle Anwendungen nach der Aktivierung der Statistikerfassung zu beenden und erneut zu starten.

Wenn Sie den Wert des Warteschlangenattributs STATQ ändern möchten, verwenden Sie den MQSC-Befehl ALTER QLOCAL und geben Sie den Parameter STATQ an. Das Warteschlangenattribut STATQ kann die folgenden Werte haben:

ON

Warteschlangenstatistikdaten werden für jede Verbindung zu dem Warteschlangenmanager erfasst, der die Warteschlange öffnet.

OFF

Warteschlangenstatistikdaten für diese Warteschlange werden nicht erfasst.

QMGR

Die Erfassung der Warteschlangenstatistikdaten für diese Warteschlange wird entsprechend dem Wert des Warteschlangenmanagerattributs STATQ gesteuert. Dies ist der Standardwert.

Wenn Sie den Wert des Warteschlangenmanagerattributs STATQ ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter STATQ an. Das WS-Manager-Attribut STATQ kann die folgenden Werte haben:

ON

Warteschlangenstatistikdaten werden für Warteschlangen erfasst, für die das Warteschlangenattribut STATQ als QMGR festgelegt ist.

OFF

Warteschlangenstatistikdaten werden nicht für Warteschlangen erfasst, für die das Warteschlangenattribut STATQ als QMGR festgelegt ist. Dies ist der Standardwert.

KEINE

Die Erfassung von Warteschlangenstatistikdaten ist für alle Warteschlangen inaktiviert, unabhängig vom Warteschlangenattribut STATQ.

Wenn das Warteschlangenmanagerattribut STATQ auf NONE gesetzt ist, ist die Erfassung von Warteschlangenstatistikdaten für alle Warteschlangen inaktiviert, unabhängig vom Warteschlangenattribut STATQ.

Wenn Sie beispielsweise die Erfassung statistischer Daten für die Warteschlange Q1 aktivieren möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QLOCAL(Q1) STATQ(ON)
```

Verwenden Sie den folgenden MQSC-Befehl, um die Erfassung statistischer Daten für alle Warteschlangen zu aktivieren, die das Warteschlangenattribut STATQ als QMGR angeben:

```
ALTER QMGR STATQ(ON)
```

Erfassung von Kanalstatistikdaten steuern

Verwenden Sie das Kanalattribut STATCHL, um die Erfassung von Kanalstatistikdaten zu steuern. Sie können auch WS-Manager-Attribute festlegen, um die Informationssammlung zu steuern.

Sie können die Erfassung von Kanalstatistikdaten für einzelne Kanäle oder für mehrere Kanäle aktivieren oder inaktivieren. Um einzelne Kanäle zu steuern, müssen Sie das Kanalattribut STATCHL festlegen, um die Erfassung der Kanalstatistikdaten zu aktivieren oder zu inaktivieren. Um viele Kanäle gemeinsam zu steuern, aktivieren oder inaktivieren Sie die Erfassung von Kanalstatistikdaten auf WS-Managerebene mit dem WS-Managerattribut STATCHL. Für alle Kanäle, die über das Kanalattribut STATCHL verfügen, das mit dem Wert QMGR angegeben wurde, wird die Erfassung der Kanalstatistikdaten auf WS-Managerebene gesteuert.

Automatisch definierte Clustersenderkanäle sind keine IBM MQ-Objekte und weisen daher keine Attribute auf dieselbe Weise wie Kanalobjekte auf. Wenn Sie automatisch definierte Clustersenderkanäle steuern möchten, verwenden Sie das WS-Managerattribut STATACLS. Dieses Attribut bestimmt, ob automatisch definierte Clustersenderkanäle innerhalb eines Warteschlangenmanagers für die Erfassung von Kanalstatistikdaten aktiviert oder inaktiviert sind.

Sie können die Erfassung von Kanalstatistikdaten auf eine der drei folgenden Überwachungsstufen festlegen: niedrig, mittel oder hoch. Sie können die Überwachungsstufe auf Objektebene oder auf WS-Managerebene festlegen. Die Auswahl der zu verwendenden Ebene hängt von Ihrem System ab. Die Erfassung statistischer Daten erfordert möglicherweise einige Anweisungen, die relativ teuer sind, um die Auswirkungen der Erfassung von Kanalstatistikdaten zu reduzieren, die Optionen für mittlere und niedrige Überwachungsdaten eine Stichprobe der Daten in regelmäßigen Intervallen zu messen und nicht die ganze Zeit zu erfassen. In [Tabelle 22 auf Seite 152](#) sind die Stufen zusammengefasst, die mit der 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.

Tabelle 22. Detailstufe der Erfassung von Kanalstatistikdaten (Forts.)

Ebene	Beschreibung	Verwendung
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.



z/OS Anz/OS Systeme: Durch die Aktivierung dieses Parameters wird einfach die Erfassung statistischer Daten aktiviert, unabhängig von dem von Ihnen ausgewählten Wert. Die Angabe von LOW, MEDIUM oder HIGH hat keine Auswirkung auf die Ergebnisse. Dieser Parameter muss aktiviert sein, damit Datensätze zur Kanalabrechnung erfasst werden können.

Das WS-Managerattribut STATCHL kann die folgenden Werte haben:

LOW

Kanalstatistikdaten werden mit einem niedrigen Detaillierungsgrad für alle Kanäle erfasst, für die das Kanalattribut STATCHL als QMGR festgelegt ist.

MITTEL

Kanalstatistikdaten werden mit einer mittleren Detaillierungsebene für alle Kanäle erfasst, für die das Kanalattribut STATCHL als QMGR festgelegt ist.

HIGH

Die Kanalstatistikdaten werden mit einem hohen Detaillierungsgrad für alle Kanäle erfasst, für die das Kanalattribut STATCHL als QMGR festgelegt ist.

OFF

Die Kanalstatistikdaten werden nicht für alle Kanäle erfasst, für die das Kanalattribut STATCHL als QMGR festgelegt ist.

Dies ist der Standardwert.

KEINE

Die Erfassung der Kanalstatistikdaten ist für alle Kanäle inaktiviert, unabhängig vom Kanalattribut STATCHL.

z/OS Anz/OS Systeme: Durch die Aktivierung dieses Parameters wird einfach die Erfassung statistischer Daten aktiviert, unabhängig von dem von Ihnen ausgewählten Wert. Die Angabe von LOW, MEDIUM oder HIGH hat keine Auswirkung auf die Ergebnisse. Dieser Parameter muss aktiviert sein, damit Datensätze zur Kanalabrechnung erfasst werden können.

Das WS-Managerattribut STATACLS kann die folgenden Werte haben:

LOW

Statistikinformationen werden mit einem niedrigen Detaillierungsgrad für automatisch definierte Clustersenderkanäle erfasst.

MITTEL

Statistikinformationen werden mit einer mittleren Detaillierungsebene für automatisch definierte Clustersenderkanäle erfasst.

HIGH

Statistikinformationen werden mit einem hohen Detaillierungsgrad für automatisch definierte Clustersenderkanäle erfasst.

OFF

Statistikinformationen werden für automatisch definierte Clustersenderkanäle nicht erfasst.

QMGR

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

Dies ist der Standardwert.

z/OS Anz/OS Systeme: Durch die Aktivierung dieses Parameters wird einfach die Erfassung statistischer Daten aktiviert, unabhängig von dem von Ihnen ausgewählten Wert. Die Angabe von LOW, MEDIUM oder HIGH hat keine Auswirkung auf die Ergebnisse. Dieser Parameter muss aktiviert sein, damit Datensätze zur Kanalabrechnung erfasst werden können.

Wenn Sie beispielsweise die Erfassung statistischer Daten mit einer mittleren Detaillierungsebene für den Senderkanal QM1 . TO . QM2 aktivieren möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER CHANNEL (QM1 . TO . 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). Wenn Sie das Statistikintervall ändern möchten, verwenden Sie den MQSC-Befehl ALTER QMGR und geben Sie den Parameter **STATINT** an. Wenn Sie zum Beispiel das Statistikintervall auf 900 Sekunden (15 Minuten) ändern möchten, verwenden Sie den folgenden MQSC-Befehl:

```
ALTER QMGR STATINT (900)
```

Verwenden Sie den MQSC-Befehl `RESET QMGR TYPE(STATISTICS)`, um die derzeit erfassten Statistikdaten in die Statistikwarteschlange zu schreiben, bevor das Intervall für die Statistikerfassung abgelaufen ist. Wenn Sie diesen Befehl absetzen, werden die erfassten Statistikdaten in die Statistikwarteschlange geschrieben und ein neues Datenerfassungsintervall für die Statistik gestartet.

Abrechnungs- und Statistikdaten anzeigen

Um die in Abrechnungs- und Statistiknachrichten aufgezeichneten Informationen zu verwenden, führen Sie eine Anwendung wie das Beispielprogramm **amqsmn** aus, um die aufgezeichneten Informationen in ein geeignetes Format umzuwandeln.

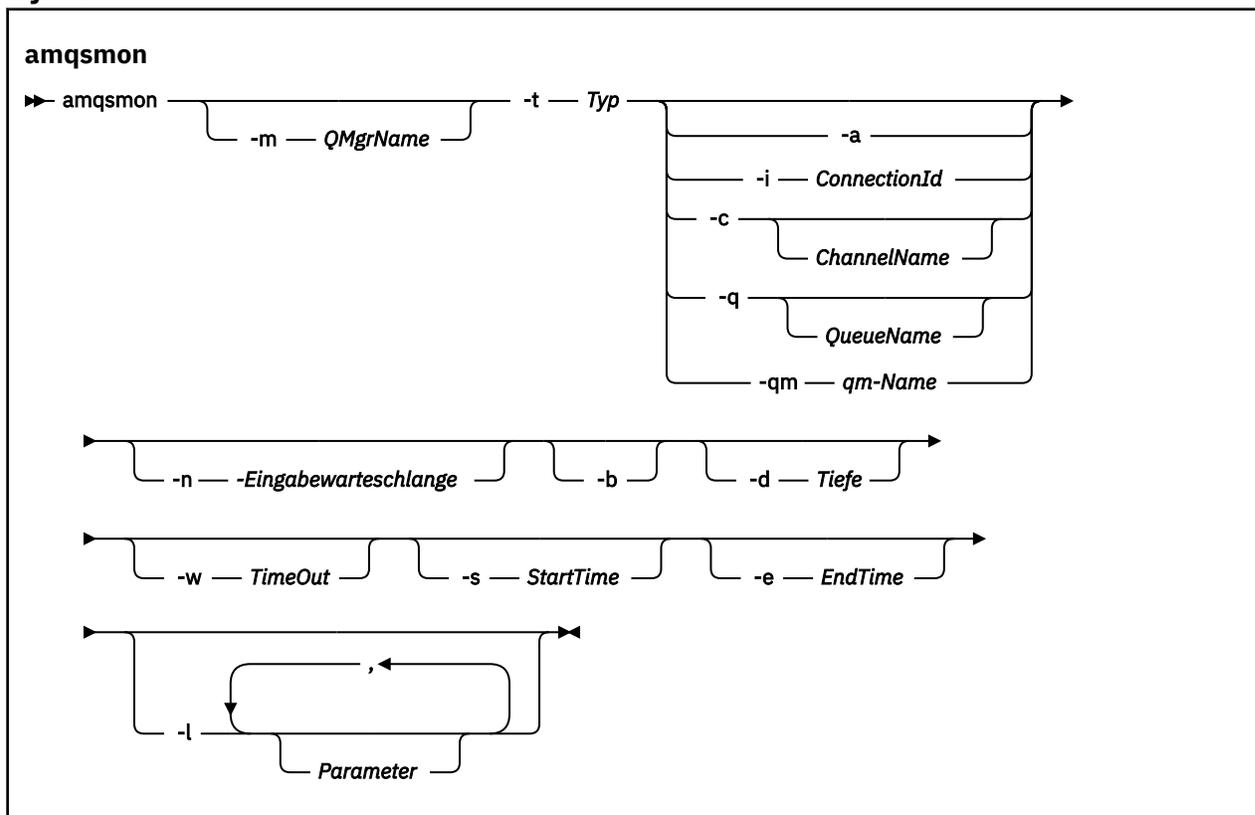
Abrechnungs- und Statistiknachrichten werden in die Systemabrechnungs- und Statistikwarteschlangen geschrieben. **amqsmn** ist ein mit IBM MQ bereitgestelltes Beispielprogramm, in dem Nachrichten aus Abrechnungs- und Statistikwarteschlangen verarbeitet und die Informationen in einem lesbaren Format in der Anzeige angezeigt werden.

Da **amqsmn** ein Beispielprogramm ist, können Sie den bereitgestellten Quellcode als Vorlage zum Schreiben Ihrer eigenen Anwendung verwenden, um Abrechnungs- oder Statistiknachrichten 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 Statistiknachrichten enthaltenen Informationen in einem lesbaren Format anzuzeigen. Das Programm **amqsmn** liest Abrechnungsnachrichten aus der Abrechnungswarteschlange `SYSTEM.ADMIN.ACCOUNTING.QUEUE.` und liest Statistiknachrichten aus der Statistikwarteschlange, `SYSTEM.ADMIN.STATISTICS.QUEUE.`

Syntax



Erforderliche Parameter

-t Type

Der Typ der Nachrichten, die verarbeitet werden sollen. Geben Sie *Type* als einen der folgenden Werte an:

Buchhaltung

Abrechnungssätze werden verarbeitet. Nachrichten werden aus der Systemwarteschlange SYSTEM.ADMIN.ACCOUNTING.QUEUE , es sei denn, Sie haben mit dem Parameter -n eine bestimmte Warteschlange ausgewählt, aus der gelesen werden soll.

statistics

Statistikdatensätze werden verarbeitet. Nachrichten werden aus der Systemwarteschlange SYSTEM.ADMIN.STATISTICS.QUEUE **V 9.4.0** Es sei denn, Sie haben mit dem Parameter -n eine bestimmte Warteschlange ausgewählt, aus der gelesen werden soll.

Optionale Parameter

-m *QMgrName*

Der Name des Warteschlangenmanagers, aus dem Abrechnungs-oder Statistiknachrichten verarbeitet werden sollen.

Wenn Sie diesen Parameter nicht angeben, wird der Standardwarteschlangenmanager verwendet.

-a

Prozessnachrichten, die nur MQI-Datensätze enthalten.

Nur MQI-Datensätze anzeigen. Nachrichten, die keine MQI-Datensätze enthalten, bleiben immer in der Warteschlange, aus der sie gelesen wurden.

-q *QueueName*

QueueName ist ein optionaler Parameter.

Wenn *QueueName* nicht angegeben wird:

Zeigt nur die Warteschlangenabrechnungs-und Warteschlangenstatistikdatensätze an

Wenn *QueueName* angegeben wird:

Zeigt die Warteschlangenabrechnungs-und Warteschlangenstatistikdatensätze für die Warteschlange an, die nur von *QueueName* angegeben wurde

Wenn -b nicht angegeben wird, werden die Abrechnungs-und Statistiknachrichten, aus denen die Datensätze stammen, gelöscht. Da Abrechnungs-und Statistiknachrichten auch Datensätze aus anderen Warteschlangen enthalten können, wenn -b nicht angegeben ist, können nicht gesehener Datensätze gelöscht werden.

-c *ChannelName*

ChannelName ist ein optionaler Parameter.

Wenn *ChannelName* nicht angegeben wird:

Zeigt nur Kanalstatistikdatensätze an.

Wenn *ChannelName* angegeben wird:

Zeigt Kanalstatistikdatensätze für den Kanal an, der nur von *ChannelName* angegeben wird.

Wenn -b nicht angegeben wird, werden die Statistiknachrichten, aus denen die Datensätze stammen, gelöscht. Da Statistiknachrichten auch Datensätze aus anderen Kanälen enthalten können, wenn -b nicht angegeben wird, können nicht gesehener Datensätze gelöscht werden.

Dieser Parameter ist nur verfügbar, wenn die Statistiknachrichten angezeigt werden (-t statistics).

V 9.4.0

-qm *qm-Name*

Dieser Parameter ist optional, aber wenn Sie ihn angeben, müssen Sie einen Warteschlangenmanagernamen angeben.

Gibt an, dass nur die vom angegebenen Warteschlangenmanager erzeugten Abrechnungs-oder Statistiknachrichten aus der Warteschlange gelesen werden sollen. Wird normalerweise nur verwendet, wenn Nachrichten von mehreren Warteschlangenmanagern an einen zentralen Warteschlangenmanager weitergeleitet wurden.

-i ConnectionId

Zeigt Datensätze an, die sich auf die Verbindungskennung beziehen, die nur von *ConnectionId* angegeben wurde.

Dieser Parameter ist nur verfügbar, wenn nur Abrechnungsnachrichten angezeigt werden (*-t accounting*).

Wenn *-b* nicht angegeben wird, werden die Statistiknachrichten, aus denen die Datensätze stammen, gelöscht. Da Statistiknachrichten auch Datensätze aus anderen Kanälen enthalten können, wenn *-b* nicht angegeben wird, können nicht gesehener Datensätze gelöscht werden.

V 9.4.0 -n Eingabewarteschlange

Die Warteschlange, aus der Abrechnungs-oder Statistiknachrichten gelesen werden. Wenn diese Option nicht angegeben ist, wird die Standardwarteschlange für den ausgewählten *-t < Typ >* verwendet.

-b

Nachrichten durchsuchen.

Nachrichten werden nicht destruktiv abgerufen.

-d Depth

Die maximale Anzahl der Nachrichten, die verarbeitet werden können.

Wenn Sie diesen Parameter nicht angeben, kann eine unbegrenzte Anzahl von Nachrichten verarbeitet werden.

-w TimeOut

Die maximale Zeit in Sekunden, die auf die Verfügbarkeit einer Nachricht gewartet wird.

Wenn Sie diesen Parameter nicht angeben, wird *amqsmon* beendet, wenn keine weiteren Nachrichten mehr verarbeitet werden.

-s StartTime

Prozessnachrichten werden nur nach dem angegebenen *StartTime* gesendet.

StartTime wird im Format *yyyy-mm-dd hh.mm.ss* angegeben. Wenn ein Datum ohne Uhrzeit angegeben wird, wird die Zeit standardmäßig *00.00.00* an dem angegebenen Datum angegeben. Die Zeiten sind in GMT.

Informationen zur Angabe dieses Parameters finden Sie in [Anmerkung 1](#).

-e EndTime

Verarbeiten von Nachrichten, die vor dem angegebenen *EndTime* gestellt werden.

Der *EndTime* wird im Format *yyyy-mm-dd hh.mm.ss* angegeben. Wenn ein Datum ohne Uhrzeit angegeben wird, wird die Zeit standardmäßig *23.59.59* an dem angegebenen Datum angegeben. Die Zeiten sind in GMT.

Informationen zur Angabe dieses Parameters finden Sie in [Anmerkung 1](#).

-l Parameter

Nur die ausgewählten Felder aus den verarbeiteten Datensätzen anzeigen. *Parameter* ist eine durch Kommas getrennte Liste mit ganzzahligen Werten, wobei jede Ganzzahlwertzuordnung der numerischen Konstanten eines Felds zugeordnet wird. Weitere Informationen finden Sie im Abschnitt [amqsmon-Beispiel 5](#).

Wenn Sie diesen Parameter nicht angeben, werden alle verfügbaren Felder angezeigt.

Anmerkung:

1. Wenn Sie *-s StartTime* oder *-e EndTime* nicht angeben, werden die Nachrichten, die verarbeitet werden können, nicht durch die Eingeschränkte Zeit eingeschränkt.

amqsmmon (Formatierte Überwachungsdaten anzeigen), Beispiele

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

1. Siehe „[Nachrichtendaten der Warteschlangenstatistik](#)“ auf Seite 199 für eine Erklärung der Attribute.

Der folgende Befehl zeigt alle MQI-Statistiken vom Warteschlangenmanager saturn.queue.manager an:

```
amqsmmon -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]
OpenFailCount: [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]
CloseFailCount: [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]
InqFailCount: [0, 31, 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]
SetFailCount: [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. Der folgende Befehl zeigt alle Warteschlangenstatistiken für Warteschlange LOCALQ auf Warteschlangenmanager saturn.queue.manager an:

```
amqsmmon -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. Der folgende Befehl zeigt alle Statistikenachrichten an, die seit 15:30 Uhr am 30. April 2005 vom Warteschlangenmanager saturn.queue.manager aufgezeichnet wurden.

```
amqsmon -m saturn.queue.manager -t statistics -s "2005-04-30 15.30.00"
```

Die Ausgabe dieses Befehls folgt:

```

RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
...
RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics: 0
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...
QueueStatistics: 1
  QueueName: 'SAMPLEQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...

```

4. Siehe [„Nachrichtendaten der Warteschlangenabrechnung“](#) auf Seite 176 für eine Erklärung der Attribute.

Der folgende Befehl zeigt alle Abrechnungsnachrichten an, die am 30. April 2005 vom Warteschlangenmanager saturn.queue.manager aufgezeichnet wurden:

```
amqsmon -m saturn.queue.manager -t accounting -s "2005-04-30" -e "2005-04-30"
```

Die Ausgabe dieses Befehls folgt:

```

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.29'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.09.30'

```

```

CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'
SeqNumber: 0
ApplicationName: 'amqsput'
ApplicationPid: 8572
ApplicationTid: 1
UserId: 'admin'
ConnDate: '2005-03-16'
ConnTime: '15.09.29'
DiscDate: '2005-03-16'
DiscTime: '15.09.30'
DiscType: Normal
OpenCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [1, 0]
PutFailCount: 0
PutBytes: [4, 0]
GetCount: [0, 0]
GetFailCount: 0
GetBytes: [0, 0]
BrowseCount: [0, 0]
BrowseFailCount: 0
BrowseBytes: [0, 0]
CommitCount: 0
CommitFailCount: 0
BackCount: 0
InqCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-03-16'
IntervalStartTime: '15.16.22'
IntervalEndDate: '2005-03-16'
IntervalEndTime: '15.16.22'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010c0020'
SeqNumber: 0
ApplicationName: 'runmqsc'
ApplicationPid: 8615
ApplicationTid: 1
...

```

5. Der folgende Befehl durchsucht die Abrechnungswarteschlange und zeigt den Anwendungsnamen und die Verbindungskennung jeder Anwendung an, für die MQI-Abrechnungsdaten verfügbar sind:

```
amqsmon -m saturn.queue.manager -t accounting -b -l 7006,3024
```

Die Ausgabe dieses Befehls folgt:

```

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01e80322'
ApplicationName: 'WebSphere MQ\bin\amqsput.exe'
QueueAccounting: 0

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01ea0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01e90322'
ApplicationName: 's\IBM\MQ_4\bin64\amqsput.exe'
QueueAccounting: 0

```

```

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01ef0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d393035202020202020fcf1855e01e60322'
ApplicationName: 's\IBM\MQ_4\bin64\runmqsc.exe'
QueueAccounting: 0

5 Records Processed

```

Weitere Informationen zur Lokalisierung der in diesem Abschnitt verwendeten Variablen finden Sie unter [„Zuordnung für ApplicationName \(3024\) und ConnectionId \(7006\) suchen“](#) auf Seite 161 .

Details zur Lokalisierung von Variablen bei Auswahl von *statistics* für den Parameter **Type** im Befehl **amqsmn** finden Sie in [„Zuordnung für AvgTimeOnQ \(703\) und QmaxDepth \(739\) suchen“](#) auf Seite 161 .

*Zuordnung für **ApplicationName** (3024) und **ConnectionId** (7006) suchen*

In diesem Abschnitt wird erläutert, wie die im Beispiel verwendeten Variablen für die Überwachung von Abrechnungsdaten gefunden werden.

Zusammenfassung:

- 3024 bedeutet MQCACF_APPL_NAME, d. h. **ApplicationName**
- 7006 bedeutet MQBACF_CONNECTION_ID, d. h. **ConnectionId**

Zur Ermittlung der Zuordnung müssen Sie zwei Prozeduren ausführen:

1. Eine Erläuterung der Attribute „ApplicationName“ auf Seite 178 und „ConnectionId“ auf Seite 178 bei Auswahl von *accounting* für den Parameter **Type** im Befehl **amqsmn** finden Sie unter [„Nachrichtendaten der Warteschlangenabrechnung“](#) auf Seite 176 .

Suchen Sie in jedem Fall nach dem Attribut **Identifizier**.

Der **Identifizier** für **ApplicationName** ist MQCACF_APPL_NAME und für **ConnectionId** ist MQBACF_CONNECTION_ID

2. Suchen Sie nach den IDs, die Sie in Schritt „1“ auf Seite 161 gefunden haben.

Wechseln Sie zum Abschnitt [Konstanten](#) und blättern Sie abwärts, bis Sie die Liste MQCACF_* ([Befehlsformat Zeichenparametertypen](#)) finden. Suchen Sie MQCACF_APPL_NAME und der Wert 3024 wird angezeigt.

Suchen Sie in ähnlicher Weise nach der Liste MQBACF_* ([Befehlsformat Byteparametertypen](#)). Suchen Sie MQBACF_CONNECTION_ID und der Wert 7006 wird angezeigt.

Zuordnung für AvgTimeOnQ (703) und QmaxDepth (739) suchen

In diesem Abschnitt wird erläutert, wie die im Beispiel verwendeten Variablen für Überwachungsstatistikdaten gefunden werden.

Zusammenfassung:

- 703 bedeutet MQIAMO64_AVG_Q_TIME, d. h. **AvgTimeOnQ**
- 739 bedeutet MQIAMO_Q_MAX_DEPTH, d. h. **QMaxDepth**

Zur Ermittlung der Zuordnung müssen Sie zwei Prozeduren ausführen:

1. Eine Erläuterung der Attribute „AvgTimeOnQ“ auf Seite 203 und „QMaxDepth“ auf Seite 202 bei Auswahl von *statistics* für den Parameter **Type** im Befehl **amqsmn** finden Sie unter „Nachrichtendaten der Warteschlangenstatistik“ auf Seite 199 .

Suchen Sie in jedem Fall nach dem Attribut **Identifizier**.

Identifizier für **AvgTimeOnQ** ist MQIAMO64_AVG_Q_TIME und für **QMaxDepth** ist MQIAMO_Q_MAX_DEPTH.

2. Suchen Sie nach den IDs, die Sie in Schritt „1“ auf Seite 162 gefunden haben.

Wechseln Sie zum Abschnitt Konstanten und blättern Sie abwärts, bis Sie die Liste MQIAMO_* (Befehlsformat Ganzzahlüberwachung Parametertypen) finden. Suchen Sie MQIAMO_Q_MAX_DEPTH und Sie sehen den Wert 739.

Suchen Sie in ähnlicher Weise nach der Liste MQIAMO64_* (Befehlsformat 64-Bit-Parametertypen für Ganzzahlüberwachung) . Suchen Sie MQIAMO64_AVG_Q_TIME und der Wert 703 wird angezeigt.

Nachrichtenreferenz für Abrechnung und Statistik

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

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

Nachrichtendeskriptor

- Eine MQMD-Struktur

Nachrichtendaten

- Ein PCF-Header (MQCFH)
- Accounting- oder Statistiknachrichtendaten, die immer zurückgegeben werden
- Abrechnungs- oder Statistiknachrichtendaten, die zurückgegeben werden, falls verfügbar

Nachrichtenformat für Abrechnung und Statistik

Verwenden Sie diese Seite als Beispiel für die Struktur einer MQI-Abrechnungsnachricht.

Tabelle 23. MQI-Abrechnungsnachrichtenstruktur

MQMD-Struktur	MQCFH-Struktur des Abrechnungsnachrichtenheaders	MQI-Abrechnungsnachrichtendaten 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 Statistknachricht MQMD (Nachrichtendeskriptor)

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

Die Parameter und Werte im Nachrichtendeskriptor der Abrechnungs-und Statistknachricht 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 Statistknachricht 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 Statistknachrichten

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

Abrechnungs-und Statistknachricht MQCFH (PCF-Header)

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

Command

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

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 IBM MQ for z/OS.
Systemwarteschlange:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

Beschreibung:	Der Name des Warteschlangenmanagers.
ID:	MQCA_Q_MGR_NAME
Datentyp:	MQCFST
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH
Zurückgegeben:	Immer

IntervalStartDate

Beschreibung:	Das Datum des Startzeitbeginns des Überwachungszeitraums
ID:	MQCAMO_START_DATE
Datentyp:	MQCFST
Maximale Länge:	MQ_DATE_LENGTH
Zurückgegeben:	Immer

IntervalStartTime

Beschreibung:	Der Zeitpunkt des Startzeitpunktes des Überwachungszeitraums.
ID:	MQCAMO_START_TIME

Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer

IntervalEndDate

Beschreibung: Das Datum des Endes des Überwachungszeitraums
ID: MQCAMO_END_DATE
Datentyp: MQCFST
Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Immer

IntervalEndTime

Beschreibung: Die Zeit bis zum Ende des Überwachungszeitraums
ID: MQCAMO_END_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer

CommandLevel

Beschreibung: Die Befehlsebene des Warteschlangenmanagers
ID: MQIA_COMMAND_LEVEL
Datentyp: MQCFIN
Zurückgegeben: Immer

ConnectionId

Beschreibung: Die Verbindungs-ID für die IBM MQ-Verbindung
ID: MQBACF_CONNECTION_ID
Datentyp: MQCFBS
Maximale Länge: MQ_CONNECTION_ID_LENGTH
Zurückgegeben: Immer

SeqNumber

Beschreibung: Die Folgenummer. Dieser Wert wird für jeden nachfolgenden Datensatz für lange laufende Verbindungen erhöht.
ID: MQIACF_SEQUENCE_NUMBER
Datentyp: MQCFIN
Zurückgegeben: Immer

ApplicationName

Beschreibung: Der Name der Anwendung. Der Inhalt dieses Feldes entspricht dem Inhalt des Feldes *PutApplName* im Nachrichtendeskriptor.
ID: MQCACF_APPL_NAME

Datentyp: MQCFST
Maximale Länge: MQ_APPL_NAME_LENGTH
Zurückgegeben: Immer

ApplicationPid

Beschreibung: Die Prozess-ID des Betriebssystems der Anwendung.
ID: MQIACF_PROCESS_ID
Datentyp: MQCFIN
Zurückgegeben: Immer

ApplicationTid

Beschreibung: Die IBM MQ-Thread-ID der Verbindung in der Anwendung
ID: MQIACF_THREAD_ID
Datentyp: MQCFIN
Zurückgegeben: Immer

UserId

Beschreibung: Der Benutzer-ID-Kontext der Anwendung.
ID: MQCACF_USER_IDENTIFIER
Datentyp: MQCFST
Maximale Länge: MQ_USER_ID_LENGTH
Zurückgegeben: Immer

ConnDate

Beschreibung: Datum der MQCONN-Operation
ID: MQCAMO_CONN_DATE
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Wenn verfügbar

ConnTime

Beschreibung: Zeit der MQCONN-Operation
ID: MQCAMO_CONN_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Wenn verfügbar

ConnName

Beschreibung: Verbindungsname für Clientverbindung
ID: MQCACH_CONNECTION_NAME
Datentyp: MQCFST

Maximale Länge: MQ_CONN_NAME_LENGTH
Zurückgegeben: Wenn verfügbar

ChannelName

Beschreibung: Kanalname für Clientverbindung
ID: MQCACH_CHANNEL_NAME
Datentyp: MQCFST
Maximale Länge: MQ_CHANNEL_NAME_LENGTH
Zurückgegeben: Wenn verfügbar

RemoteProduct

Beschreibung: Ferne Produktkennung für Clientverbindung, die im Feld **RPRODUCT** von DISPLAY CHSTATUS angezeigt wird
ID: MQCACH_REMOTE_PRODUCT
Datentyp: MQCFST
Maximale Länge: MQ_REMOTE_PRODUCT_LENGTH
Zurückgegeben: Wenn verfügbar

RemoteVersion

Beschreibung: Die Version des fernen Produkts für die Clientverbindung, die im Feld **RVERSION** von DISPLAY CHSTATUS angezeigt wird.
ID: MQCACH_REMOTE_VERSION
Datentyp: MQCFST
Maximale Länge: MQ_REMOTE_VERSION_LENGTH
Zurückgegeben: Wenn verfügbar

DiscDate

Beschreibung: Datum der MQDISC-Operation
ID: MQCAMO_DISC_DATE
Datentyp: MQCFST
Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Wenn verfügbar

DiscTime

Beschreibung: Zeit der MQDISC-Operation
ID: MQCAMO_DISC_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Wenn verfügbar

DiscType

Beschreibung: Typ der Verbindung

ID: MQIAMO_DISC_TYPE
Datentyp: MQCFIN
Werte: Folgende Werte sind möglich:
MQDISCONNECT_NORMAL
Angefordert von Anwendung
MQDISCONNECT_IMPLICIT
Abnormale Beendigung der Anwendung
MQDISCONNECT_Q_MGR
Verbindung des WS-Managers unterbrochen
Zurückgegeben: Wenn verfügbar

OpenCount

Beschreibung: Die Anzahl der erfolgreich geöffneten Objekte, durch direkte Ausgabe eines Aufrufs an MQOPEN oder durch Verwendung des Verbs MQPUT1. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indiziert 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 indiziert 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 indiziert 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 indiziert 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 indiziert 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 indiziert 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 indiziert ist (siehe Referenznote 2).
ID:	MQIAMO_BROWSES
Datentyp:	MQCFIL
Zurückgegeben:	Wenn verfügbar

BrowseFailCount

Beschreibung:	Die Anzahl der nicht erfolgreichen MQGET-Aufrufe ohne Löschen.
ID:	MQIAMO_BROWSES_FAILED
Datentyp:	MQCFIN
Zurückgegeben:	Wenn verfügbar

BrowseBytes

Beschreibung:	Die Gesamtzahl der Byte, die für persistente und nicht persistente Nachrichten durchsucht wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe Referenznote 2).
ID:	MQIAMO64_BROWSE_BYTES
Datentyp:	MQCFIL64
Zurückgegeben:	Wenn verfügbar

CommitCount

Beschreibung: Die Anzahl der erfolgreichen Transaktionen. Diese Zahl schließt die Transaktionen ein, die implizit von der verbundenen Anwendung festgeschrieben wurden. Commit-Anforderungen, bei denen keine ausstehenden Arbeiten vorhanden sind, werden in dieser Anzahl berücksichtigt.

ID: MQIAMO_COMMITS

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar

CommitFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Transaktion abzuschließen.

ID: MQIAMO_COMMITS_FAILED

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar

BackCount

Beschreibung: Die Anzahl der verarbeiteten Zurücksetzungen, einschließlich impliziter Zurücksetzungen aufgrund einer abnormalen Abschaltung.

ID: MQIAMO_BACKOUTS

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar

InqCount

Beschreibung: Die Anzahl der erfolgreichen Objekte, die in den Quot-Wert versetzt wurden. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp 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 indiziert werden.

0-Die Subskription wurde geschlossen, aber nicht entfernt.

1-Die Subskription wurde geschlossen und entfernt.

ID: MQIAMO_UNSUBS_NDUR

Datentyp: MQCFIL

Zurückgegeben: Wenn verfügbar.

UnsubFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen Abmeldeanforderungen.

ID: MQIAMO_UNSUBS_FAILED

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar.

SubRqCount

Beschreibung: Die Anzahl der erfolgreichen MQSUBRQ-Anforderungen.

ID: MQIAMO_SUBRQS

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar.

SubRqFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQSUB-Anforderungen.

ID: MQIAMO_SUBRQS_FAILED

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar.

CBCount

Beschreibung: Die Anzahl der erfolgreichen MQCB-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indiziert werden.

0-Ein Callback wurde erstellt oder geändert.

1-Ein Callback wurde entfernt

2-Ein Callback wurde wieder aufgenommen

3-Ein Callback wurde ausgesetzt

ID: MQIAMO_CBS

Datentyp: MQCFIN

Zurückgegeben: Wenn verfügbar.

CBFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQCB-Anforderungen.
ID: MQIAMO_CBS_FAILED
Datentyp: MQCFIN
Zurückgegeben: Wenn verfügbar.

CtlCount

Beschreibung: Die Anzahl der erfolgreichen MQCTL-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indiziert werden.
0-Die Verbindung wurde gestartet.
1-Die Verbindung wurde gestoppt.
2-Die Verbindung wurde wieder aufgenommen
3-Die Verbindung wurde unterbrochen.
ID: MQIAMO_CTLs
Datentyp: MQCFIL
Zurückgegeben: Wenn verfügbar.

CtlFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQCTL-Anforderungen.
ID: MQIAMO_CTLs_FAILED
Datentyp: MQCFIN
Zurückgegeben: Wenn verfügbar.

StatCount

Beschreibung: Die Anzahl der erfolgreichen MQSTAT-Anforderungen.
ID: MQIAMO_STATS.
Datentyp: MQCFIN
Zurückgegeben: Wenn verfügbar.

StatFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQSTAT-Anforderungen.
ID: MQIAMO_STATS_FAILED
Datentyp: MQCFIN
Zurückgegeben: Wenn verfügbar.

PutTopicCount

Beschreibung: Die Anzahl persistenter und nicht persistenter Nachrichten, die erfolgreich in ein Thema gestellt wurden, mit Ausnahme von Nachrichten, die mit dem MQPUT1-Aufruf ausgegeben werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).
Hinweis: Nachrichten, die einen Warteschlangenalias verwenden, der in ein Thema aufgelöst wird, sind in diesem Wert enthalten.

ID: MQIAMO_TOPIC_PUTS
Datentyp: MQCFIL
Zurückgegeben: Wenn verfügbar.

PutTopicFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Nachricht in ein Thema zu stellen.
ID: MQIAMO_TOPIC_PUTS_FAILED
Datentyp: MQCFIN
Zurückgegeben: Wenn verfügbar.

Put1TopicCount

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Aufrufen erfolgreich in ein Thema gestellt wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert 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.

Nachrichtename: Warteschlangenabrechnungsnachricht.

Plattformen: Alle außer IBM MQ for z/OS.

Systemwarteschlange: SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

Beschreibung: Der Name des Warteschlangenmanagers.
ID: MQCA_Q_MGR_NAME
Datentyp: MQCFST
Maximale Länge: MQ_Q_MGR_NAME_LENGTH
Zurückgegeben: Immer

IntervalStartDate

Beschreibung: Das Datum des Startzeitbeginns des Überwachungszeitraums
ID: MQCAMO_START_DATE
Datentyp: MQCFST
Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Immer

IntervalStartTime

Beschreibung: Der Zeitpunkt des Startzeitpunktes des Überwachungszeitraums.
ID: MQCAMO_START_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer

IntervalEndDate

Beschreibung: Das Datum des Endes des Überwachungszeitraums
ID: MQCAMO_END_DATE
Datentyp: MQCFST
Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Immer

IntervalEndTime

Beschreibung: Die Zeit bis zum Ende des Überwachungszeitraums
ID: MQCAMO_END_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer

CommandLevel

Beschreibung: Die Befehlsebene des Warteschlangenmanagers
ID: MQIA_COMMAND_LEVEL
Datentyp: MQCFIN

Zurückgegeben: Immer

ConnectionId

Beschreibung: Die Verbindungs-ID für die IBM MQ-Verbindung

ID: MQBACF_CONNECTION_ID

Datentyp: MQCFBS

Maximale Länge: MQ_CONNECTION_ID_LENGTH

Zurückgegeben: Immer

SeqNumber

Beschreibung: Die Folgenummer. Dieser Wert wird für jeden nachfolgenden Datensatz für lange laufende Verbindungen erhöht.

ID: MQIACF_SEQUENCE_NUMBER

Datentyp: MQCFIN

Zurückgegeben: Immer

ApplicationName

Beschreibung: Der Name der Anwendung. Der Inhalt dieses Felds entspricht dem Inhalt des Felds PutApplName im Nachrichtendeskriptor.

ID: MQCACF_APPL_NAME

Datentyp: MQCFST

Maximale Länge: MQ_APPL_NAME_LENGTH

Zurückgegeben: Immer

ApplicationPid

Beschreibung: Die Prozess-ID des Betriebssystems der Anwendung.

ID: MQIACF_PROCESS_ID

Datentyp: MQCFIN

Zurückgegeben: Immer

ApplicationTid

Beschreibung: Die IBM MQ-Thread-ID der Verbindung in der Anwendung

ID: MQIACF_THREAD_ID

Datentyp: MQCFIN

Zurückgegeben: Immer

UserId

Beschreibung: Der Benutzer-ID-Kontext der Anwendung.

ID: MQCACF_USER_IDENTIFIER

Datentyp: MQCFST

Maximale Länge: MQ_USER_ID_LENGTH

Zurückgegeben: Immer

ChannelName

Beschreibung: Kanalname für Clientverbindung
ID: MQCACH_CHANNEL_NAME
Datentyp: MQCFST
Maximale Länge: MQ_CHANNEL_NAME_LENGTH
Zurückgegeben: Wenn verfügbar

ConnName

Beschreibung: Verbindungsname für Clientverbindung
ID: MQCACH_CONNECTION_NAME
Datentyp: MQCFST
Maximale Länge: MQ_CONN_NAME_LENGTH
Zurückgegeben: Wenn verfügbar

ObjectCount

Beschreibung: Die Anzahl der Warteschlangen, auf die in dem Intervall zugegriffen wurde, für das Abrechnungsdaten aufgezeichnet wurden. Dieser Wert wird auf die Anzahl der in der Nachricht enthaltenen *QAccountingData* -PCF-Gruppen festgelegt.
ID: MQIAMO_OBJECT_COUNT
Datentyp: MQCFIN
Zurückgegeben: Immer

QAccountingData

Beschreibung: Gruppierte Parameter, die Abrechnungsdetails für eine Warteschlange angeben
ID: MQGACF_Q_ACCOUNTING_DATA
Datentyp: MQCFGR

Parameter in Gruppe:	<i>QName</i> <i>CreateDate</i> <i>CreateTime</i> <i>QType</i> <i>QDefinitionType</i> <i>OpenCount</i> <i>OpenDate</i> <i>OpenTime</i> <i>CloseDate</i> <i>CloseTime</i> <i>PutCount</i> <i>PutFailCount</i> <i>Put1Count</i> <i>Put1FailCount</i> <i>PutBytes</i> <i>PutMinBytes</i> <i>PutMaxBytes</i> <i>GetCount</i> <i>GetFailCount</i> <i>GetBytes</i> <i>GetMinBytes</i> <i>GetMaxBytes</i> <i>BrowseCount</i> <i>BrowseFailCount</i> <i>BrowseBytes</i> <i>BrowseMinBytes</i> <i>BrowseMaxBytes</i> <i>TimeOnQMin</i> <i>TimeOnQAvg</i> <i>TimeOnQMax</i>
----------------------	---

Zurückgegeben: Immer

QName

Beschreibung:	Der Name der Warteschlange.
ID:	MQCA_Q_NAME
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Wenn verfügbar

CreateDate

Beschreibung:	Das Datum, an dem die Warteschlange erstellt wurde.
ID:	MQCA_CREATION_DATE
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>

Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Wenn verfügbar

CreateTime

Beschreibung: Die Zeit, zu der die Warteschlange erstellt wurde.
ID: MQCA_CREATION_TIME
Datentyp: MQCFST
In PCF-Gruppe enthalten: *QAccountingData*
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Wenn verfügbar

QType

Beschreibung: Der Typ der Warteschlange.
ID: MQIA_Q_TYPE
Datentyp: MQCFIN
In PCF-Gruppe enthalten: *QAccountingData*
Wert: MQQT_LOCAL
Zurückgegeben: Wenn verfügbar

QDefinitionType

Beschreibung: Der Typ der Warteschlangendefinition.
ID: MQIA_DEFINITION_TYPE
Datentyp: MQCFIN
In PCF-Gruppe enthalten: *QAccountingData*
Werte: Mögliche Werte:
MQQDT_PREDEFINED
MQQDT_PERMANENT_DYNAMIC
MQQDT_TEMPORARY_DYNAMIC
Zurückgegeben: Wenn verfügbar

OpenCount

Beschreibung: Gibt an, wie oft diese Warteschlange in dem betreffenden Intervall von der Anwendung geöffnet wurde, durch direkte Ausgabe eines Aufrufs an MQOPEN oder durch Verwendung des Verbs MQPUT1.
ID: MQIAMO_OPENS
Datentyp: MQCFIL
In PCF-Gruppe enthalten: *QAccountingData*
Zurückgegeben: Wenn verfügbar

OpenDate

Beschreibung:	Das Datum, an dem die Warteschlange zum ersten Öffnen in diesem Aufzeichnungsintervall geöffnet wurde. Wenn die Warteschlange zu Beginn dieses Intervalls bereits geöffnet war, gibt dieser Wert das Datum wieder, an dem die Warteschlange ursprünglich geöffnet wurde.
ID:	MQCAMO_OPEN_DATE
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

OpenTime

Beschreibung:	Die Zeit, zu der die Warteschlange zum ersten Mal in diesem Aufzeichnungsintervall geöffnet wurde. Wenn die Warteschlange bereits zu Beginn dieses Intervalls geöffnet war, spiegelt dieser Wert die Zeit wider, zu der die Warteschlange ursprünglich geöffnet wurde.
ID:	MQCAMO_OPEN_TIME
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

CloseDate

Beschreibung:	Das Datum der endgültigen Schließung der Warteschlange in diesem Aufzeichnungsintervall. Wenn die Warteschlange noch geöffnet ist, wird der Wert nicht zurückgegeben.
ID:	MQCAMO_CLOSE_DATE
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

CloseTime

Beschreibung:	Die Zeit der endgültigen Schließung der Warteschlange in diesem Aufzeichnungsintervall. Wenn die Warteschlange noch geöffnet ist, wird der Wert nicht zurückgegeben.
ID:	MQCAMO_CLOSE_TIME
Datentyp:	MQCFST
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

PutCount

Beschreibung:	Die Anzahl der persistenten und nicht persistenten Nachrichten, die erfolgreich in die Warteschlange gestellt wurden, mit Ausnahme von MQPUT1-Aufrufen. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe Referenznote 2).
ID:	MQIAMO_PUTS
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

PutFailCount

Beschreibung:	Die Anzahl der nicht erfolgreichen Versuche zum Angeben einer Nachricht mit Ausnahme von MQPUT1-Aufrufen.
ID:	MQIAMO_PUTS_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

Put1Count

Beschreibung:	Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Aufrufen erfolgreich in die Warteschlange gestellt wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert 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 Gesamtzahl der Byte, die für persistente und nicht persistente Nachrichten gestellt werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe Referenznote 2).
ID:	MQIAMO64_PUT_BYTES

Datentyp: MQCFIL64
In PCF-Gruppe
enthalten: *QAccountingData*
Zurückgegeben: Wenn verfügbar

PutMinBytes

Beschreibung: Die kleinste persistente und nicht persistente Nachrichtengröße, die in die Warteschlange gestellt wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

ID: MQIAMO_PUT_MIN_BYTES
Datentyp: MQCFIL
In PCF-Gruppe
enthalten: *QAccountingData*
Zurückgegeben: Wenn verfügbar

PutMaxBytes

Beschreibung: Die größte persistente und nicht persistente Nachrichtengröße, die in die Warteschlange gestellt wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

ID: MQIAMO_PUT_MAX_BYTES
Datentyp: MQCFIL
In PCF-Gruppe
enthalten: *QAccountingData*
Zurückgegeben: Wenn verfügbar

GeneratedMsgCount

Beschreibung: Die Anzahl der generierten Nachrichten. Generierte Nachrichten:

- Warteschlangenlänge-Hi-Ereignisse
- Niedrige Ereignisse in der Warteschlange

ID: MQIAMO_GENERATED_MSGS
Datentyp: MQCFIN
In PCF-Gruppe
enthalten: *QAccountingData*
Zurückgegeben: Wenn verfügbar

GetCount

Beschreibung: Die Anzahl der erfolgreichen destruktiven MQGET-Aufrufe für persistente und nicht persistente Nachrichten. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

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

GetFailCount

Beschreibung:	Die Anzahl der fehlgeschlagenen destruktiven MQGET-Aufrufe.
ID:	MQIAMO_GETS_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

GetBytes

Beschreibung:	Die Anzahl der Byte, die in destruktiven MQGET-Aufrufen für persistente und nicht persistente Nachrichten gelesen werden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe Referenznote 2).
ID:	MQIAMO64_GET_BYTES
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

GetMinBytes

Beschreibung:	Die Größe der kleinsten persistenten und nicht persistenten Nachricht, die aus der Warteschlange abgerufen wurde. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe Referenznote 2).
ID:	MQIAMO_GET_MIN_BYTES
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

GetMaxBytes

Beschreibung:	Die Größe der größten persistenten und nicht persistenten Nachricht, die aus der Warteschlange abgerufen wurde. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe Referenznote 2).
ID:	MQIAMO_GET_MAX_BYTES
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

BrowseCount

Beschreibung:	Die Anzahl der erfolgreichen nicht-destruktiven MQGET-Aufrufe für persistente und nicht persistente Nachrichten. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe Referenznote 2).
ID:	MQIAMO_BROWSES
Datentyp:	MQCFIL

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

BrowseFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQGET-Aufrufe ohne Löschen.
ID: MQIAMO_BROWSES_FAILED
Datentyp: MQCFIN
In PCF-Gruppe
enthalten: *QAccountingData*
Zurückgegeben: Wenn verfügbar

BrowseBytes

Beschreibung: Die Anzahl der Byte, die in nicht-destruktiven MQGET-Aufrufen gelesen wurden, die persistente Nachrichten zurückgegeben haben
ID: MQIAMO64_BROWSE_BYTES
Datentyp: MQCFIL64
In PCF-Gruppe
enthalten: *QAccountingData*
Zurückgegeben: Wenn verfügbar

BrowseMinBytes

Beschreibung: Die Größe der kleinsten persistenten und nicht persistenten Nachricht, die aus der Warteschlange durchsucht wurde. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).
ID: MQIAMO_BROWSE_MIN_BYTES
Datentyp: MQCFIL
In PCF-Gruppe
enthalten: *QAccountingData*
Zurückgegeben: Wenn verfügbar

BrowseMaxBytes

Beschreibung: Die Größe der größten persistenten und nicht persistenten Nachricht, die aus der Warteschlange durchsucht wurde. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).
ID: MQIAMO_BROWSE_MAX_BYTES
Datentyp: MQCFIL
In PCF-Gruppe
enthalten: *QAccountingData*
Zurückgegeben: Wenn verfügbar

TimeOnQMin

Beschreibung:	Die kürzeste Zeit, in der eine persistente und nicht persistente Nachricht in der Warteschlange verblieb, bevor sie destruktiv abgerufen wurde, in Mikrosekunden. Für Nachrichten, die unter Synchronisationspunkt abgerufen werden, enthält dieser Wert keine Zeit, bevor die get-Operation festgeschrieben wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe Referenznote 2).
ID:	MQIAMO64_Q_TIME_MIN
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

TimeOnQAvg

Beschreibung:	Die durchschnittliche Zeit, in der eine persistente und nicht persistente Nachricht in der Warteschlange verblieb, bevor sie destruktiv abgerufen wurde, in Mikrosekunden. Für Nachrichten, die unter Synchronisationspunkt abgerufen werden, enthält dieser Wert keine Zeit, bevor die get-Operation festgeschrieben wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe Referenznote 2).
ID:	MQIAMO64_Q_TIME_AVG
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

TimeOnQMax

Beschreibung:	Die längste Zeit, in der eine persistente und nicht persistente Nachricht in der Warteschlange verblieb, bevor sie destruktiv abgerufen wurde (in Mikrosekunden). Für Nachrichten, die unter Synchronisationspunkt abgerufen werden, enthält dieser Wert keine Zeit, bevor die get-Operation festgeschrieben wird. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe Referenznote 2).
ID:	MQIAMO64_Q_TIME_MAX
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QAccountingData</i>
Zurückgegeben:	Wenn verfügbar

CBCCount

Beschreibung:	Die Anzahl der erfolgreichen MQCB-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indiziert werden. 0-Ein Callback wurde erstellt oder geändert. 1-Ein Callback wurde entfernt 2-Ein Callback wurde wieder aufgenommen 3-Ein Callback wurde ausgesetzt
---------------	--

ID: MQIAMO_CBS
Datentyp: MQCFIN
Zurückgegeben: Wenn verfügbar.

CBFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQCB-Anforderungen.
ID: MQIAMO_CBS_FAILED
Datentyp: MQCFIN
Zurückgegeben: Wenn verfügbar.

MQI-Statistikdaten, Nachrichtendaten

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

Nachrichtenname:	MQI-Statistiknachricht.
Plattformen:	Alle außer IBM MQ for z/OS.
Systemwarteschlange:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

Beschreibung: Name des Warteschlangenmanagers.
ID: MQCA_Q_MGR_NAME.
Datentyp: MQCFST.
Maximale Länge: MQ_Q_MGR_NAME_LENGTH.
Zurückgegeben: Immer.

IntervalStartDate

Beschreibung: Das Datum zu Beginn des Überwachungszeitraums.
ID: MQCAMO_START_DATE.
Datentyp: MQCFST.
Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Immer.

IntervalStartTime

Beschreibung: Die Uhrzeit am Anfang des Überwachungszeitraums.
ID: MQCAMO_START_TIME.
Datentyp: MQCFST.
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer.

IntervalEndDate

Beschreibung: Das Datum am Ende des Überwachungszeitraums.
ID: MQCAMO_END_DATE.
Datentyp: MQCFST.

Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Immer.

IntervalEndTime

Beschreibung: Die Zeit am Ende des Überwachungszeitraums.
ID: MQCAMO_END_TIME.
Datentyp: MQCFST.
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer.

CommandLevel

Beschreibung: Die Befehlsebene des Warteschlangenmanagers.
ID: MQIA_COMMAND_LEVEL.
Datentyp: MQCFIN.
Zurückgegeben: Immer.

ConnCount

Beschreibung: Die Anzahl der erfolgreichen Verbindungen zum WS-Manager.
ID: MQIAMO_CONNS.
Datentyp: MQCFIN.
Zurückgegeben: Wenn verfügbar.

ConnFailCount

Beschreibung: Die Anzahl der fehlgeschlagenen Verbindungsversuche.
ID: MQIAMO_CONNS_FAILED.
Datentyp: MQCFIN.
Zurückgegeben: Wenn verfügbar.

ConnsMax

Beschreibung: Die maximale Anzahl gleichzeitiger Verbindungen im Aufzeichnungsintervall.
ID: MQIAMO_CONNS_MAX.
Datentyp: MQCFIN.
Zurückgegeben: Wenn verfügbar.

DiscCount

Beschreibung: Die Anzahl der Trennungsverbindungen vom Warteschlangenmanager. Dies ist eine ganze Zahl, die durch die folgenden Konstanten indexiert wird:

- MQDISCONNECT_NORMAL
- MQDISCONNECT_IMPLICIT
- MQDISCONNECT_Q_MGR

ID: MQIAMO_DISCS.
Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

OpenCount

Beschreibung: Die Anzahl der erfolgreich geöffneten Objekte, durch direkte Ausgabe eines Aufrufs an MQOPEN oder durch Verwendung des Verbs MQPUT1. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO_OPENS.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

OpenFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche zum Öffnen von Objekten. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO_OPENS_FAILED.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

CloseCount

Beschreibung: Die Anzahl der erfolgreich abgeschlossenen Objekte. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO_CLOSES.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

CloseFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche zum Schließen von Objekten. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO_CLOSES_FAILED.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

InqCount

Beschreibung: Die Anzahl der Objekte, die erfolgreich in den Quiet-Wert versetzt wurden. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO_INQS.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

InqFailCount

Beschreibung: Die Anzahl der fehlgeschlagenen Versuche, die Objekt inquire zu versuchen. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).

ID: MQIAMO_INQS_FAILED.
Datentyp: MQCFIL.
Zurückgegeben: Wenn verfügbar.

SetCount

Beschreibung: Die Anzahl der erfolgreich aktualisierten Objekte (SET). Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).
ID: MQIAMO_SETS.
Datentyp: MQCFIL.
Zurückgegeben: Wenn verfügbar.

SetFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen SET-Versuche. Dieser Parameter ist eine ganze Zahl, die nach Objekttyp indexiert ist (siehe [Referenznote 1](#)).
ID: MQIAMO_SETS_FAILED.
Datentyp: MQCFIL.
Zurückgegeben: Wenn verfügbar.

PutCount

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die erfolgreich in eine Warteschlange gestellt wurden, mit Ausnahme von MQPUT1-Anforderungen. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).
ID: MQIAMO_PUTS.
Datentyp: MQCFIL.
Zurückgegeben: Wenn verfügbar.

PutFailCount

Beschreibung: Die Anzahl der fehlgeschlagenen Versuche zum Angeben von Nachrichten.
ID: MQIAMO_PUTS_FAILED.
Datentyp: MQCFIN.
Zurückgegeben: Wenn verfügbar.

Put1Count

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Anforderungen erfolgreich in eine Warteschlange gestellt wurden. Dieser Parameter ist eine ganze Zahl, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).
ID: MQIAMO_PUT1S.
Datentyp: MQCFIL.
Zurückgegeben: Wenn verfügbar.

Put1FailCount

Beschreibung:	Die Anzahl der nicht erfolgreichen Versuche, eine persistente und nicht persistente Nachricht mit Hilfe von MQPUT1-Anforderungen in eine Warteschlange zu stellen. Dieser Parameter ist eine ganze Zahl, die nach dem Persistenzwert indexiert ist (siehe 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.

CBCCount

Beschreibung: Die Anzahl der erfolgreichen MQCB-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden.
0-Ein Callback wurde erstellt oder geändert.
1-Ein Callback wurde entfernt
2-Ein Callback wurde wieder aufgenommen
3-Ein Callback wurde ausgesetzt

ID: MQIAMO_CBS.
Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

CBFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQCB-Anforderungen.

ID: MQIAMO_CBS_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

CtlCount

Beschreibung: Die Anzahl der erfolgreichen MQCTL-Anforderungen. Dies ist ein Array mit Werten, die durch den Typ der Operation indexiert werden:

0-Die Verbindung wurde gestartet.

1-Die Verbindung wurde gestoppt.

2-Die Verbindung wurde wieder aufgenommen

3-Die Verbindung wurde unterbrochen.

ID: MQIAMO_CTLS.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

CtlFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQCTL-Anforderungen.

ID: MQIAMO_CTLS_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

StatCount

Beschreibung: Die Anzahl der erfolgreichen MQSTAT-Anforderungen.

ID: MQIAMO_STATS.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

StatFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen MQSTAT-Anforderungen.

ID: MQIAMO_STATS_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

SubCountDurHighWater

Beschreibung: Die hohe Wassermarkierung für die Anzahl der permanenten Subskriptionen während des Zeitintervalls. Dies ist ein Array mit Werten, die durch SUBTYPE indexiert werden.

- 0-Die Hochwasser-Markierung für alle permanenten Subskriptionen im System
- 1-Die Hochwassermarkierung für permanente Anwendungssubskriptionen (MQSUBTYPE_API)
- 2-Die Hochwasser-Markierung für die permanente Administratorsubskription (MQSUBTYPE_ADMIN)
- 3-Die Obere Wassermarke für dauerhafte Proxy-Subskriptionen (MQSUBTYPE_PROXY)

ID: MQIAMO_SUB_DUR_HIGHWATER

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

SubCountDurLowWater

Beschreibung: Die Niedrigwasserzeichen auf der Anzahl der permanenten Subskriptionen während des Zeitintervalls. Dies ist ein Array mit Werten, die von SUBTYPE indexiert werden.

- 0-Die Niedrigwassermarkierung für alle permanenten Subskriptionen im System
- 1-Die Unterwasser-Markierung für permanente Anwendungssubskriptionen (MQSUBTYPE_API)
- 2-Die Niedrigwassermarkierung für dauerhafte Verwaltungssubskriptionen (MQSUBTYPE_ADMIN)
- 3-Die Niedrigwassermarkierung für permanente Proxy-Subskriptionen (MQSUBTYPE_PROXY)

ID: MQIAMO_SUB_DUR_LOWWATER

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

SubCountNDurHighWater

Beschreibung: Die Hochwasser-Markierung für die Anzahl der nicht permanenten Subskriptionen während des Zeitintervalls. Dies ist ein Array mit Werten, die durch SUBTYPE indexiert werden.

- 0-Die Hochwasser-Markierung für alle nicht permanenten Subskriptionen im System
- 1-Die Hochwassermarkierung für nicht permanente Anwendungssubskriptionen (MQSUBTYPE_API)
- 2-Die Hochwasser-Markierung für die nicht permanente Administratorsubskription (MQSUBTYPE_ADMIN)
- 3-Die Obere Wassermarke für nicht permanente Proxy-Subskriptionen (MQSUBTYPE_PROXY)

ID: MQIAMO_SUB_NDUR_HIGHWATER

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

SubCountNDurLowWater

Beschreibung: Die Niedrigwassermarkierung für die Anzahl der nicht permanenten Subskriptionen während des Zeitintervalls. Dies ist ein Array mit Werten, die von SUBTYPE indiziert 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 indiziert ist (siehe [Referenznote 2](#)).

Hinweis: Nachrichten, die einen Warteschlangenalias verwenden, der in ein Thema aufgelöst wird, sind in diesem Wert enthalten.

ID: MQIAMO_TOPIC_PUTS.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

PutTopicFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Nachricht in ein Thema zu stellen.

ID: MQIAMO_TOPIC_PUTS_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

Put1TopicCount

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Aufrufen erfolgreich in ein Thema gestellt wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

Hinweis: Nachrichten, die einen Warteschlangenalias verwenden, der in ein Thema aufgelöst wird, sind in diesem Wert enthalten.

ID: MQIAMO_TOPIC_PUT1S.

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

Put1TopicFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Nachricht mit Hilfe von MQPUT1-Aufrufen in ein Thema zu stellen.

ID: MQIAMO_TOPIC_PUT1S_FAILED.

Datentyp: MQCFIN.

Zurückgegeben: Wenn verfügbar.

PutTopicBytes

Beschreibung: Die Anzahl der Byte, die mit Hilfe von put-Aufrufen für persistente und nicht persistente Nachrichten geschrieben werden, die in eine Veröffentlichungsoperation aufgelöst werden. Dies ist die Anzahl der Byte, die von der Anwendung gestellt werden, und nicht die resultierende Anzahl an Subskribenten, die an die Subskribenten zugestellt werden, siehe PublishMsgBytes für diesen Wert. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64_TOPIC_PUT_BYTES.

Datentyp: MQCFIL64.

Zurückgegeben: Wenn verfügbar.

PublishMsgCount

Beschreibung: Die Anzahl der Nachrichten, die innerhalb des Zeitintervalls an Subskriptionen gesendet wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64_PUBLISH_MSG_COUNT

Datentyp: MQCFIL.

Zurückgegeben: Wenn verfügbar.

PublishMsgBytes

Beschreibung: Die Anzahl der Byte, die in dem Zeitintervall an Subskriptionen gesendet wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indiziert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64_PUBLISH_MSG_BYTES

Datentyp: MQCFIL64.

Zurückgegeben: Wenn verfügbar.

Nachrichtendaten der Warteschlangenstatistik

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

Nachrichtenname: Warteschlangenstatistiknachricht.

Plattformen: Alle außer IBM MQ for z/OS.

Systemwarteschlange: SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

Beschreibung: Name des Warteschlangenmanagers

ID: MQCA_Q_MGR_NAME
Datentyp: MQCFST
Maximale Länge: MQ_Q_MGR_NAME_LENGTH
Zurückgegeben: Immer

IntervalStartDate

Beschreibung: Das Datum zu Beginn des Überwachungszeitraums
ID: MQCAMO_START_DATE
Datentyp: MQCFST
Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Immer

IntervalStartTime

Beschreibung: Die Uhrzeit am Beginn des Überwachungszeitraums.
ID: MQCAMO_START_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer

IntervalEndDate

Beschreibung: Das Datum am Ende des Überwachungszeitraums
ID: MQCAMO_END_DATE
Datentyp: MQCFST
Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Immer

IntervalEndTime

Beschreibung: Die Zeit am Ende des Überwachungszeitraums
ID: MQCAMO_END_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer

CommandLevel

Beschreibung: Die Befehlsebene des Warteschlangenmanagers
ID: MQIA_COMMAND_LEVEL
Datentyp: MQCFIN
Zurückgegeben: Immer

ObjectCount

Beschreibung:	Die Anzahl der Warteschlangenobjekte, auf die in dem Intervall zugegriffen wurde, für das Statistikdaten aufgezeichnet wurden. Dieser Wert wird auf die Anzahl der in der Nachricht enthaltenen QStatisticsData-PCF-Gruppen gesetzt.
ID:	MQIAMO_OBJECT_COUNT
Datentyp:	MQCFIN
Zurückgegeben:	Immer

QStatisticsData

Beschreibung:	Gruppierte Parameter, die Statistikdetails für eine Warteschlange angeben
ID:	MQGACF_Q_STATISTICS_DATA
Datentyp:	MQCFGR
Parameter in Gruppe:	<i>QName</i> <i>CreateDate</i> <i>CreateTime</i> <i>QType</i> <i>QDefinitionType</i> <i>QMinDepth</i> <i>QMaxDepth</i> <i>AvgTimeOnQ</i> <i>PutCount</i> <i>PutFailCount</i> <i>Put1Count</i> <i>Put1FailCount</i> <i>PutBytes</i> <i>GetCount</i> <i>GetFailCount</i> <i>GetBytes</i> <i>BrowseCount</i> <i>BrowseFailCount</i> <i>BrowseBytes</i> <i>NonQueuedMsgCount</i> <i>ExpiredMsgCount</i> <i>PurgeCount</i>
Zurückgegeben:	Immer

QName

Beschreibung:	Der Name der Warteschlange.
ID:	MQCA_Q_NAME
Datentyp:	MQCFST
Maximale Länge:	MQ_Q_NAME_LENGTH
Zurückgegeben:	Immer

CreateDate

Beschreibung:	Das Datum, an dem die Warteschlange erstellt wurde.
---------------	---

ID: MQCA_CREATION_DATE
Datentyp: MQCFST
Maximale Länge: MQ_DATE_LENGTH
Zurückgegeben: Immer

CreateTime

Beschreibung: Die Zeit, zu der die Warteschlange erstellt wurde.
ID: MQCA_CREATION_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer

QType

Beschreibung: Der Typ der Warteschlange.
ID: MQIA_Q_TYPE
Datentyp: MQCFIN
Wert: MQOT_LOCAL
Zurückgegeben: Immer

QDefinitionType

Beschreibung: Der Typ der Warteschlangendefinition.
ID: MQIA_DEFINITION_TYPE
Datentyp: MQCFIN
Werte: Mögliche Werte:

- MQQDT_PREDEFINED
- MQQDT_PERMANENT_DYNAMIC
- MQQDT_TEMPORARY_DYNAMIC

Zurückgegeben: Wenn verfügbar

QMinDepth

Beschreibung: Die minimale Warteschlangenlänge während des Überwachungszeitraums
ID: MQIAMO_Q_MIN_DEPTH
Datentyp: MQCFIN
In PCF-Gruppe enthalten: *QStatisticsData*
Zurückgegeben: Wenn verfügbar

QMaxDepth

Beschreibung: Die maximale Warteschlangenlänge während des Überwachungszeitraums
ID: MQIAMO_Q_MAX_DEPTH
Datentyp: MQCFIN

In PCF-Gruppe
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

AvgTimeOnQ

Beschreibung: Die durchschnittliche Latenzzeit (in Mikrosekunden) von Nachrichten, die während des Überwachungszeitraums aus der Warteschlange destruktiv abgerufen wurden. Dieser Parameter ist eine ganzzahlige Liste, die nach dem Persistenzwert indexiert ist (siehe [Referenznote 2](#)).

ID: MQIAMO64_AVG_Q_TIME

Datentyp: MQCFIL64

In PCF-Gruppe
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

PutCount

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die erfolgreich in die Warteschlange gestellt wurden, mit Ausnahme von MQPUT1-Anforderungen. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe [Referenznote 2](#).

ID: MQIAMO_PUTS

Datentyp: MQCFIL

In PCF-Gruppe
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

PutFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen Versuche, eine Nachricht in die Warteschlange zu stellen.

ID: MQIAMO_PUTS_FAILED

Datentyp: MQCFIN

In PCF-Gruppe
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

Put1Count

Beschreibung: Die Anzahl der persistenten und nicht persistenten Nachrichten, die mit Hilfe von MQPUT1-Aufrufen erfolgreich in die Warteschlange gestellt wurden. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe [Referenznote 2](#).

ID: MQIAMO_PUT1S

Datentyp: MQCFIL

In PCF-Gruppe
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

Put1FailCount

Beschreibung:	Die Anzahl der nicht erfolgreichen Versuche zum Angeben einer Nachricht unter Verwendung von MQPUT1-Aufrufen
ID:	MQIAMO_PUT1S_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

PutBytes

Beschreibung:	Die Anzahl der Byte, die in der Warteschlange in die Warteschlange geschrieben wurden.
ID:	MQIAMO64_PUT_BYTES
Datentyp:	MQCFIL64
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

GetCount

Beschreibung:	Die Anzahl der erfolgreichen zerstörerischen Anforderungen zum Abrufen persistenter und nicht persistenter Nachrichten. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe Referenznote 2 .
ID:	MQIAMO_GETS
Datentyp:	MQCFIL
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

GetFailCount

Beschreibung:	Die Anzahl der nicht erfolgreichen Anforderungen zum Löschen von Zerstör
ID:	MQIAMO_GETS_FAILED
Datentyp:	MQCFIN
In PCF-Gruppe enthalten:	<i>QStatisticsData</i>
Zurückgegeben:	Wenn verfügbar

GetBytes

Beschreibung:	Die Anzahl der Byte, die in destruktiven 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:	<i>QStatisticsData</i>

Zurückgegeben: Wenn verfügbar

BrowseCount

Beschreibung: Die Anzahl der erfolgreichen nicht-destruktiven Anforderungen zum Abrufen von persistenten und nicht persistenten Nachrichten. Dieser Parameter ist eine ganze Zahl, die nach Persistenzwert indexiert wird. Siehe [Referenznote 2](#).

ID: MQIAMO_BROWSES

Datentyp: MQCFIL

In PCF-Gruppe
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

BrowseFailCount

Beschreibung: Die Anzahl der nicht erfolgreichen get-Anforderungen ohne Löschen.

ID: MQIAMO_BROWSES_FAILED

Datentyp: MQCFIN

In PCF-Gruppe
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

BrowseBytes

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

ID: MQIAMO64_BROWSE_BYTES

Datentyp: MQCFIL64

In PCF-Gruppe
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

NonQueuedMsgCount

Beschreibung: Die Anzahl der Nachrichten, die die Warteschlange umgangen und direkt in eine wartende Anwendung übertragen wurden.

Das Umgehen einer Warteschlange kann unter bestimmten Umständen nur auftreten. Diese Zahl gibt an, wie oft IBM MQ in der Lage war, die Warteschlange zu umgehen, und nicht die Anzahl der Wartezeiten einer Anwendung.

ID: MQIAMO_MSGS_NOT_QUEUED

Datentyp: MQCFIN

In PCF-Gruppe
enthalten: *QStatisticsData*

Zurückgegeben: Wenn verfügbar

ExpiredMsgCount

Beschreibung: Die Anzahl persistenter und nicht persistenter Nachrichten, die gelöscht wurden, weil sie abgelaufen waren, bevor sie abgerufen werden konnten.

ID: MQIAMO_MSGS_EXPIRED
Datentyp: MQCFIN
In PCF-Gruppe
enthalten: *QStatisticsData*
Zurückgegeben: Wenn verfügbar

PurgeCount

Beschreibung: Die Anzahl der bereinigten Nachrichten.
ID: MQIAMO_MSGS_PURGED
Datentyp: MQCFIN
In PCF-Gruppe
enthalten: *QStatisticsData*
Zurückgegeben: Wenn verfügbar

Nachrichtendaten der Kanalstatistik

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

Nachrichtenname:	Kanalstatistiknachricht.
Plattformen:	Alle außer IBM MQ for z/OS.
Systemwarteschlange:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

Beschreibung: Der Name des Warteschlangenmanagers.
ID: MQCA_Q_MGR_NAME.
Datentyp: MQCFST.
Maximale Länge: MQ_Q_MGR_NAME_LENGTH.
Zurückgegeben: Immer.

IntervalStartDate

Beschreibung: Das Datum zu Beginn des Überwachungszeitraums.
ID: MQCAMO_START_DATE.
Datentyp: MQCFST.
Maximale Länge: MQ_DATE_LENGTH.
Zurückgegeben: Immer.

IntervalStartTime

Beschreibung: Die Uhrzeit am Anfang des Überwachungszeitraums.
ID: MQCAMO_START_TIME.
Datentyp: MQCFST.
Maximale Länge: MQ_TIME_LENGTH.
Zurückgegeben: Immer.

IntervalEndDate

Beschreibung: Das Datum am Ende des Überwachungszeitraums
ID: MQCAMO_END_DATE.
Datentyp: MQCFST.
Maximale Länge: MQ_DATE_LENGTH.
Zurückgegeben: Immer.

IntervalEndTime

Beschreibung: Die Zeit am Ende des Überwachungszeitraums
ID: MQCAMO_END_TIME.
Datentyp: MQCFST.
Maximale Länge: MQ_TIME_LENGTH
Zurückgegeben: Immer.

CommandLevel

Beschreibung: Die Befehlsebene des Warteschlangenmanagers.
ID: MQIA_COMMAND_LEVEL.
Datentyp: MQCFIN.
Zurückgegeben: Immer.

ObjectCount

Beschreibung: Die Anzahl der Kanalobjekte, auf die in dem Intervall zugegriffen wurde, für das Statistikdaten aufgezeichnet wurden. Dieser Wert wird auf die Anzahl der PCF-Gruppen 'ChlStatisticsData' gesetzt, die in der Nachricht enthalten sind.
ID: MQIAMO_OBJECT_COUNT
Datentyp: MQCFIN.
Zurückgegeben: Immer.

ChlStatisticsData

Beschreibung: Gruppierete Parameter, die Statistikdetails für einen Kanal angeben.
ID: MQGACF_CHL_STATISTICS_DATA.
Datentyp: MQCFGR.

Parameter in Gruppe:

ChannelName
ChannelType
RemoteQmgr
ConnectionName
MsgCount
TotalBytes
NetTimeMin
NetTimeAvg
NetTimeMax
ExitTimeMin
ExitTimeAvg
ExitTimeMax
FullBatchCount
IncplBatchCount
AverageBatchSize
PutRetryCount

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 vom Kanal verarbeiteten Stapel, 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 der Kanalattribute BATCHSZ oder BATCHLIM 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 165
- „Nachrichtendaten der Warteschlangenabrechnung“ auf Seite 176
- „MQI-Statistikdaten, Nachrichtendaten“ auf Seite 188
- „Nachrichtendaten der Warteschlangenstatistik“ auf Seite 199
- „Nachrichtendaten der Kanalstatistik“ auf Seite 206

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

Tabelle 24. Feldgruppe nach Objekttyp indiziert

Objekttyp	Wertkontext
MQOT_Q (1)	Enthält den Wert, der sich auf Warteschlangenobjekte bezieht.
MQOT_NAMELIST (2)	Enthält den Wert, der sich auf namelistische Objekte bezieht.
MQOT_PROCESS (3)	Enthält den Wert, der sich auf Prozessobjekte bezieht.
MQOT_Q_MGR (5)	Enthält den Wert, der sich auf WS-Manager-Objekte bezieht.
MQOT_CHANNEL (6)	Enthält den Wert, der sich auf Kanalobjekte bezieht.
MQOT_AUTH_INFO (7)	Enthält den Wert, der sich auf Authentifizierungsobjekte bezieht.
MQOT_TOPIC (8)	Enthält den Wert, der sich auf Themenobjekte bezieht.

Anmerkung: Es werden ein Array mit 13 MQCFIL- oder MQCFIL64-Werten zurückgegeben, aber nur die aufgeführten Werte sind aussagekräftig.

2. Dieser Parameter bezieht sich auf IBM MQ-Nachrichten. Dieser Parameter ist ein Array von Werten (MQCFIL oder MQCFIL64), die durch die folgenden Konstanten indiziert werden:

Tabelle 25. Array indexiert durch Persistenzwert	
Konstant	Wert
1	Enthält den Wert für nicht persistente Nachrichten.
2	Enthält den Wert für persistente Nachrichten.

Anmerkung: Der Index für jedes dieser Arrays beginnt bei Null, so dass sich ein Index von 1 auf die zweite Zeile des Arrays bezieht. Elemente dieser Arrays, die nicht in diesen Tabellen aufgeführt sind, enthalten keine Abrechnungs- oder Statistikdaten.

Anwendungsaktivitätstrace

Der Aktivitätstrace der Anwendung erzeugt detaillierte Informationen zum Verhalten von Anwendungen, die mit einem Warteschlangenmanager verbunden sind. Dabei wird das Verhalten einer Anwendung verfolgt und es wird eine detaillierte Ansicht der Parameter bereitgestellt, die von einer Anwendung während der Interaktion mit IBM MQ verwendet werden. Es zeigt auch die Reihenfolge der MQI-Aufrufe, die von einer Anwendung ausgegeben werden.

Verwenden Sie den Anwendungsaktivitätstrace, wenn Sie mehr Informationen benötigen, als von Ereignisüberwachung, Nachrichtenüberwachung, Accounting- und Statistiknachrichten und Real-time Monitoring bereitgestellt werden.

Anmerkung: Der Aktivitätstrace wird in der IBM MQ -Verbindung für jede Anwendung generiert. Wenn also Abrechnungsnachrichten aktiviert sind, zählen die Aktivitätstraceoperationen zu den MQI-Abrechnungsdaten für jede Anwendung.

IBM MQ unterstützt zwei Methoden zur Erfassung von Tracedaten zur Anwendungsaktivität.

- Zentrale Erfassung von Trace-Informationen der Anwendungsaktivität, wobei der Trace für Anwendungsaktivitäten erfasst und gelesen wird, indem die PCF-Nachrichten des Aktivitätentrace in die Systemwarteschlange SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE geschrieben werden.
- Subskription von Aktivitätstracedaten, die in bestimmte IBM MQ-Systemthemen geschrieben werden.

Beachten Sie, dass der Aktivitätstrace von IBM MQ for z/OS nicht unterstützt wird.

 Sie können jetzt Anwendungsnamen in den meisten Programmiersprachen angeben, die von IBM MQ unterstützt werden, und finden unter [Anwendungsname in unterstützten Programmiersprachen angeben](#) weitere Informationen hierzu.

Zentrale Erfassung von Trace-Informationen für Anwendungsaktivitäten konfigurieren

Eine Tracenachricht der Anwendungsaktivität ist eine PCF-Nachricht. Sie können den Aktivitätstrace mit Hilfe einer Konfigurationsdatei konfigurieren. Wenn Sie die zentrale Erfassung von Trace-Informationen für die Anwendungsaktivität konfigurieren möchten, setzen Sie das WS-Manager-Attribut ACTVTRC. Sie können diese Einstellung auf Verbindungsebene mit Hilfe von MQCONNX-Optionen oder unter Verwendung der Aktivitätstracekonfigurationsdatei auf Anwendungszeilengruppe überschreiben.

Informationen zu diesem Vorgang

Aktivitätstracenachrichten bestehen aus einer MQMD-Struktur: einer PCF-Headerstruktur (MQCFH), gefolgt von einer Reihe von PCF-Parametern. Eine Sequenz der PCF-Gruppen der Anwendung "ApplicationTraceData" folgt den PCF-Parametern. Diese PCF-Gruppen erfassen Informationen zu den MQI-Operationen, die eine Anwendung ausführt, während eine Verbindung zu einem Warteschlangenmanager besteht. Zum Konfigurieren des Aktivitätstrace verwenden Sie eine Konfigurationsdatei namens mqat.ini.

Um zu steuern, ob die Trace-Informationen der Anwendungsaktivität erfasst werden, konfigurieren Sie eine oder mehrere der folgenden Einstellungen:

1. Das WS-Manager-Attribut ACTVTRC.
2. Die ACTVCONO-Einstellungen (in der MQCNO-Struktur, die in MQCONNX übergeben wurde).
3. Die übereinstimmende Zeilengruppe für die Anwendung in der Aktivitätstracekonfigurationsdatei `mqat.ini`.

Die vorherige Sequenz ist signifikant. Das Attribut ACTVTRC wird durch die Einstellungen für ACTVCONO außer Kraft gesetzt, die durch die Einstellungen in der Datei `mqat.ini` außer Kraft gesetzt werden.

Traceeinträge werden nach Abschluss jeder Operation geschrieben, sofern nicht anders angegeben. Diese Einträge werden zuerst in die Systemwarteschlange `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE` geschrieben und dann in die Anwendungsaktivitätstracenachrichten geschrieben, wenn die Anwendung die Verbindung zum Warteschlangenmanager trennt. Bei Anwendungen mit langer Laufzeit werden temporäre Nachrichten geschrieben, wenn eines der folgenden Ereignisse eintritt:

- Die Lebensdauer der Verbindung erreicht einen definierten Zeitlimitwert.
- Die Anzahl der Operationen erreicht eine angegebene Zahl.
- Die Menge der im Speicher erfassten Daten erreicht die maximale Nachrichtenlänge, die für die Warteschlange zulässig ist.

Sie legen den Zeitlimitwert mit dem Parameter **ActivityInterval** fest. Die Anzahl der Operationen wird mit dem Parameter **ActivityCount** festgelegt. Beide Parameter werden in der Aktivitätstracekonfigurationsdatei `mqat.ini` angegeben.

Das Aktivieren des Trace der Anwendungsaktivität kann sich auf die Leistung auswirken. Der Systemaufwand kann reduziert werden, indem der **ActivityCount** und die **ActivityInterval**-Einstellungen optimiert werden. Siehe „Leistungswirkung des Trace der Anwendungsaktivität optimieren“ auf Seite 221.

Die einfachste Methode zur Anzeige der Inhalte von Tracenachrichten zur Anwendungsaktivität ist, das „Beispielprogramm 'amqsact'“ auf Seite 222 zu verwenden.

Vorgehensweise

1. [„Festlegen von ACTVTRC zum Steuern der Erfassung von Aktivitätstraceinformationen“](#) auf Seite 213.
2. [„Festlegen von MQCONNX-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen“](#) auf Seite 214.
3. [„Aktivitätstraceverhalten mit mqat.ini konfigurieren“](#) auf Seite 215.
4. [„Leistungswirkung des Trace der Anwendungsaktivität optimieren“](#) auf Seite 221.

Festlegen von ACTVTRC zum Steuern der Erfassung von Aktivitätstraceinformationen

Verwenden Sie das WS-Managerattribut ACTVTRC, um die Erfassung von Trace-Informationen für MQI-Anwendungsaktivitäten zu steuern.

Informationen zu diesem Vorgang

Tracenachrichten für Anwendungsaktivitäten werden nur für Verbindungen generiert, die nach der Aktivierung der Anwendungsaktivitätstrace gestartet werden. Der Parameter **ACTVTRC** kann die folgenden Werte haben:

ON

Die API-Aktivitätstracensammlung ist

OFF

Die API-Aktivitätstracesammlung ist

Anmerkung: Die Einstellung **ACTVTRC** kann durch den Parameter **ACTVCONO** des Warteschlangenmanagers überschrieben werden. Wenn Sie den Parameter **ACTVCONO** auf `ENABLED` setzen, kann die Einstellung **ACTVTRC** für eine bestimmte Verbindung mit dem Feld **Options** in der Struktur `MQCNO` überschrieben werden. Siehe [„Festlegen von MQCONNX-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen“](#) auf Seite 214.

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 222 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. Siehe „Leistungswirkung des Trace der Anwendungsaktivität optimieren“ auf Seite 221.

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 mqat.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 mqat.inidefiniert.

2. Setzen Sie den Parameter **ConnectOpts** auf dem MQCONNX-Aufruf auf MQCNO_ACTIVITY_TRACE_ENABLED .

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

MQCNO_ACTIVITY_TRACE_DISABLED

Der Aktivitätstrace ist für die Verbindung inaktiviert.

MQCNO_ACTIVITY_TRACE_ENABLED

Der Aktivitätstrace ist für die Verbindung aktiviert.

Anmerkung: Wenn eine Anwendung MQCNO_ACTIVITY_TRACE_ENABLED und MQCNO_ACTIVITY_TRACE_DISABLED für MQCONNX auswählt, schlägt der Aufruf mit dem Ursachencode MQRC_OPTIONS_ERROR fehl.

3. Stellen Sie sicher, dass diese Aktivitätstraceeinstellungen nicht durch Einstellungen in der Aktivitätstracekonfigurationsdatei mqat.ini außer Kraft gesetzt werden.

Weitere Informationen finden Sie unter „Aktivitätstraceverhalten mit mqat.ini konfigurieren“ auf Seite 215.

Nächste Schritte

Die einfachste Methode zur Anzeige der Inhalte von Tracenachrichten zur Anwendungsaktivität ist, das „Beispielprogramm 'amqsact'“ auf Seite 222 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. Siehe „Leistungswirkung des Trace der Anwendungsaktivität optimieren“ auf Seite 221.

Aktivitätstraceverhalten mit mqat.ini konfigurieren

Das Verhalten des Aktivitätstrace wird mit Hilfe einer Konfigurationsdatei namens `mqat.ini` konfiguriert. Diese Datei wird verwendet, um die Stufe und Häufigkeit der Tracedaten für die Berichtsaktivität zu definieren. Die Datei bietet außerdem die Möglichkeit, Regeln zu definieren, mit deren Hilfe der Aktivitätstrace auf der Basis des Namens einer Anwendung aktiviert und inaktiviert werden kann.

Informationen zu diesem Vorgang

Linux **AIX** Auf AIX and Linux-Systemen befindet sich die Datei `mqat.ini` im Verzeichnis mit den Warteschlangenmanagerdaten, das die gleiche Position wie die Datei `qm.ini` hat.

Windows Auf Windows-Systemen befindet sich die Datei `mqat.ini` im Verzeichnis `C:\Program Files\IBM\WebSphere MQ\mqgrs\queue_manager_name` mit den Warteschlangenmanagerdaten. Benutzer, die Anwendungen ausführen, für die ein Trace durchgeführt wird, benötigen die Berechtigung zum Lesen dieser Datei

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

Beim Ändern der Datei `mqat.ini` werden neu erstellte IBM MQ-Verbindungen entsprechend der geänderten Version verarbeitet. Vorhandene Verbindungen verwenden weiterhin die vorherige Version, es sei denn, die Parameter des Warteschlangenmanagers werden geändert, z. B. nach einem Befehl `ALTER QMGR`.

Diese Datei befolgt dasselbe Format für Zeilengruppenschlüssel und Parameter/Wert-Paare wie die Dateien `mqs.ini` und `qm.ini`.

Die Datei besteht aus einer einzelnen Zeilengruppe, **AllActivityTrace**, um die Stufe und Häufigkeit der Trace-Daten der Berichtsaktivitäten standardmäßig für alle Aktivitätstrace zu konfigurieren.

Die Datei kann auch mehrere Zeilengruppen **ApplicationTrace** enthalten. Jede dieser Eigenschaften definiert eine Regel für das Traceverhalten für eine oder mehrere Verbindungen, die auf der Übereinstimmung des Anwendungsnamens der Verbindungen zur Regel basiert.

Zeilengruppe 'AllActivityTrace'

Eine einzelne Zeilengruppe **AllActivityTrace** definiert Einstellungen für den Aktivitätstrace, der für alle IBM MQ-Verbindungen angewendet wird, bis er überschrieben wird.

Einzelne Werte in der Zeilengruppe **AllActivityTrace** können durch spezifischere Informationen in einer Zeilengruppe **ApplicationTrace stanza** außer Kraft gesetzt werden.

Wenn mehr als eine **AllActivityTrace**-Zeilengruppe angegeben wird, werden die Werte in der letzten Zeilengruppe verwendet. Für die in der ausgewählten **AllActivityTrace** angegebenen Parameter werden Standardwerte verwendet. Parameter und Werte aus früheren Zeilengruppen **AllActivityTrace** werden ignoriert.

Die folgenden Parameter können unter der Zeilengruppe **AllActivityTrace** angegeben werden:

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

Name	Werte (Standardwert in Fett-druck)	Beschreibung
ActivityInterval	0-99999999 (1)	Ungefächertes Zeitintervall in Sekunden zwischen Tracenachrichten. Alle Aktivitäten, die von einer Verbindung in diesem Intervall ausgeführt werden, werden in einer einzigen Nachricht geschrieben. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung disconnects (oder wenn die Anzahl der Aktivitäten erreicht ist).
ActivityCount	0-99999999 (100)	Die Anzahl der MQI-oder XA-Operationen zwischen Tracenachrichten. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung die Verbindung trennt (oder wenn das Aktivitätsintervall abgelaufen ist).
TraceLevel	LOW/ 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.
TraceMessageData	0 -104 857 600 (maximal 100 MB)	Die Menge der Nachrichtendaten, die in Byte für MQGET-, MQPUT-, MQPUT1- und Callback-Operationen aufgezeichnet wurden.
StopOnGetTraceMsg	EIN /AUS	Die Verwendung von Aktivitätstrace, Trace-Anwendungen, die auch Aktivitäts-Trace-Nachrichten verarbeiten, ist aufgrund einer möglichen Schleife nicht ratsam.
SubscriptionDelivery	BATCHED /SOFORT	Legt fest, ob die Parameter "ActivityInterval" und "ActivityCount" verwendet werden sollen, wenn eine oder mehrere Aktivitätstrace-Subskriptionen vorhanden sind. Wenn Sie diesen Parameter auf IMMEDIATE setzen, werden die Werte für "ActivityInterval" und "ActivityCount" mit den tatsächlichen Werten von 1 überschrieben, wenn die Trace-Daten eine übereinstimmende Subskription aufweisen. Jeder Aktivitätstracesatz wird nicht mit anderen Datensätzen aus derselben Verbindung ausgeliefert und stattdessen sofort an die Subskription gesendet, ohne dass es zu einer Verzögerung kommt.

Zeilengruppe 'ApplicationTrace'

Die Zeilengruppe 'ApplicationTrace' enthält eine Regel, mit der definiert wird, für welche IBM MQ-Verbindungen auf Basis des Anwendungsnamens ein Trace durchgeführt wird. Optional das Standardverhalten, das unter den Einstellungen "Allsettings" definiert ist, die die globale Tracestufe und die Häufigkeitseinstellungen außer Kraft setzen

Diese Zeilengruppe kann die Parameter ApplName, ApplFunction und ApplClass enthalten, die entsprechend den übereinstimmenden Regeln verwendet werden, die in Verbindungsübereinstimmungsregeln definiert sind, um zu bestimmen, ob die Zeilengruppe für eine bestimmte Verbindung gilt.

Die Zeilengruppe muss den Parameter 'Trace' enthalten, um festzustellen, ob diese Regel den Aktivitätstrace für übereinstimmende Verbindungen aktiviert oder inaktiviert.

Eine off-Regel kann verwendet werden, um den Trace für spezifischere Anwendungsnamen explizit zu inaktivieren und die ACTVTRC-Einstellung des Warteschlangenmanagers oder der Aktivitätstrace-Verbindungsoptionen außer Kraft zu setzen.

Die folgenden Parameter können in der Zeilengruppe **ApplicationTrace** angegeben werden:

<i>Tabelle 27. Parameter/Wert-Paare, die in der Anwendungstracekonfigurations-Konfigurationsdatei verwendet werden können</i>		
Name	Werte (Standardwert in Fett-druck)	Beschreibung
Trace	ON/OFF (Erforderlicher Parameter-kein Standardwert)	Aktivitätstrace-Switch. Dieser Schalter kann in der anwendungsspezifischen Zeilengruppe verwendet werden, um festzustellen, ob der Aktivitätstrace für den Geltungsbereich der aktuellen Anwendungszeilengruppe aktiv ist. Beachten Sie, dass dieser Wert die Einstellungen ACTVTRC und ACTVCONO für den WS-Manager außer Funktion setzt.
ApplName	Zeichenfolge (Erforderlicher Parameter-kein Standardwert)	<p>Dieser Wert wird verwendet, um festzustellen, auf welche Anwendungen die Zeilengruppe 'ApplicationTrace' angewendet wird. Sie wird mit dem Wert ApplName aus der API-Exit-Kontextstruktur (die äquivalent zu MQMD.PutApplName) ist, abgeglichen. Der Inhalt des ApplName-Werts variiert abhängig von der Anwendungsumgebung.</p> <p>Auf allen Plattformen außer z/OS wird nur der Teil von MQAXC.ApplName, der den Dateinamen enthält, mit dem Wert in der Zeilengruppe abgeglichen. Zeichen links vom Pfadtrennzeichen rechts werden ignoriert, wenn der Vergleich durchgeführt wird.</p> <p>z/OS Bei z/OS-Anwendungen wird der gesamte Wert von MQAXC.ApplName mit dem Wert in der Zeilengruppe abgeglichen.</p> <p>Ein einzelnes Platzhalterzeichen (*) kann am Ende des ApplName-Werts verwendet werden, um eine beliebige Anzahl von Zeichen nach diesem Punkt abzugleichen. Wenn der Wert für "ApplName" auf ein einzelnes Platzhalterzeichen (*) gesetzt wird, entspricht der Wert für "ApplName" allen Anwendungen.</p>

Tabelle 27. Parameter/Wert-Paare, die in der Anwendungstracekonfigurations-Konfigurationsdatei verwendet werden können (Forts.)

Name	Werte (Standardwert in Fett- druck)	Beschreibung
 Appl- Function	Zeichenfolge (Standardwert *)	Dieser Wert wird verwendet, um zu qualifizieren, auf welche Anwendungsprogramme die Zeilengruppe ApplicationTrace und der Wert AppName angewendet werden. Die Zeilengruppe ist optional, ist aber nur für IBM i-Warteschlangenmanager gültig. Ein einzelnes Platzhalterzeichen (*) kann am Ende des AppName -Werts verwendet werden, um eine beliebige Anzahl von Zeichen abzugleichen. Eine Zeilengruppe ApplicationTrace , die AppName = * und AppFunction = AMQSPUTO angibt, gilt beispielsweise für alle Invocations des AMQSPUTO-Programms von einem beliebigen Job aus.
AppClass	USER / MCA / ALL	Die Klasse der Anwendung. In der folgenden Tabelle finden Sie eine Erläuterung, wie die AppType -Werte den IBM MQ-Verbindungen entsprechen.

In der folgenden Tabelle ist dargestellt, wie die *AppClass* -Werte den Feldern **APICallerType** und **APIEnvironment** in der Kontext-Struktur der Verbindungs-API entsprechen.

Tabelle 28. Appclass-Werte und ihre Übereinstimmung mit den Feldern APICallerType und APIEnvironment

APPLCLASS	API-Aufruertyp:	API-Umgebung:	Beschreibung
BENUTZER	MQXACT_EXTERNAL	MQXE_OTHER	Es wird nur ein Trace für Benutzeranwendungen
Nachrichtenkanal-agent	(Beliebiger Wert)	MQXE_MCA MQXE_MCA_CLNTCONN MQXE_MCA_SVRCONN	Clients und Kanäle (amqrmppa)
ALLE	(Beliebiger Wert)	(Beliebiger Wert)	Trace für alle Verbindungen



Achtung: Sie müssen eine **APPLCLASS** von **MCA** für Clientbenutzeranwendungen verwenden, da eine Klasse von **USER** nicht mit diesen übereinstimmt.

Sie können beispielsweise den folgenden Code verwenden, um einen Trace für die Beispielanwendung **amqspuic** zu erstellen:

```

ApplicationTrace:
AppClass=MCA                                # Application type
                                             # Values: (USER | MCA | INTERNAL | ALL)
                                             # Default: USER
AppName=amqspuic    # Application name (may be wildcard)
                                             # (matched to app name without path)
                                             # Default: *
Trace=ON                                           # Activity trace switch for application
                                             # Values: ( ON | OFF )
                                             # Default: OFF
ActivityInterval=30                               # Time interval between trace messages
                                             # Values: 0-99999999 (0=off)
                                             # Default: 0
ActivityCount=1                                   # Number of operations between trace msgs
    
```

```

TraceLevel=MEDIUM
TraceMessageData=1000
# Values: 0-99999999 (0=off)
# Default: 0
# Amount of data traced for each operation
# Values: LOW | MEDIUM | HIGH
# Default: MEDIUM
# Amount of message data traced
# Values: 0-100000000
# Default: 0

```

Der Standardwert `mqat.ini`, der bei der Erstellung eines Warteschlangenmanagers generiert wird, enthält eine einzige Regel zum expliziten Inaktivieren des Aktivitätstrace für das bereitgestellte Aktivitätstrace-Beispiel `amqsact`.

Regeln für die Übereinstimmungen von Verbindungen

Der Warteschlangenmanager wendet die folgenden Regeln an, um festzustellen, welche Zeilengruppen für eine Verbindung verwendet werden sollen.

1. Ein in der Zeilengruppe **AllActivityTrace** angegebener Wert wird für die Verbindung verwendet, es sei denn, der Wert tritt auch in einer Zeilengruppe **ApplicationTrace** auf und die Zeilengruppe erfüllt die Übereinstimmungskriterien für die Verbindung, die in den Punkten [2](#), [3](#) und [4](#) beschrieben wird.
2. Der Wert von **App1Class** wird mit dem Typ der IBM MQ-Verbindung abgeglichen. Wenn der **App1Class** nicht mit dem Verbindungstyp übereinstimmt, wird die Zeilengruppe für diese Verbindung ignoriert.
3. Der **App1Name** -Wert in der Zeilengruppe wird mit dem Dateinamensabschnitt des Feldes **App1Name** von der API-Exit-Kontextstruktur (MQAXC) für die Verbindung abgeglichen.

Der Dateinamensabschnitt wird von den Zeichen rechts vom letzten Pfadtrennzeichen (/oder \) abgeleitet. Wenn die Zeilengruppe **App1Name** ein Platzhalterzeichen (*) enthält, werden nur die Zeichen links vom Platzhalterzeichen mit der äquivalenten Anzahl von Zeichen aus der **App1Name** der Verbindung verglichen.

Wenn z. B. der Zeilengruppenwert "FRE*" angegeben wird, werden nur die ersten drei Zeichen im Vergleich verwendet, so dass "path/FREEDOM" und "path\FREDDY" übereinstimmen, aber "path/FRIEND" nicht. Wenn der **App1Name** -Wert der Zeilengruppe nicht mit der Verbindung **App1Name** übereinstimmt, wird die Zeilengruppe für diese Verbindung ignoriert.

4. Wenn mehr als eine Zeilengruppe mit dem **App1Name** und dem **App1Class** der Verbindung übereinstimmt, wird die Zeilengruppe mit dem spezifischsten **App1Name** verwendet.

Der spezifischste **App1Name** ist als der definiert, der die meisten Zeichen verwendet, um den **App1Name** der Verbindung abzugleichen.

Beispiel: Wenn die Datei `ini` eine Zeilengruppe mit **App1Name** = "FRE*" und eine andere Zeilengruppe mit **App1Name** = "FREE*" enthält, wird die Zeilengruppe mit **App1Name** = "FREE*" als beste Übereinstimmung für eine Verbindung mit **App1Name** = "path/FREEDOM" ausgewählt, da sie vier Zeichen enthält (in der Erwägung, dass **App1Name** = "FRE*" stimmt mit nur drei überein).

5. Wenn nach dem Anwenden der Regeln in den Punkten [2](#), [3](#) und [4](#) mehr als eine Zeilengruppe mit den Verbindungen **App1Name** und **App1Class** der Verbindung übereinstimmt, werden die Werte aus dem letzten Abgleich verwendet und alle anderen Zeilengruppen werden ignoriert.

Standardeinstellungen für jede Regel überschreiben

Optional können die globalen Tracestufe und die Häufigkeitseinstellungen unter der Zeilengruppe **AllActivityTrace** für die Verbindungen überschrieben werden, die mit der Zeilengruppe **ApplicationTrace** übereinstimmen.

Die folgenden Parameter können unter der Zeilengruppe **ApplicationTrace** festgelegt werden. Wenn sie nicht festgelegt sind, wird der Wert von den Zeilengruppeneinstellungen **AllActivityTrace** übernommen:

- **ActivityInterval**
- **ActivityCount**
- **TraceLevel**
- **TraceMessageData**

- **StopOnTraceMsg**

Syntax von mqat.ini

Für das Format der Datei mqat.ini gelten die folgenden Syntaxregeln:

- Text, der mit einem Hash-oder Semikolon beginnt, wird als Kommentar betrachtet, der sich bis zum Ende der Zeile erstreckt.
- Die erste signifikante Zeile (ohne Kommentar) muss ein Zeilengruppenschlüssel sein.
- Ein Zeilengruppenschlüssel setzt sich aus dem Namen der Zeilengruppe gefolgt von einem Doppelpunkt zusammen.
- Ein Parameterwert-Paar besteht aus dem Namen eines Parameters, gefolgt von einem Gleichheitszeichen und dann dem Wert.
- Es kann nur ein Paar aus einem Parameterwert in einer Zeile stehen. (Ein Parameter-Wert darf nicht in eine andere Zeile umgebrochen werden.)
- Führende und abschließende Leerzeichen werden ignoriert. Es gibt keine Begrenzung für den Leerraum zwischen Zeilengruppenamen, Parameternamen und -werten oder Parameter/Wert-Paaren. Zeilenumbrüche sind bedeutsam und werden nicht ignoriert.
- Die maximale Länge für eine Zeile beträgt 2048 Zeichen.
- Bei den Zeilengruppenschlüsseln, Parameternamen und konstanten Parameterwerten wird die Groß-/Kleinschreibung nicht beachtet, die Variablenparameterwerte (*ApplName* und *DebugPath*) werden jedoch von der Groß-/Kleinschreibung abhängig gemacht.

Beispiel für Anwendungsaktivität-Trace-Datei

Das folgende Beispiel zeigt, wie die Konfigurationsdaten in der INI-Datei des Aktivitätstrace angegeben sind.

```
AllActivityTrace:
ActivityInterval=1
ActivityCount=100
TraceLevel=MEDIUM
TraceMessageData=0
StopOnGetTraceMsg=ON

ApplicationTrace:
ApplName=amqs*
Trace=ON
TraceLevel=HIGH
TraceMessageData=1000

ApplicationTrace:
ApplName=amqsact*
Trace=OFF
```

Die oben genannte Zeilengruppe **AllActivityTrace** definiert, wie der Aktivitätstrace standardmäßig ausgeführt wird, wenn er entweder über ApplicationTrace-Regeln oder durch das Attribut ACTVTRC des Warteschlangenmanagers oder über das Programm durch eine Anwendung aktiviert wird.

Die erste Zeilengruppe **ApplicationTrace** definiert eine Regel, die zu einer beliebigen MQI-Aktivität durch eine Anwendung führt, deren Name mit "amqs" beginnt, für die ein Trace durchgeführt wird. Der für diese Anwendungen generierte Trace hat einen hohen Detailausschnitt und enthält bis zu 1000 Byte an Nachrichtendaten. Das Aktivitätsintervall und die Anzahl der Parameter werden übernommen.

Die zweite Zeilengruppe **ApplicationTrace** definiert eine Regel, die den Trace für Anwendungen inaktiviert, die den Namen "amqsact" (die Aktivitätstrace-Stichprobe) starten. Diese Regel setzt die frühere 'on'-Regel für die Anwendung 'amqsact' außer Kraft, woraus sich kein Trace für diese Anwendung ergibt.

Ein Beispiel wird auch unter dem Namen mqat.ini im Verzeichnis mit den Beispielen für die Programmiersprache C bereitgestellt (dies ist dasselbe Verzeichnis, in dem sich auch die Datei 'amqsact.c' befindet). Diese Datei kann für Warteschlangenmanager, die aus einem früheren Release von IBM MQ migriert wurden, in das Verzeichnis mit den Warteschlangenmanagerdaten kopiert werden.

Nächste Schritte

Das Aktivieren des Trace der Anwendungsaktivität kann sich auf die Leistung auswirken. Der Systemaufwand kann reduziert werden, indem der **ActivityCount** und die **ActivityInterval** -Einstellungen optimiert werden. Siehe [„Leistungswirkung des Trace der Anwendungsaktivität optimieren“](#) auf Seite 221.

Leistungswirkung des Trace der Anwendungsaktivität optimieren

Die Aktivierung der Anwendungsaktivitätstrace kann zu einer Leistungseinbuße 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 `mqt.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 `mqt.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 `mqt.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. Weitere Informationen hierzu finden Sie unter [„Aktivitätstraceverhalten mit mqt.ini konfigurieren“](#) auf Seite 215 und [„Festlegen von MQCONN-Optionen zur Steuerung der Erfassung von Aktivitätstraceinformationen“](#) auf Seite 214.

- Prüfen 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 `mqt.ini` -Datei fest, um die erforderliche Mindestmenge an Daten zu erfassen.

`TraceLevel=LOW` hat die geringste Auswirkung auf die Messaging-Leistung. Siehe [„Aktivitätstraceverhalten mit mqt.ini konfigurieren“](#) auf Seite 215.

- Optimieren Sie die **ActivityCount** -und **ActivityInterval** -Werte in `mqt.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

Die einfachste Methode zur Anzeige der Inhalte von Tracenachrichten zur Anwendungsaktivität ist, das [„Beispielprogramm 'amqsact'“](#) auf Seite 222 zu verwenden.

Beispielprogramm 'amqsact'

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

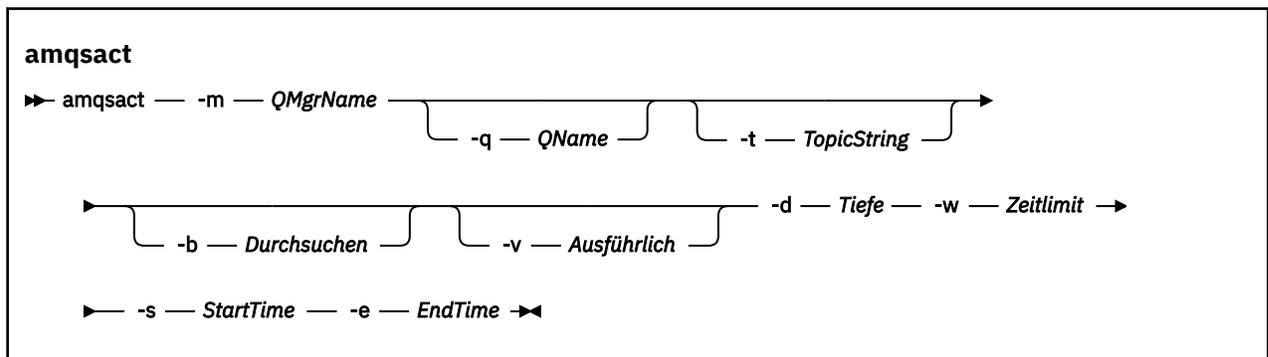
Das kompilierte Programm befindet sich im Verzeichnis 'samples':

- **Linux** **AIX** Unter AIX and Linux MQ_INSTALLATION_PATH/samp/bin
- **Windows** Unter Windows MQ_INSTALLATION_PATH\tools\c\Samples\Bin

Anzeigemodus

Standardmäßig verarbeitet **amqsact** im Anzeigemodus Nachrichten unter SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. Sie können dieses Verhalten außer Kraft setzen, indem Sie einen Warteschlangennamen oder eine Themenzeichenfolge angeben.

Sie können auch den angezeigten Ablaufverfolgungszeitraum steuern und angeben, ob die Aktivitätstrace-Nachrichten nach der Anzeige entfernt oder beibehalten werden sollen.



Erforderliche Parameter für den Anzeigemodus

-m QMgrName

Name des Warteschlangenmanagers.

-d Depth

Die Anzahl der Datensätze, die angezeigt werden sollen.

-w Timeout

Zeit zu warten, in Sekunden. Wenn im angegebenen Zeitraum keine Tracenachrichten angezeigt werden, wird **amqsact** beendet.

-s StartTime

Startzeit des Datensatzes für den Prozess.

-e EndTime

Endzeit des Datensatzes für den Prozess.

Optionale Parameter für den Anzeigemodus

-q QName

Geben Sie eine bestimmte Warteschlange an, um den Standardwarteschlangennamen zu überschreiben.

-t TopicString

Abonnieren eines Ereignisthemas

-b

Nur Datensätze durchsuchen

-v

Ausführliche Ausgabe

Beispielausgabe für den Anzeigemodus

Verwenden Sie **amqsact** im Warteschlangenmanager *TESTQM* mit einer ausführlichen Ausgabe in einem MQCONN-API-Aufruf:

```
amqsact -m TESTQM -v
```

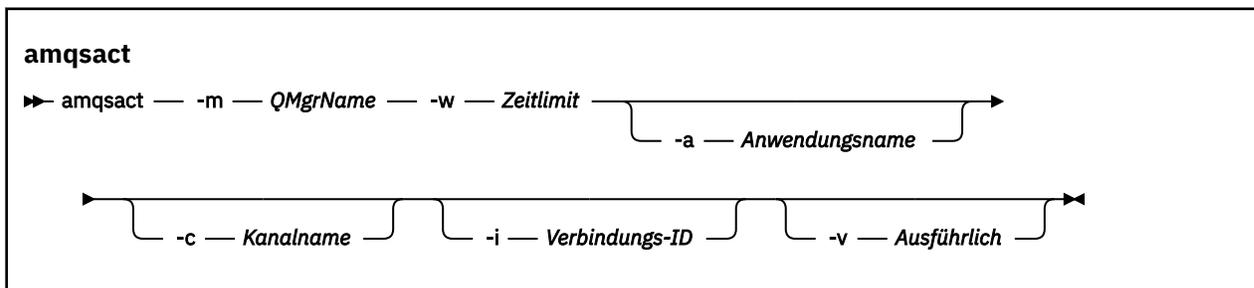
Der vorhergehende Befehl gibt die folgende Beispielausgabe aus:

```
MonitoringType: MQI Activity Trace
Correl_id:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM '
00000010: B5F6 4251 2000 E601
QueueManager: 'TESTQM'
Host Name: 'ADMINIB-1VTJ6N1'
IntervalStartDate: '2014-03-15'
IntervalStartTime: '12:08:10'
IntervalEndDate: '2014-03-15'
IntervalEndTime: '12:08:10'
CommandLevel: 750
SeqNumber: 0
ApplicationName: 'IBM MQ_1\bin\amqsput.exe'
Application Type: MQAT_WINDOWS_7
ApplicationPid: 14076
UserId: 'Emma_Bushby'
API Caller Type: MQXACT_EXTERNAL
API Environment: MQXE_OTHER
Application Function: ''
Appl Function Type: MQFUN_TYPE_UNKNOWN
Trace Detail Level: 2
Trace Data Length: 0
Pointer size: 4
Platform: MQPL_WINDOWS_7
MQI Operation: 0
Operation Id: MQXF_CONN
ApplicationTid: 1
OperationDate: '2014-03-15'
OperationTime: '12:08:10'
ConnectionId:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM '
00000010: FFFFFFFB5FFFFFFF6 4251 2000 FFFFFFFE601 '
QueueManager: 'TESTQM'
Completion Code: MQCC_OK
Reason Code: 0
```

Dynamischer Modus

Sie können den dynamischen Modus aktivieren, indem Sie einen Anwendungsnamen, einen Kanalnamen oder eine Verbindungs-ID als Argument für **amqsact** angeben. Beachten Sie, dass Sie Platzhalterzeichen im Namen verwenden können.

Im dynamischen Modus werden die Aktivitätstracedaten am Anfang der Stichprobe durch die Verwendung eines unfähigen Subskriptionsabonnements für ein Systemthema aktiviert. Die Erfassung von Aktivitätstracedaten wird gestoppt, wenn **amqsact** gestoppt wird. Sie müssen ein Zeitlimit für **amqsact** im dynamischen Modus angeben. Sie können mehrere Kopien von **amqsact** gleichzeitig ausführen, wobei jede Instanz eine Kopie aller Aktivitätstracedaten erhält.



Erforderliche Parameter für den dynamischen Modus

-m *QMgrName*

Name des Warteschlangenmanagers.

-w *Timeout*

Zeit zu warten, in Sekunden. Wenn im angegebenen Zeitraum keine Tracenachrichten angezeigt werden, wird **amqsact** beendet.

Optionale Parameter für den dynamischen Modus

-a *Application name*

Geben Sie einen Anwendungsnamen für die Erfassung von Nachrichten für

-c *Channel name*

Geben Sie einen Kanal an, in dem Nachrichten erfasst werden

-i *Connection ID*

Geben Sie eine Verbindung an, für die Nachrichten erfasst werden sollen.

-v

Ausführliche Ausgabe

Beispielausgabe für den dynamischen Modus

Mit dem folgenden Befehl werden Aktivitätstracenachrichten für Verbindungen generiert, die von Anwendungen, die mit dem Text " amqs " beginnen, hergestellt werden. Nach 30 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

Mit dem folgenden Befehl werden Aktivitätstracenachrichten für Aktivitäten auf dem Kanal QMGR1.TO.QMGR2 generiert und angezeigt. Nach 10 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

Mit dem folgenden Befehl werden ausführliche Aktivitätstracenachrichten für eine beliebige Aktivität in der vorhandenen IBM MQ-Verbindung mit den CONN-Wert "6B576B5420000701" und dem EXTCNN-Wert "414D5143514D475231202020202020" generiert und angezeigt. Nach einer Minute Inaktivität wird das **amqsact** -Programm beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D4752312020202020206B576B5420000701 -v
```

Trace-Informationen für Anwendungsaktivitäten abonnieren

Sie können Traceinformationen für die Anwendungsaktivität als Alternative zur Erfassung der Informationen über die Konfiguration auf Warteschlangenmanagerebene dynamisch subscribieren.

Informationen zu diesem Vorgang

Der Trace zur Anwendungsaktivität verfolgt das Verhalten einer Anwendung und stellt eine detaillierte Ansicht der Parameter bereit, die von einer Anwendung bei der Interaktion mit IBM MQ-Ressourcen verwendet werden. Es zeigt auch die Reihenfolge der MQI-Aufrufe, die von einer Anwendung ausgegeben werden.

Zusätzlich zum Schreiben von Tracedaten in die Systemwarteschlange bietet das Produkt die Möglichkeit, Aktivitätstracedaten dynamisch zu subscribieren, die in spezielle IBM MQ -Systemthemen geschrieben werden, anstatt die Informationen über die Konfiguration auf Warteschlangenmanagerebene zu erfassen.

Beachten Sie, dass das Produkt ab IBM MQ 9.0 keine Exits für diesen Zweck verwendet. Wenn Sie zuvor Exits verwendet haben, um die Anwendungsaktivität zu verfolgen, müssen Sie zur Verwendung der Ersetzungsmethode für die Erfassung des Anwendungsaktivitätstrace wechseln.

Die Erstellung einer Subskription aktiviert den Aktivitätstrace. Sie müssen keine WS-Manager-oder Anwendungsattribute für die zentrale Erfassung von Tracedaten festlegen. Eine explizite Blockierung des Aktivitätstrace durch Inaktivieren des Trace auf WS-Manager-oder Anwendungsebene blockiert jedoch auch den Aktivitätstrace, wenn er an alle übereinstimmenden Subskriptionen zugestellt wird.

Prozedur

- „Trace für Subskriptionen für Anwendungsaktivitäten“ auf Seite 225
- „Subskriptionen für Anwendungsaktivitätstrace erstellen“ auf Seite 225
- „amqsact zum Anzeigen von Tracenachrichten verwenden“ auf Seite 227
- „Tracestufen mit mqat.ini konfigurieren“ auf Seite 229

Trace für Subskriptionen für Anwendungsaktivitäten

Sie können ein IBM MQ-Systemthema subscribieren, um Traceinformationen für Anwendungsaktivitäten zu erfassen.

Sie subscribieren eine bestimmte IBM MQ-Systemthemenzeichenfolge, die die Aktivität darstellt, für die ein Trace durchgeführt werden soll. Das Subscribieren generiert automatisch Aktivitäts-Trace-Daten- und veröffentlicht sie in der Zielwarteschlange der Subskription. Wenn Sie die Subskription löschen, wird die Generierung von Aktivitätstracedaten für diese Subskription gestoppt.

Eine Subskription kann die Aktivität auf einer der folgenden Ressourcen verfolgen:

- Eine angegebene Anwendung
- Ein angegebener IBM MQ-Kanal
- Eine vorhandene IBM MQ-Verbindung

Sie können mehrere Subskriptionen mit unterschiedlichen oder denselben Themenzeichenfolgen erstellen. Wenn Sie mehrere Subskriptionen mit denselben Trace-Themenzeichenfolgen für Systemaktivitäten erstellen, empfängt jede Subskription eine Kopie der Aktivitätstracedaten, und dies kann negative Auswirkungen auf die Leistung haben.

Die Aktivierung einer beliebigen Stufe des Aktivitätstrace kann negative Auswirkungen auf die Leistung haben. Je mehr Subskriptionen oder die mehr Ressourcen subscribiert haben, um so größer ist der potenzielle Leistungsaufwand. Um den Aufwand für die Erfassung der Aktivitätstrace zu minimieren, werden die Daten in Nachrichten geschrieben und asynchron von der Anwendungsaktivität selbst an die Subskriptionen zugestellt. Häufig werden mehrere Operationen in eine einzelne Aktivitäts-Trace-Datennachricht geschrieben. Die asynchrone Operation kann eine Verzögerung zwischen der Anwendungsoperation und dem Empfang der Tracedaten, die die Operation aufzeichnet, einleiten.

Subskriptionen für Anwendungsaktivitätstrace erstellen

Sie können Subskriptionen für bestimmte Themen erstellen, um Tracedaten für Anwendungsaktivitäten zu erfassen.

Wenn eine Subskription für bestimmte Systemthemenzeichenfolgen erstellt wird, werden die entsprechenden PCF-Datennachrichten der Aktivitätstrace automatisch in dieser Subskription veröffentlicht. Ausführliche Informationen zum Subscribieren von Themen finden Sie unter [Publish/Subscribe-Messaging](#).

Die Themenzeichenfolgen haben das folgende Format:

```
$SYS/MQ/INFO/QMGR/qmgr_name/ActivityTrace/resource_type/resource_identifizier
```

Dabei gilt Folgendes:

- *Warteschlangenmanagername* gibt den Warteschlangenmanager an, mit dem die verfolgte Anwendung verbunden ist. *Warteschlangenmanagername* ist der Name des Warteschlangenmanagers, wobei alle

abschließenden Leerzeichen entfernt und alle Schrägstriche (/) durch ein Et-Zeichen (&) ersetzt werden. Zeichen.

- *resource_type* gibt den Typ der Ressourcendaten an, die für erfasst werden, und ist eine der folgenden Zeichenfolgen:
 - *AppName* zum Angeben einer Anwendung. In der Anforderung werden alle IBM MQ-Verbindungen mit einem Anwendungsnamen subskribiert, der mit einem der durch die *Ressourcen-ID* angegebenen Anwendungsnamen übereinstimmt.
 - *ChannelName* für die Angabe eines IBM MQ-Kanals.
 - *ConnectionId* für die Angabe einer IBM MQ-Verbindung.
- *resource_identifier* gibt die tatsächliche Ressource an. Das Format richtet sich nach dem Ressourcentyp:
 - Bei einem Ressourcentyp *AppName* ist *Ressourcen-ID* der abschließende Teil (der Wert, der auf das letzte Zeichen / oder \ folgt) im Anwendungsnamen für den Warteschlangenmanager, wobei alle abschließenden Leerzeichen entfernt wurden. Der Wert stimmt mit dem Wert für *AppName* aus der API-Exit-Kontextstruktur (MQAXC) überein. Der *AppName* einer Verbindung wird als APPLTAG-Wert zurückgegeben, wenn Sie den MQSC-Befehl **DISPLAY CONN** verwenden.
 - Für einen Ressourcentyp von *ChannelName* ist der *resource_identifier* der Name des Kanals, für den ein Trace durchgeführt werden soll. Wenn der Kanalname einen SVRCONN-Kanal identifiziert, wird ein Trace für alle Anwendungsaktivitäten für verbundene Clients durchgeführt. Wenn der Kanalname einen Warteschlangenmanager für den WS-Manager-Kanal identifiziert, wird ein Trace der eingehenden und abgehenden Nachrichten durchgeführt. Die *Ressourcen-ID* ist der Kanalname, aus dem alle abschließenden Leerzeichen entfernt und alle Zeichen '/' durch ein '&' ersetzt werden. Zeichen.
 - Für den Ressourcentyp *ConnectionId* ist die *Ressourcen-ID* die eindeutige Verbindungs-ID, die jeder Verbindung zugeordnet ist. Die Verbindungs-ID in der Themenzeichenfolge ist der vollständige 24-Byte-Wert, der als hexadezimale Zeichenfolge geschrieben wird. Dieser Wert ist die Verkettung des EXTCNN gefolgt von den CONN-Werten, die vom MQSC-Befehl **DISPLAY CONN** zurückgegeben werden.

Sie können Platzhalterzeichen in einem *resource_identifier* verwenden, um mehrere Ressourcenidentitäten in einer einzigen Subskription abzugleichen. Das Platzhalterzeichen kann entweder im Standardthemenstil ('#' oder '+') oder im Zeichenstil ('*' oder '?') verwendet werden. Wenn Sie das Platzhalterzeichen für den Topic-Stil verwenden, kann es nicht mit einem Teil eines Ressourcennamens kombiniert werden. Es kann nur verwendet werden, um alle möglichen Anwendungen, Kanäle oder Verbindungen abzugleichen. Die Verwendung von Platzhalterzeichen erhöht die Stufe der generierten Tracedaten, die sich auf die Leistung auswirken können.

Wenn Sie diese Themenzeichenfolgen subskribieren möchten, müssen Sie die Berechtigung " subscribe " haben. Systemthemen übernehmen keine Berechtigungen vom Stamm der Themenstruktur des Warteschlangenmanagers. Ein Benutzer muss Zugriff auf ein verwaltetes Themenobjekt erhalten, das sich an oder über dem \$SYS/MQ-Punkt in der Themenstruktur befinden kann. Sie können subskribieren, wenn Sie Zugriff auf die Warteschlange SYSTEM.ADMIN.TOPIC haben, obwohl dies Zugriff auf alle \$SYS/MQ-Themenzeichenfolgen gewährt, nicht nur für den Aktivitätstrace. Um den Zugriff genauer zu steuern, können neue verwaltete Themenobjekte für tiefere Punkte in der Baumstruktur definiert werden, entweder für alle Aktivitätstrace oder z. B. für einen bestimmten Anwendungsnamen oder einen bestimmten Kanalnamen.

Beispiele

Das folgende Beispiel zeigt eine Themenzeichenfolge für eine Anwendung namens amqspu, die auf einem Windows-System ausgeführt wird:

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqspu.c.exe
```

Das folgende Beispiel zeigt eine Themenzeichenfolge für einen Kanal:

```
$$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/SYSTEM.DEF.SVRCONN
```

Das folgende Beispiel zeigt eine Themenzeichenfolge für eine Verbindung:

```
$$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/Connection□  
Id/414D5143514D4752312020202020206B576B5420000701
```

Das folgende Beispiel zeigt eine Themenzeichenfolge, die eine Subskription für die Tracedaten für alle Kanäle auf dem Warteschlangenmanager QMGR1 erstellt:

```
$$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/#
```

Das folgende Beispiel zeigt eine Themenzeichenfolge, die eine Subskription erstellt, um Tracedaten für Anwendungen mit Namen zu erstellen, die mit " amqs " beginnen. (Beachten Sie, dass die Subskription mit dem Platzhalterzeichen " * " erstellt werden muss.)

```
$$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqs*
```

Zugehörige Konzepte

„Systemthemen für Überwachung und Aktivitätstrace“ auf Seite 301

Systemthemen in Themenstrukturen des Warteschlangenmanagers werden für die Ressourcenüberwachung (einige davon ähneln dem Inhalt von Statistiknachrichten) und als Methode zur Verarbeitung von Anwendungsaktivitätstraces verwendet.

amqsact zum Anzeigen von Tracenachrichten verwenden

Mit dem **amqsact** -Programm können Sie Tracenachrichten generieren und anzeigen.

Bei dem Programm **amqsact** handelt es sich um ein IBM MQ-Beispiel. Um dieses Beispiel zu verwenden, müssen Sie die mit dem Client verbundene ausführbare Datei **amqsactc** verwenden. Die ausführbare Datei befindet sich im Verzeichnis 'samples':

- Linux- und UNIX-Plattformen: `MQ_INSTALLATION_PATH/samp/bin64`
- Windows-Plattformen: `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`

Sie können **amqsact** auf zwei Arten verwenden:

Anzeigemodus

Formatieren und zeigen Sie die Tracedaten für Aktivitätstraces an, die an SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE zugestellt werden.

Dynamischer Modus

Erstellen Sie eine Subskription für eine Gruppe von Ressourcen und zeigen Sie den generierten Aktivitätstrace an, indem Sie **amqsact** ausführen.

Anzeigemodus

Standardmäßig verarbeitet **amqsact** im Anzeigemodus Nachrichten unter SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. Sie können dieses Verhalten außer Kraft setzen, indem Sie einen Warteschlangennamen oder eine Themenzeichenfolge angeben. Der Aktivitätstrace muss aktiviert werden, indem eine der Methoden verwendet wird, die im Abschnitt [Traceinformationen für Anwendungsaktivitäten erfassen](#) beschrieben werden. Sie können den angezeigten Ablaufverfolgungszeitraum steuern und angeben, ob die Aktivitätstrace-Nachrichten nach der Anzeige entfernt oder beibehalten werden sollen. Im Anzeigemodus verwendet **amqsact** die folgenden Argumente:

-m queue_manager_name

Erforderlich. Geben Sie den Warteschlangenmanager an, für den Tracenachrichten erfasst werden.

-q queue_name

Es werden nur Tracenachrichten angezeigt, die sich auf die benannte Warteschlange beziehen.

-t *topic_string*

Es werden nur Tracenachrichten angezeigt, die sich auf das benannte Thema beziehen.

-b

Geben Sie an, dass Tracenachrichten nach der Anzeige beibehalten werden

-v

Tracenachrichten im ausführlichen Modus anzeigen.

-d *depth*

Die Anzahl der Nachrichten, die angezeigt werden sollen.

-w *timeout*

Geben Sie ein Zeitlimit an. Wenn in diesem Zeitraum keine Tracenachrichten angezeigt werden, wird **amqsact** beendet.

-s *start_time*

Verwenden Sie dieses Argument mit dem Argument -e, um einen Zeitraum anzugeben. Es werden Tracenachrichten aus dem angegebenen Zeitraum angezeigt.

-e *end_time*

Verwenden Sie dieses Argument mit dem Argument -s, um einen Zeitraum anzugeben. Es werden Tracenachrichten aus dem angegebenen Zeitraum angezeigt.

Der folgende Befehl zeigt beispielsweise Aktivitätstrace-Nachrichten an, die in SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE gehalten werden, und löscht die Nachrichten nach der Anzeige:

```
amqsact -m QMGR1
```

Mit dem folgenden Befehl werden Aktivitätstracenachrichten in der angegebenen Warteschlange SUB.QUEUE angezeigt, und die Nachrichten werden nach der Anzeige gelöscht. Die Nachrichten werden weiterhin bis zu einem Zeitraum von 30 Sekunden angezeigt, ohne dass neue Nachrichten vergehen. Dieser Befehl kann beispielsweise mit einer Subskription für eine Themenzeichenfolge des Aktivitätstrace-Systems verwendet werden.

```
amqsact -m QMGR1 -q SUB.QUEUE.1 -w 30
```

Der folgende Befehl zeigt alle Aktivitätstracedaten im ausführlichen Format an, die derzeit in der Warteschlange SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE enthalten sind, die in dem angegebenen Zeitraum von 20 Minuten aufgetreten ist. Die Nachrichten bleiben nach der Anzeige in der Warteschlange.

```
amqsact -m QMGR1 -b -v -s 2014-12-31 23.50.00 -e 2015-01-01 00.10.00
```

Dynamischer Modus

Sie aktivieren den dynamischen Modus, indem Sie einen Anwendungsnamen, einen Kanalnamen oder eine Verbindungs-ID als Argument für **amqsact** angeben. Sie können Platzhalterzeichen im Namen verwenden. Im dynamischen Modus werden die Aktivitätstracedaten am Anfang des Beispiels durch die Verwendung einer nicht permanenten Subskription für ein Systemthema aktiviert. Die Erfassung von Aktivitätstracedaten wird gestoppt, wenn **amqsact** gestoppt wird. Sie müssen ein Zeitlimit für **amqsact** im dynamischen Modus angeben. Sie können mehrere Kopien von **amqsact** gleichzeitig ausführen und jede Instanz erhält eine Kopie aller Aktivitätstracedaten. Im dynamischen Modus verwendet **amqsact** die folgenden Argumente:

-m *queue_manager_name*

Erforderlich. Geben Sie den Warteschlangenmanager an, für den Tracenachrichten erfasst werden.

-w *timeout*

Erforderlich. Geben Sie ein Zeitlimit an. Wenn in diesem Zeitraum keine Tracenachrichten angezeigt werden, wird **amqsact** beendet.

-a *application_name*

Geben Sie eine Anwendung an, für die Nachrichten erfasst werden sollen.

-c channel_name

Geben Sie einen Kanal an, für den Nachrichten erfasst werden sollen.

-i connection_id

Geben Sie eine Verbindung an, für die Nachrichten erfasst werden sollen.

-v

Tracenachrichten im ausführlichen Modus anzeigen.

Der folgende Befehl generiert beispielsweise Aktivitäts-Trace-Nachrichten für alle Verbindungen, die von Anwendungen mit dem Namen " amqsget.exe " erstellt werden. Nach 30 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 30 -a amqsget.exe
```

Der folgende Befehl generiert Aktivitäts-Trace-Nachrichten für alle Verbindungen, die von Anwendungen erstellt werden, die mit dem Text " amqs " beginnen. Nach 30 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

Mit dem folgenden Befehl werden Aktivitätstracenachrichten für Aktivitäten auf dem Kanal QMGR1.TO.QMGR2 generiert und angezeigt. Nach 10 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

Mit dem folgenden Befehl werden Aktivitätstracenachrichten für jede Aktivität auf beliebigen Kanälen generiert und angezeigt. Nach 10 Sekunden Inaktivität wird das Programm **amqsact** beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 10 -c #
```

Mit dem folgenden Befehl werden ausführliche Aktivitätstracenachrichten für eine beliebige Aktivität in der vorhandenen IBM MQ-Verbindung mit den CONN-Wert "6B576B5420000701" und dem EXTCNN-Wert "414D5143514D475231202020202020" generiert und angezeigt. Nach einer Minute Inaktivität wird das **amqsact** -Programm beendet und es werden keine neuen Aktivitätstracedaten generiert.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D4752312020202020206B576B5420000701 -v
```

Tracestufen mit mqat.ini konfigurieren

Sie konfigurieren Tracestufen für einen Warteschlangenmanager, indem Sie der Konfigurationsdatei mqat.ini Werte für die Zeilengruppe 'AllActivityTrace' festlegen.

Sie können die folgenden Werte für die Zeilengruppe AllActivityTrace festlegen:

ActivityInterval

Zeitintervall in Sekunden zwischen Tracenachrichten. Der Aktivitätstrace verwendet keinen Zeitgeberthread, daher wird die Tracenachricht nicht zu dem Zeitpunkt geschrieben, zu dem die Zeit vergeht, sondern geschrieben wird, wenn die erste MQI-Operation nach Ablauf des Zeitintervalls ausgeführt wird. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung disconnects (oder wenn die Anzahl der Aktivitäten erreicht ist). Der Standardwert ist 1.

ActivityCount

Die Anzahl der MQI-Operationen zwischen Tracenachrichten. Wenn dieser Wert 0 ist, wird die Tracenachricht geschrieben, wenn die Verbindung die Verbindung trennt (oder wenn das Aktivitätsintervall abgelaufen ist). Der Standardwert ist 100.

TraceLevel

Die Menge an Parameterdetails, für die für jede Operation ein Trace durchgeführt wird. In der Beschreibung der einzelnen Operationen wird angegeben, welche Parameter für die einzelnen Trace-Ebenen angegeben werden. Setzen Sie die Option auf LOW, MEDIUM oder HIGH. Standardwert ist MEDIUM.

TraceMessageData

Die Menge der Nachrichtendaten, für die ein Trace in Byte für MQGET-, MQPUT-, MQPUT1- und Call-back-Operationen durchgeführt wird. Der Standardwert ist 0.

StopOnGetTraceMsg

Kann auf ON oder OFF gesetzt werden. Standardwert ist ON.

SubscriptionDelivery

Kann auf BATCHED oder IMMEDIATE gesetzt werden. Legt fest, ob die Parameter "ActivityInterval" und "ActivityCount" verwendet werden sollen, wenn eine oder mehrere Aktivitätstrace-Subskriptionen vorhanden sind. Wenn Sie diesen Parameter auf IMMEDIATE setzen, werden die Werte für "ActivityInterval" und "ActivityCount" mit den tatsächlichen Werten von 1 überschrieben, wenn die Trace-Daten eine übereinstimmende Subskription aufweisen. Jeder Aktivitätstracesatz wird nicht mit anderen Datensätzen aus derselben Verbindung ausgeliefert und stattdessen sofort an die Subskription gesendet, ohne dass es zu einer Verzögerung kommt. Mit der Einstellung IMMEDIATE wird der Leistungsaufwand für die Erfassung von Aktivitätstracedaten erhöht. Die Standardeinstellung ist BATCHED.

Trace-Nachrichtenreferenz für Anwendungsaktivitäten

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

Tracenachrichten für die Anwendungsaktivität sind standardmäßige IBM MQ-Nachrichten, die einen Nachrichtendeskriptor und Nachrichtendaten enthalten. Die Nachrichtendaten enthalten Informationen zu den MQI-Operationen, die von IBM MQ-Anwendungen ausgeführt werden, oder Informationen zu den Aktivitäten in einem IBM MQ-System.

Nachrichtendeskriptor

- Eine MQMD-Struktur

Nachrichtendaten

- Ein PCF-Header (MQCFH)
- Tracenachrichtendaten der Anwendungsaktivität, die immer zurückgegeben werden
- Tracenachrichtendaten der Anwendungsaktivität, die 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:	Nachrichtenfolgenummer. Dieses Feld ist die Folgenummer der Nachricht in einer Gruppe zusammengehöriger Nachrichten.
Datentyp:	MQLONG.
Werte:	1

Control

Beschreibung:	Steueroptionen.
Datentyp:	MQLONG.
Werte:	MQCFC_LAST.

CompCode

Beschreibung:	Beendigungscode.
Datentyp:	MQLONG.
Werte:	MQCC_OK.

Reason

Beschreibung:	Beendigungscode für Ursachencode.
Datentyp:	MQLONG.

Werte: MQRC_NONE.

ParameterCount

Beschreibung: Anzahl der Parameterstrukturen. Dieses Feld ist die Anzahl der Parameterstrukturen, die auf die MQCFH-Struktur folgen. Eine Gruppenstruktur (MQCFGR) und die darin enthaltenen Parameterstrukturen werden nur als eine einzige Struktur gezählt.

Datentyp: MQLONG.

Werte: 1 oder höher

Tracenachrichtendaten für Anwendungsaktivitäten

Unmittelbar hinter dem PCF-Header befindet sich eine Gruppe von Parametern, die das Zeitintervall für den Aktivitätstrace beschreiben. Diese Parameter geben auch die Reihenfolge der Nachrichten im Falle von Nachrichten an, die geschrieben werden. Die Reihenfolge und die Anzahl der Felder, die auf den Header folgen, ist nicht gewährleistet, so dass weitere Informationen in die Zukunft hinzugefügt werden können.

Nachrichtenname: Die Aktivitätstracenachricht

Systemwarteschlange: SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

QueueManager

Beschreibung: Der Name des Warteschlangenmanagers.

ID: MQCA_Q_MGR_NAME

Datentyp: MQCFST

Maximale Länge: MQ_Q_MGR_NAME_LENGTH

QSGName



Beschreibung: Der Name der Gruppe mit gemeinsamer Warteschlange, in der der Warteschlangenmanager Mitglied ist (nur z/OS).

ID: MQCA_QSG_NAME

Datentyp: MQCFST

Maximale Länge: MQ_Q_MGR_NAME_LENGTH

HostName

Beschreibung: Der Hostname der Maschine, auf der der WS-Manager ausgeführt wird.

ID: MQCACF_HOST_NAME

Datentyp: MQCFST

IntervalStartDate

Beschreibung: Das Datum des Beginns des Überwachungszeitraums.

ID: MQCAMO_START_DATE

Datentyp: MQCFST

Maximale Länge: MQ_DATE_LENGTH

IntervalStartTime

Beschreibung: Der Zeitpunkt des Startzeitpunktes des Überwachungszeitraums.
ID: MQCAMO_START_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH

IntervalEndDate

Beschreibung: Das Datum des Endes des Überwachungszeitraums.
ID: MQCAMO_END_DATE
Datentyp: MQCFST
Maximale Länge: MQ_DATE_LENGTH

IntervalEndTime

Beschreibung: Die Zeit bis zum Ende des Überwachungszeitraums.
ID: MQCAMO_END_TIME
Datentyp: MQCFST
Maximale Länge: MQ_TIME_LENGTH

CommandLevel

Beschreibung: Die IBM MQ-Befehlsebene.
ID: MQIA_COMMAND_LEVEL
Datentyp: MQCFIN

SeqNumber

Beschreibung: Die Folgenummer ist normalerweise null. Dieser Wert wird für jeden nachfolgenden Datensatz für lange laufende Verbindungen erhöht.
ID: MQIACF_SEQUENCE_NUMBER
Datentyp: MQCFIN

ApplicationName

Beschreibung: Der Name der Anwendung (Programmname).
ID: MQCACF_APPL_NAME
Datentyp: MQCFST
Maximale Länge: MQ_APPL_NAME_LENGTH

ApplClass

Beschreibung: Typ der Anwendung, die die Aktivität ausgeführt hat. Mögliche Werte: MQAT_*
ID: MQIA_APPL_TYPE
Datentyp: MQCFIN

ApplicationPid

Beschreibung: Die Prozess-ID des Betriebssystems der Anwendung.

ID: MQIACF_PROCESS_ID
Datentyp: MQCFIN

UserId

Beschreibung: Der Benutzer-ID-Kontext der Anwendung.
ID: MQCACF_USER_IDENTIFIER
Datentyp: MQCFST
Maximale Länge: MQ_USER_ID_LENGTH

APICallerType

Beschreibung: Der Typ der Anwendung. Mögliche Werte: MQXACT_EXTERNAL oder MQXACT_INTERNAL.
ID: MQIACF_API_CALLER_TYPE
Datentyp: MQCFIN

Environment

Beschreibung: Die Laufzeitumgebung der Anwendung. Mögliche Werte: MQXE_*
ID: MQIACF_API_ENVIRONMENT
Datentyp: MQCFIN

ChannelName

Beschreibung: Der Kanalname, der der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE_MCA' oder 'MQXE_MCA_SVRCONN' ist.
ID: MQCACH_CHANNEL_NAME
Datentyp: MQCFST
Maximale Länge: MQ_CHANNEL_NAME_LENGTH

ConnectionName

Beschreibung: Der Netzverbindungsname, der der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE_MCA' oder 'MQXE_MCA_SVRCONN' ist.
ID: MQCACH_CONNECTION_NAME
Datentyp: MQCFST
Maximale Länge: MQ_CONN_NAME_LENGTH

ChannelType

Beschreibung: Der Typ des Kanals, der der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE_MCA' oder 'MQXE_MCA_SVRCONN' ist. Mögliche Werte: MQCHT_*
ID: MQIACH_CHANNEL_TYPE
Datentyp: MQCFIN

RemoteProduct

Beschreibung:	Die ID des fernen Produkts, die der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE_MCA' oder 'MQXE_MCA_SVRCONN' ist.
ID:	MQCACH_REMOTE_PRODUCT
Datentyp:	MQCFST
Maximale Länge:	MQ_REMOTE_PRODUCT_LENGTH

RemoteVersion

Beschreibung:	Die Version des fernen Produkts, die der Verbindung zugeordnet ist. Dieser Parameter wird nur zurückgegeben, wenn der Wert des Parameters 'Environment' entweder 'MQXE_MCA' oder 'MQXE_MCA_SVRCONN' ist.
ID:	MQCACH_REMOTE_VERSION
Datentyp:	MQCFST
Maximale Länge:	MQ_REMOTE_VERSION_LENGTH

FunctionName

Beschreibung:	Der Name der letzten übergeordneten Funktion, die durch den Anfangsthread eingeleitet wurde.
ID:	MQCACF_APPL_FUNCTION
Datentyp:	MQCFST

FunctionType

Beschreibung:	Der Typ der letzten übergeordneten Funktion, die durch den Anfangsthread eingeleitet wurde. Mögliche Werte: MQFUN_*
ID:	MQIACF_APPL_FUNCTION_TYPE
Datentyp:	MQCFIN

Detail

Beschreibung:	Die Detailstufe, die für die Verbindung aufgezeichnet wird. Mögliche Werte: 1 = LOW 2 = MEDIUM 3 = HIGH
ID:	MQIACF_TRACE_DETAIL
Datentyp:	MQCFIN

TraceDataLength

Beschreibung:	Die Länge der Nachrichtendaten (in Byte), für die ein Trace für diese Verbindung durchgeführt wird.
ID:	MQIACF_TRACE_DATA_LENGTH
Datentyp:	MQCFIN

PointerSize

Beschreibung:	Die Länge (in Byte) von Zeigern auf der Plattform, auf der die Anwendung ausgeführt wird (zur Unterstützung bei der Interpretation von Binärstrukturen).
ID:	MQIACF_POINTER_SIZE

Datentyp: MQCFIN

Plattform

Beschreibung: Die Plattform, auf der der WS-Manager ausgeführt wird. Mögliche Werte: MQPL_*

ID: MQIA_PLATFORM

Datentyp: MQCFIN

Variablenparameter für MQI-Operationen der Anwendungsaktivität

Auf die Struktur der Anwendungsaktivitätsdaten MQCFGR folgt die Gruppe der PCF-Parameter, die der ausgeführten Operation entspricht. Die Parameter für jede Operation werden im folgenden Abschnitt definiert.

Die Tracestufe gibt die Stufe der Trace-Granularität an, die für die Parameter erforderlich ist, die in den Trace eingeschlossen werden sollen. Folgende Werte für die Tracestufe sind möglich:

1. Niedrig

Der Parameter ist eingeschlossen, wenn die Traceverarbeitung für " low ", " medium " oder " high " für eine Anwendung konfiguriert ist. Diese Einstellung bedeutet, dass ein Parameter immer in der Gruppe AppActivityData für die Operation enthalten ist. Diese Parametergruppe reicht aus, um einen Trace für die MQI-Aufrufe, die eine Anwendung vornimmt, zu verfolgen und zu sehen, ob sie erfolgreich sind.

2. Mittel

Der Parameter ist nur dann in der Gruppe AppActivityData für die Operation enthalten, wenn die "mittlere" oder "hohe" Aktivitätstracefunktion 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 Tracefunktion für "hohe" Aktivitäten für eine Anwendung konfiguriert ist. Diese Gruppe von Parametern enthält Speicherauszüge der Strukturen, die an die MQI- und XA-Funktionen übergeben werden. Aus diesem Grund enthält er weitere Informationen zu den Parametern, die in MQI- und XA-Aufrufen verwendet werden. Die Speicherauszüge des Strukturspeichers sind flache Kopien der Strukturen. Um fehlerhafte Versuche zu vermeiden, Zeiger zu dereferenzieren, werden die Zeigerwerte in den Strukturen auf NULL gesetzt.

Anmerkung: Die Version der Struktur, für die ein Speicherauszug erstellt wird, ist nicht notwendigerweise identisch mit der Version, die von einer Anwendung verwendet wird. Die Struktur kann durch einen API-Überkreuzungsexit, durch den Aktivitäts-Trace-Code oder durch den Warteschlangenmanager geändert werden. Ein Warteschlangenmanager kann eine Struktur in einer späteren Version ändern, aber der WS-Manager ändert ihn nie in eine frühere Version der Struktur. Dazu würde das Risiko eines Datenverlustes gefährdet.

MQBACK

Die Anwendung hat die MQBACK-MQI-Funktion gestartet.

CompCode

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.

PCF-Parameter: MQIACF_COMP_CODE

Tracestufe: 1

Typ: MQCFIN

Reason

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ	MQCFIN

QMGrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde. Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

MQBEGIN

Die Anwendung hat die MQBEGIN-MQI-Funktion gestartet.

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ	MQCFIN

Reason

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ	MQCFIN

MQBO

Beschreibung:	Die MQBEGIN-Optionsstruktur. Dieser Parameter ist nicht eingeschlossen, wenn ein Nullzeiger für den MQBEGIN-Aufruf verwendet wird.
PCF-Parameter:	MQBACF_MQBO_STRUCT
Tracestufe:	3
Typ	MQCFBS
Länge:	Die Länge (in Byte) der MQBO-Struktur.

QMGrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde. Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

MQCALLBACK

Die Anwendung hat die Funktion MQCALLBACK gestartet.

ObjectHandle

Beschreibung:	Die Objektkennung
PCF-Parameter:	MQIACF_HOBJ
Tracestufe:	1
Typ	MQCFIN

CallType

Beschreibung:	Warum die Funktion aufgerufen wurde. Einer der MQCBCT_* -Werte
PCF-Parameter:	MQIACF_CALL_TYPE
Tracestufe:	1
Typ	MQCFIN

MsgBuffer

Beschreibung:	Nachrichtendaten.
PCF-Parameter:	MQBACF_MESSAGE_DATA
Tracestufe:	1
Typ	MQCFBS
Länge:	Die Länge wird durch den Parameter TRACEDATA () gesteuert, der in der Konfiguration von APPTRACE festgelegt ist. Wenn TRACEDATA=NONE angegeben wird, wird dieser Parameter weggelassen.

MsgLength

Beschreibung:	Länge der Nachricht. (Taken wird aus dem Feld DataLength in der MQCBC-Struktur übernommen.)
PCF-Parameter:	MQIACF_MSG_LENGTH
Tracestufe:	1
Typ	MQCFIN

HighResTime

Beschreibung: Zeit der Operation in Mikrosekunden seit Mitternacht, 1. Januar 1970 (UTC)

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)

QMGrOpDuration

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64_QMGR_OP_DURATION

Tracestufe: 2

Typ: MQCFIN64

MQCB

Die Anwendung hat die Verwaltung der Callback-MQI-Funktion gestartet.

CallbackOperation

Beschreibung: Die Operation "Callback-Funktion verwalten". Setzen Sie diese Option auf einen der MQOP_*-Werte.

PCF-Parameter: MQIACF_MQCB_OPERATION

Tracestufe: 1

Typ: MQCFIN

CallbackType

Beschreibung: Der Typ der Callback-Funktion (CallbackType-Feld aus der MQCBD-Struktur). Auf einen der MQCBT_*-Werte setzen

PCF-Parameter: MQIACF_MQCB_TYPE

Tracestufe: 1

Typ: MQCFIN

CallbackOptions

Beschreibung: Die Callback-Optionen. Auf einen der MQCBDO_*-Werte setzen

PCF-Parameter: MQIACF_MQCB_OPTIONS

Tracestufe: 1

Typ: MQCFIN

CallbackFunction

Beschreibung:	Der Zeiger auf die Callback-Funktion, wenn er als Funktionsaufruf gestartet wurde.
PCF-Parameter:	MQBACF_MQCB_FUNCTION
Tracestufe:	1
Typ	MQCFBS
Länge:	Größe von MQPTR

CallbackName

Beschreibung:	Der Name der Callback-Funktion, wenn er als dynamisch verlinktes Programm gestartet wird.
PCF-Parameter:	MQCACF_MQCB_NAME
Tracestufe:	1
Typ	MQCFST
Länge:	Größe von MQCHAR128

ObjectHandle

Beschreibung:	Die Objektkennung
PCF-Parameter:	MQIACF_HOBJ
Tracestufe:	1
Typ	MQCFIN

MaxMsgLength

Beschreibung:	Maximale Nachrichtenlänge. Auf eine Ganzzahl setzen oder den Sonderwert MQCBD_FULL_MSG_LENGTH
PCF-Parameter:	MQIACH_MAX_MSG_LENGTH
Tracestufe:	2
Typ	MQCFIN

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ	MQCFIN

Reason

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ	MQCFIN

ResolvedQName

Beschreibung:	Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT_Q ist.
PCF-Parameter:	MQCACF_RESOLVED_LOCAL_Q_NAME
Tracestufe:	2
Typ	MQCFST
Länge:	MQ_Q_NAME_LENGTH.

ResObjectString

Beschreibung:	Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT_TOPIC lautet.
PCF-Parameter:	MQCACF_RESOLVED_OBJECT_STRING
Tracestufe:	2
Typ	MQCFST
Länge:	Länge variiert.

ResolvedType

Beschreibung:	Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT_Q', 'MQOT_TOPIC' oder 'MQOT_NONE'.
PCF-Parameter:	MQIACF_RESOLVED_TYPE
Tracestufe:	2
Typ	MQCFIN

Callback DescriptorStructure

Beschreibung:	Die MQCBD-Struktur. Dieser Parameter wird weggelassen, wenn ein NULL-MQCBC-Wert an den MQCB-Aufruf übergeben wird.
PCF-Parameter:	MQBACF_MQCBD_STRUCT
Tracestufe:	3
Typ	MQCFBS
Länge:	Die Länge der MQCBC-Struktur in Byte.

MsgDescStructure

Beschreibung:	Die MQMD-Struktur. Der Parameter 'MsgDescStructure' wird übergangen, wenn ein MQMD-Wert NULL an den MQCB-Aufruf übergeben wird.
PCF-Parameter:	MQBACF_MQMD_STRUCT
Tracestufe:	3
Typ	MQCFBS
Länge:	Die Länge der MQMD-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

GetMsgOptsStructure

Beschreibung:	Die MQGMO-Struktur. Dieser Parameter wird weggelassen, wenn ein NULL-MQGMO-Wert an den MQCB-Aufruf übergeben wird.
---------------	--

PCF-Parameter:	MQBACF_MQGMO_STRUCT
Tracestufe:	3
Typ	MQCFBS
Länge:	Die Länge der MQGMO-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

QMGrOpDuration

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

MQCLOSE

Die Anwendung hat die MQI-Funktion MQCLOSE gestartet.

ObjectHandle

Beschreibung:	Die Objektkennung
PCF-Parameter:	MQIACF_HOBJ
Tracestufe:	1
Typ	MQCFIN

CloseOptions

Beschreibung:	Optionen schließen
PCF-Parameter:	MQIACF_CLOSE_OPTIONS
Tracestufe:	1
Typ	MQCFIN

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ	MQCFIN

Reason

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ	MQCFIN

ResolvedQName

Beschreibung:	Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT_Q ist.
PCF-Parameter:	MQCACF_RESOLVED_LOCAL_Q_NAME
Tracestufe:	2
Typ	MQCFST
Länge:	MQ_Q_NAME_LENGTH.

ResObjectString

Beschreibung:	Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT_TOPIC lautet.
PCF-Parameter:	MQCACF_RESOLVED_OBJECT_STRING
Tracestufe:	2
Typ	MQCFST
Länge:	Länge variiert.

ResolvedType

Beschreibung:	Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT_Q', 'MQOT_TOPIC' oder 'MQOT_NONE'.
PCF-Parameter:	MQIACF_RESOLVED_TYPE
Tracestufe:	2
Typ	MQCFIN

QMgrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.
---------------	--

Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

MQCMIT

Die Anwendung hat die MQCMIT-MQI-Funktion gestartet.

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ	MQCFIN

Reason

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ	MQCFIN

QMGrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde. Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

MQCONN und MQCONNX

Die Anwendung hat die MQCONN-oder MQCONNX-MQI-Funktion gestartet.

ConnectionId

Beschreibung:	Die Verbindungs-ID (falls verfügbar) oder MQCONNID_NONE (falls nicht vorhanden)
PCF-Parameter:	MQBACF_CONNECTION_ID
Tracestufe:	1
Typ:	MQCFBS
Maximale Länge:	MQ_CONNECTION_ID_LENGTH

QueueManagerName

Beschreibung:	Der (nicht aufgelöste) Name des Warteschlangenmanagers, der im Aufruf MQCONN (X) verwendet wird.
PCF-Parameter:	MQCA_Q_MGR_NAME
Tracestufe:	1
Typ:	MQCFST
Maximale Länge:	MQ_Q_MGR_NAME_LENGTH

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ:	MQCFIN

Reason

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ:	MQCFIN

ConnectOptions

Beschreibung:	Von MQCNO_* -Werten abgeleitete Verbindungsoptionen Anmerkung: Nur MQCONNX
PCF-Parameter:	MQIACF_CONNECT_OPTIONS
Tracestufe:	2
Typ:	MQCFIN

ConnectionOptionsStructure

Beschreibung:	Die MQCNO-Struktur. Anmerkung: Nur MQCONNX)
PCF-Parameter:	MQBACF_MQCNO_STRUCT
Tracestufe:	3
Typ:	MQCFBS
Maximale Länge:	Die Länge in Byte der MQCNO-Struktur (die tatsächliche Größe hängt von der Strukturversion ab)

ChannelDefinitionStructure

Beschreibung:	Die MQCD-Struktur. Anmerkung: Nur Clientverbindungen
PCF-Parameter:	MQBACF_MQCD_STRUCT
Tracestufe:	3
Typ:	MQCFBS
Maximale Länge:	Die Länge der MQCD-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

QMgrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbraucht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde. Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ:	MQCFIN64

MQCTL

Die Anwendung hat die MQCTL-MQI-Funktion gestartet.

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1
Typ:	MQCFIN

Reason

Beschreibung:	Das Ursachencode-Ergebnis der Operation.
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	1
Typ:	MQCFIN

CtlOperation

Beschreibung:	Einer der MQOP_* -Werte
PCF-Parameter:	MQIACF_CTL_OPERATION
Tracestufe:	1
Typ:	MQCFIN

QMgrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbraucht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde. Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ:	MQCFIN64

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

QMGrOpDuration

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.
Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64_QMGR_OP_DURATION
Tracestufe: 2
Typ: MQCFIN64

MQINQ

Die Anwendung hat die MQINQ-MQI-Funktion gestartet.

ObjectHandle

Beschreibung: Die Objektkennung
PCF-Parameter: MQIACF_HOBJ
Tracestufe: 1
Typ: MQCFIN

CompCode

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter: MQIACF_COMP_CODE
Tracestufe: 1
Typ: MQCFIN

Reason

Beschreibung: Das Ursachencode-Ergebnis der Operation.
PCF-Parameter: MQIACF_REASON_CODE
Tracestufe: 1
Typ: MQCFIN

SelectorCount

Beschreibung:	Die Anzahl der Selektoren, die in der Selektorenfeldgruppe bereitgestellt werden.
PCF-Parameter:	MQIACF_SELECTOR_COUNT
Tracestufe:	2
Typ:	MQCFIN

Selectors

Beschreibung:	Die Liste der Attribute (Ganzzahl oder Zeichen), deren Werte von MQINQ zurückgegeben werden müssen.
PCF-Parameter:	MQIACF_SELECTORS
Tracestufe:	2
Typ:	MQCFIL

ResolvedQName

Beschreibung:	Der Name der Warteschlange, auf die vom ObjectHandle verwiesen wird, wenn ResolvedType MQOT_Q ist.
PCF-Parameter:	MQCACF_RESOLVED_Q_NAME
Tracestufe:	2
Typ:	MQCFST
Maximale Länge:	MQ_Q_NAME_LENGTH

ResObjectString

Beschreibung:	Der Objektname, auf den das Objekt 'ObjectHandle' verweist, wenn 'Resolved-Type' MQOT_TOPIC lautet.
PCF-Parameter:	MQCACF_RESOLVED_OBJECT_STRING
Tracestufe:	2
Typ:	MQCFST
Maximale Länge:	Länge variiert

ResolvedType

Beschreibung:	Der Typ des Objekts, auf das vom ObjectHandle verwiesen wird. Mögliche Werte sind 'MQOT_Q', 'MQOT_TOPIC' oder 'MQOT_NONE'.
PCF-Parameter:	MQIACF_RESOLVED_TYPE
Tracestufe:	2
Typ:	MQCFIN

IntAttrCount

Beschreibung:	Die Anzahl der ganzzahligen Attribute, die von der Inquire-Operation zurückgegeben werden.
PCF-Parameter:	MQIACF_INTATTR_COUNT
Tracestufe:	3
Typ:	MQCFIN

IntAttr

Beschreibung:	Die ganzzahligen Attributwerte, die von der Inquire-Operation zurückgegeben werden. Dieser Parameter ist nur enthalten, wenn 'IntAttrCount' bei der Rückgabe von MQINQ größer als 0 ist.
PCF-Parameter:	MQIACF_INT_ATTRS
Tracestufe:	3
Typ:	MQCFIL

CharAttr

Beschreibung:	Die Zeichenattribute, die von der Inquire-Operation zurückgegeben werden. Die Werte werden miteinander verkettet. Dieser Parameter ist nur enthalten, wenn 'CharAttrLength' bei der Rückgabe von MQINQ größer als 0 ist.
PCF-Parameter:	MQCACF_CHAR_ATTRS
Tracestufe:	3
Typ:	MQCFST

QMgrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde. Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ:	MQCFIN64

MQOPEN

Die Anwendung hat die MQOPEN-MQI-Funktion gestartet.

ObjectType

Beschreibung:	Der in MQOT.ObjectType übergebene Objekttyp
PCF-Parameter:	MQIACF_OBJECT_TYPE
Tracestufe:	1
Typ:	MQCFIN

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

ResolvedLocalQName ^{1 2}

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

ResolvedLocalQMgrName ^{1 2}

Beschreibung: Der Name des lokalen WS-Managers nach der Namensauflösung wurde ausgeführt.

PCF-Parameter: MQCACF_RESOLVED_LOCAL_Q_MGR

Tracestufe: 2

Typ: MQCFST

Bereich: Nur bei MQOD.Version >= MQOD_VERSION_3

Maximale Länge: MQ_Q_MGR_NAME_LENGTH

ResolvedQName ^{1 2}

Beschreibung: Der 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

ResolvedQMgrName ^{1 2}

Beschreibung: Enthält den Namen des Warteschlangenmanagers, nachdem die Namensauflösung ausgeführt wurde. Wenn MQOD.Version kleiner als MQOD_VERSION_3 ist, enthält dies den Wert von MQOD. Feld "ObjectQMgrName", nachdem der MQOPEN-Aufruf abgeschlossen wurde. Wenn MQOD.Version gleich oder größer als MQOD_VERSION_3 ist, enthält dies den Wert in der MQOD-Datei. Das Feld "ResolvedQMgrName".

PCF-Parameter: MQCACF_RESOLVED_Q_MGR

Tracestufe: 2

Typ: MQCFST

Maximale Länge: MQ_Q_MGR_NAME_LENGTH

AlternateSecurityId

Beschreibung: Alternative Sicherheitskennung. Nur vorhanden, wenn MQOD.Version gleich oder größer als MQOD_VERSION_3 ist, MQOO_ALTERNATE_USER_AUTHORITY angegeben ist und MQOD.AlternateSecurityId nicht gleich MQSID_NONE ist.

PCF-Parameter: MQBACF_ALTERNATE_SECURITYID

Tracestufe: 2

Typ: MQCFBS

Maximale Länge: MQ_SECURITY_ID_LENGTH

ObjectString

Beschreibung: Langer Objektname. Nur wenn MQOD.Version gleich oder größer als MQOD_VERSION_4 ist und das Feld VSLength von MQOD.ObjectString MQVS_NULL_TERMINATED oder größer als null ist.

PCF-Parameter: MQCACF_OBJECT_STRING

Tracestufe: 2

Typ: MQCFST

Maximale Länge: Länge variiert.

SelectionString

Beschreibung: Auswahlzeichenfolge. Wird nur eingeschlossen, wenn MQOD.Version gleich oder größer als MQOD_VERSION_4 und das Feld für die VSLength von MQOD ist. SelectionString ist MQVS_NULL_TERMINATED oder größer als null.

PCF-Parameter: MQCACF_SELECTION_STRING

Tracestufe: 2

Typ: MQCFST

Maximale Länge: Länge variiert.

ResObjectString

Beschreibung: Der lange Objektname, nachdem der WS-Manager den im Feld ObjectName angegebenen Namen aufgelöst hat. Nur für Themen und Warteschlangenaliasnamen, die auf ein Themenobjekt verweisen, wenn MQOD.Version gleich oder größer als MQOD_VERSION_4 und VSLength ist, ist MQVS_NULL_TERMINATED oder größer als null.

PCF-Parameter: MQCACF_RESOLVED_OBJECT_STRING

Tracestufe: 2

Typ: MQCFST

Maximale Länge: Länge variiert.

ResolvedType

Beschreibung: Der Typ des aufgelösten Objekts (Basisobjekts), das geöffnet wird. Wird nur eingeschlossen, wenn MQOD.Version gleich oder größer als MQOD_VERSION_4 ist. Mögliche Werte sind 'MQOT_Q', 'MQOT_TOPIC' oder 'MQOT_NONE'.

PCF-Parameter: MQIACF_RESOLVED_TYPE

Tracestufe: 2

Typ: MQCFIN

QMGrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde. Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

PCF-Gruppenheader der Verteilerliste der Anwendungsaktivität

Wenn die Funktion MQOPEN eine Verteilerliste öffnet, enthält die MQOPEN-Parameter eine AppActivityDistList-PCF-Gruppe für jede der Warteschlangen in der Verteilerliste bis zur Anzahl der Strukturen, die in RecsPresent nummeriert sind. Die PCF-Gruppe "Ap-pActivityDistList" kombiniert Informationen aus den MQOR- und MQRR-Strukturen, um den Namen der Warteschlange zu identifizieren, und zeigt das Ergebnis der offenen Operation in der Warteschlange an. Eine AppActivityDistList-Gruppe beginnt immer mit der folgenden MQCFGR-Struktur:

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.

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.

ObjectQMgrName

Beschreibung:	Der Name des Warteschlangenmanagers, auf dem die in Objektname angegebene Warteschlange definiert ist.
PCF-Parameter:	MQCACF_OBJECT_Q_MGR_NAME
Tracestufe:	2
Typ:	MQCFST
Länge:	MQ_Q_MGR_NAME_LENGTH. Wird nur eingeschlossen, wenn MQOR-Strukturen bereitgestellt werden.

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis des geöffneten Objekts für dieses Objekt anzeigt. Nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für den MQOPEN-Befehl MQRC_MULTIPLE_REASONS ist
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	2
Typ:	MQCFIN

Reason

Beschreibung:	Der Ursachencode, der das Ergebnis des offenen Objekts für dieses Objekt angibt. Nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für den MQOPEN-Befehl MQRC_MULTIPLE_REASONS ist
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	2
Typ:	MQCFIN

MQPUT

Die Anwendung hat die MQPUT-MQI-Funktion gestartet.

ObjectHandle

Beschreibung:	Die Objektkennung
PCF-Parameter:	MQIACF_HOBJ
Tracestufe:	1
Typ:	MQCFIN

PutOptions

Beschreibung:	Die Optionen 'put' aus 'MQPMO.Options'
PCF-Parameter:	MQIACF_PUT_OPTIONS
Tracestufe:	1
Typ:	MQCFIN

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1

Typ: MQCFIN

Reason

Beschreibung: Das Ursachencode-Ergebnis der Operation.

PCF-Parameter: MQIACF_REASON_CODE

Tracestufe: 1

Typ: MQCFIN

MsgBuffer

Beschreibung: Nachrichtendaten.

PCF-Parameter: MQBACF_MESSAGE_DATA

Tracestufe: 1

Typ: MQCFBS

Länge: Die Länge wird durch den Parameter TRACEDATA () gesteuert, der in der Konfiguration von APPTRACE festgelegt ist. Wenn TRACEDATA=NONE angegeben wird, wird dieser Parameter weggelassen.

MsgLength

Beschreibung: Länge der Nachricht.

PCF-Parameter: MQIACF_MSG_LENGTH

Tracestufe: 1

Typ: MQCFIN

RecsPresent

Beschreibung: Die Anzahl der vorhandenen Nachrichtendatensätze oder Antwortdatensätze. Nur enthalten, wenn MQPMO Version \geq 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

ResolvedLocalQName ³

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

ResolvedLocalQMgrName ³

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_STRUCT

Tracestufe: 3

Typ: MQCFBS

Länge: Die Länge der MQPMO-Struktur in Byte (tatsächliche Größe hängt von der Strukturversion ab)

QMGrOpDuration

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64_QMGR_OP_DURATION

Tracestufe: 2

Typ: MQCFIN64

MQPUT Application Activity Distribution List PCF-Gruppen-Header-Struktur

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

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 265. Die PCF-Gruppe AppActivityDistList kombiniert Informationen aus den MQPMR- und MQRR-Strukturen, um die PUT-Parameter zu identifizieren, und geben das Ergebnis der PUT-Operation für jede Warteschlange an. Für MQPUT-Operationen enthält die Gruppe AppActivityDistList einige oder alle der folgenden Parameter (der Code und der Ursachencode sind vorhanden, wenn der Ursachencode MQRC_MULTIPLE_REASONS lautet und die anderen Parameter durch das Feld MQPMO.PutMsgRecFields bestimmt werden):

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt. Wird nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für den MQPUT MQRC_MULTIPLE_REASONS ist
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	2
Typ:	MQCFIN

Reason

Beschreibung:	Der Ursachencode, der das Ergebnis der Einordnung für dieses Objekt angibt. Wird nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für den MQPUT MQRC_MULTIPLE_REASONS ist
PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	2
Typ:	MQCFIN

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

ResolvedLocalQName ⁴

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

ResolvedLocalQMgrName ⁴

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)

QMGrOpDuration

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

MQPUT1 AppActivityDistList PCF-Header-Struktur

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 265. Die PCF-Gruppe "AppActivityDistList" kombiniert Informationen aus den MQOR-, MQPMR- und MQRR-Strukturen, um die Objekte und die PUT-Parameter zu identifizieren, und geben das Ergebnis der PUT-Operation für jede Warteschlange an. Für MQPUT1-Operationen enthält die Gruppe AppActivityDistList einige oder alle der folgenden Parameter (der CompCode, Reason, ObjectName und ObjectQMGrName ist vorhanden, wenn der Ursachencode MQRC_MULTIPLE_REASONS ist und die anderen Parameter durch das Feld MQPMO.PutMsgRecFields bestimmt werden):

CompCode

Beschreibung: Der Beendigungscode, der das Ergebnis der Einlegung für dieses Objekt angibt. Wird nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für MQPUT1 MQRC_MULTIPLE_REASONS ist

PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	2
Typ:	MQCFIN

Reason

Beschreibung: Der Ursachencode, der das Ergebnis der Einordnung für dieses Objekt angibt. Wird nur eingeschlossen, wenn MQRR-Strukturen bereitgestellt werden und der Ursachencode für MQPUT1 MQRC_MULTIPLE_REASONS ist

PCF-Parameter:	MQIACF_REASON_CODE
Tracestufe:	2

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

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

IntAttrs

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.

CharAttrs

Beschreibung:	Die Zeichenattribute, die durch die festgelegte Operation aktualisiert werden sollen. Die Werte werden miteinander verkettet.
PCF-Parameter:	MQCACF_CHAR_ATTRS
Tracestufe:	3

Typ: MQCFST
Bereich: Dieser Parameter ist nur enthalten, wenn 'CharAttrLength' größer als 0 ist.

QMGrOpDuration

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.
Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64_QMGR_OP_DURATION
Tracestufe: 2
Typ: MQCFIN64

MQSUB

Die Anwendung hat die MQSUB-MQI-Funktion gestartet.

CompCode

Beschreibung: Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter: MQIACF_COMP_CODE
Tracestufe: 1
Typ: MQCFIN

Reason

Beschreibung: Das Ursachencode-Ergebnis der Operation.
PCF-Parameter: MQIACF_REASON_CODE
Tracestufe: 1
Typ: MQCFIN

SubHandle

Beschreibung: Die Subskriptionskennung
PCF-Parameter: MQIACF_HSUB
Tracestufe: 1
Typ: MQCFIN

ObjectHandle

Beschreibung: Die Objektkennung
PCF-Parameter: MQIACF_HOBJ
Tracestufe: 1
Typ: MQCFIN

Options

Beschreibung: Subskriptionsoptionen

PCF-Parameter: MQIACF_SUB_OPTIONS
Tracestufe: 1
Typ: MQCFIN

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.

QMgrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbraucht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde. Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ	MQCFIN64

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

QMGrOpDuration

Beschreibung:	Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers. Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde. Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.
PCF-Parameter:	MQIAMO64_QMGR_OP_DURATION
Tracestufe:	2
Typ:	MQCFIN64

MQSTAT

Die Anwendung hat die MQSTAT-MQI-Funktion gestartet.

CompCode

Beschreibung:	Der Beendigungscode, der das Ergebnis der Operation angibt.
PCF-Parameter:	MQIACF_COMP_CODE
Tracestufe:	1

Typ: MQCFIN

Reason

Beschreibung: Das Ursachencode-Ergebnis der Operation.

PCF-Parameter: MQIACF_REASON_CODE

Tracestufe: 1

Typ: MQCFIN

Type

Beschreibung: Typ der angeforderten Statusinformationen

PCF-Parameter: MQIACF_STATUS_TYPE

Tracestufe: 2

Typ: MQCFIN

StatusStructure

Beschreibung: Die MQSTS-Struktur.

PCF-Parameter: MQBACF_MQSTS_STRUCT

Tracestufe: 3

Typ: MQCFBS

Länge: Die Länge in Byte der MQSTS-Struktur (die tatsächliche Größe hängt von der Strukturversion ab)

QMGrOpDuration

Beschreibung: Ungefähre API-Aufrufdauer (in Mikrosekunden) innerhalb des Warteschlangenmanagers.

Die Dauer enthält nicht die Zeit, die außerhalb des Warteschlangenmanagers verbracht wurde. Dies ist zum Beispiel die Zeit, die als IBM MQ-Client verwendet wurde.

Anmerkung: Die Genauigkeit dieses Zeitgebers variiert abhängig von der Plattform, die Ihr Unternehmen verwendet.

PCF-Parameter: MQIAMO64_QMGR_OP_DURATION

Tracestufe: 2

Typ: MQCFIN64

Variablenparameter für Anwendungsaktivitäten-XA-Operationen

Bei XA-Operationen handelt es sich um API-Aufrufe, die Anwendungen ermöglichen, um MQ die Teilnahme an einer Transaktion zu ermöglichen. Die Parameter für jede Operation werden im folgenden Abschnitt definiert.

Die Tracestufe gibt die Stufe der Trace-Granularität an, die für die Parameter erforderlich ist, die in den Trace eingeschlossen werden sollen. Folgende Werte für die Tracestufe sind möglich:

1. Niedrig

Der Parameter ist eingeschlossen, wenn die Traceverarbeitung für " low ", " medium " oder " high " für eine Anwendung konfiguriert ist. Diese Einstellung bedeutet, dass ein Parameter immer in der Gruppe AppActivityData für die Operation enthalten ist. Diese Parametergruppe reicht aus, um einen Trace für die MQI-Aufrufe, die eine Anwendung vornimmt, zu verfolgen und zu sehen, ob sie erfolgreich sind.

2. Mittel

Der Parameter ist nur dann in der Gruppe `AppActivityData` für die Operation enthalten, wenn die "mittlere" oder "hohe" Aktivitätstracefunktion 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 Tracefunktion für "hohe" Aktivitäten für eine Anwendung konfiguriert ist. Diese Gruppe von Parametern enthält Speicherauszüge der Strukturen, die an die MQI- und XA-Funktionen übergeben werden. Aus diesem Grund enthält er weitere Informationen zu den Parametern, die in MQI- und XA-Aufrufen verwendet werden. Die Speicherauszüge des Strukturspeichers sind flache Kopien der Strukturen. Um fehlerhafte Versuche zu vermeiden, Zeiger zu dereferenzieren, werden die Zeigerwerte in den Strukturen auf NULL gesetzt.

Anmerkung: Die Version der Struktur, für die ein Speicherauszug erstellt wird, ist nicht notwendigerweise identisch mit der Version, die von einer Anwendung verwendet wird. Die Struktur kann durch einen API-Überkreuzungsexit, durch den Aktivitäts-Trace-Code oder durch den Warteschlangenmanager geändert werden. Ein Warteschlangenmanager kann eine Struktur in einer späteren Version ändern, aber der WS-Manager ändert ihn nie in eine frühere Version der Struktur. Dazu würde das Risiko eines Datenverlustes gefährdet.

AXREG

Anwendung hat die AXREG-AX-Funktion gestartet

XID

Beschreibung:	Die XID-Struktur
PCF-Parameter:	MQBACF_XA_XID
Tracestufe:	1
Typ:	MQCFBS
Länge:	Sizeof (XID)

Rmid

Beschreibung:	Ressourcenmanager-ID
PCF-Parameter:	MQIACF_XA_RMID
Tracestufe:	1
Typ:	MQCFIN

Flags

Beschreibung:	Flaggen
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: Flaggen
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: Flaggen
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: Flaggen
PCF-Parameter: MQIACF_XA_FLAGS
Tracestufe: 1
Typ: MQCFIN

XARetCode

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

XACOMplete

Die Anwendung hat die Funktion XACOMplete AX gestartet.

Handle

Beschreibung: Operation für asynchrone Verarbeitung handhaben
PCF-Parameter: MQIACF_XA_HANDLE
Tracestufe: 1
Typ: MQCFIN

Retval

Beschreibung: Rückgabewert der asynchronen Funktion
PCF-Parameter: MQIACF_XA_RETVAL
Tracestufe: 1
Typ: MQCFINMQCFBS

Rmid

Beschreibung: Ressourcenmanager-ID
PCF-Parameter: MQIACF_XA_RMID
Tracestufe: 1
Typ: MQCFIN

Flags

Beschreibung: Flaggen
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: Flaggen
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: Flaggen
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: Flaggen
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: Flaggen
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: Flaggen
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: Flaggen
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:	Flaggen
PCF-Parameter:	MQIACF_XA_FLAGS
Tracestufe:	1
Typ:	MQCFIN

XARetCode

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

Multi

Systemthemen für Überwachung und Aktivitätstrace

Systemthemen in Themenstrukturen des Warteschlangenmanagers werden für die Ressourcenüberwachung (einige davon ähneln dem Inhalt von Statistiknachrichten) und als Methode zur Verarbeitung von Anwendungsaktivitätstraces verwendet.

\$SYS/MQ-Zweig der Themenstruktur des Warteschlangenmanagers

Die Themenstruktur jedes Warteschlangenmanagers enthält die Verzweigung \$SYS/MQ . Der Warteschlangenmanager veröffentlicht die Themenzeichenfolgen in dieser Verzweigung. Ein berechtigter Benutzer kann diese Themenzeichenfolgen subscribieren, um Informationen über den Warteschlangenmanager und die Aktivität auf ihm zu erhalten. Diese Systemthemen werden für den Trace und die Überwachung von Anwendungsaktivitäten verwendet. Weitere Informationen finden Sie unter [Themenbaumstrukturen](#).

Das Stammverzeichnis der \$SYS/MQ-Verzweigung wird durch das Themenobjekt SYSTEM.ADMIN.TOPIC dargestellt. Der Zweig '\$SYS/MQ' der Themenstruktur wird auf die folgenden Arten vom Rest der Themenstruktur isoliert:

- Eine Subskription, die mit Platzhalterzeichen an einem höheren Punkt in der Baumstruktur als \$SYS/MQ erstellt wird, stimmt nicht mit einer Themenzeichenfolge innerhalb der \$SYS/MQ-Verzweigung überein. Die Platzhalteroperation für SYSTEM.ADMIN.TOPIC ist auf " Block " gesetzt und kann nicht geändert werden. Diese Einschränkung gilt auch, wenn Sie Platzhalterzeichen mit dem **runmqsc** -Befehl DISPLAY TPSTATUS verwenden, um Knoten in der Themenstruktur anzuzeigen. Um Themenknoten in der \$SYS/MQ-Verzweigung anzuzeigen, starten Sie die Themenzeichenfolge mit \$SYS/MQ. Verwenden Sie beispielsweise \$SYS/MQ/# , um alle Knoten anzuzeigen.
- Sie müssen berechtigt sein, die Berechtigung zum Verwenden der \$SYS/MQ-Themenstruktur zu erteilen oder tiefer als \$SYS/MQ zu erteilen. Die Berechtigung zum Subscribieren einer Themenzeichenfolge basiert auf der Berechtigung, die für ein verwaltetes Themenobjekt an oder höher als die Themenzeichenfolge in der Themenstruktur erteilt wird. Berechtigungen, die im Stammverzeichnis (SYSTEM.BASE.TOPIC) erteilt werden, würden eine Benutzerberechtigung für alle Themenzeichenfolgen erteilen. Im Fall der \$SYS/MQ-Verzweigung gilt jedoch der Zugriff, der höher als \$SYS/MQ erteilt wurde, nicht auf die \$SYS/MQ-Themenzeichenfolgen.
- Die Verzweigung '\$SYS/MQ' der Themenstruktur wird von den Themenattributen isoliert, die in der Baumstruktur höher gesetzt sind. Die Warteschlange SYSTEM.ADMIN.TOPIC übernimmt keine Attribute aus einem Themenobjekt, das in der Themenstruktur höher definiert ist. Die Änderung der Attribute von SYSTEM.BASE.TOPIC hat zum Beispiel keine Auswirkungen auf das Verhalten der \$SYS/MQ-Verzweigung.

Alle Themenzeichenfolgen, die mit \$SYS/MQ beginnen, sind für die Verwendung durch IBM MQ reserviert. Für diese Themenzeichenfolgen gelten die folgenden Einschränkungen:

- Sie können Multicasting nicht aus der \$SYS/MQ-Verzweigung der Themenstruktur aktivieren.
- Clustering wird für die \$SYS/MQ-Verzweigung nicht unterstützt.
- Der Proxy-Subskriptionsmechanismus kann nicht auf " force " gesetzt werden.
- Anwendungen können nicht in einer \$SYS/MQ-Themenzeichenfolge veröffentlichen.
- Der Standardwert für Veröffentlichungs- und Subskriptionsbereich ist nur der lokale WS-Manager.
- Es können keine Platzhalterzeichen an den folgenden Punkten verwendet werden:
 - \$SYS/MQ/
 - \$SYS/MQ/INFO
 - \$SYS/MQ/INFO/QMGR
 - \$SYS/MQ/INFO/QMGR/ *queue_manager_name*
 - \$SYS/MQ/INFO/QMGR/ *queue_manager_name* /ActivityTrace

Versuche, an diesen Punkten Platzhalterzeichen zu verwenden, führen zu einem Subskriptionsfehler mit der Ursache MQRC_ADMIN_TOPIC_STRING_ERROR.

Leistungsüberwachungs-API, die über Publish/Subscribe verfügbar ist

Sie können Warteschlangenmanagerstatistiken überwachen, indem Sie den einfachen Publish/Subscribe-Mechanismus und entweder die amqsrua-Beispielanwendung oder Ihre eigene Anwendung verwenden. Statistikdaten werden in einem Systemthema unter \$SYS/MQ/INFO/QMGR veröffentlicht, das Benutzer bei der Überwachung von Ressourcen unterstützt. Diese Statistikdaten können entweder durch Ausführen der Beispielanwendung 'amqsrua' oder durch Schreiben einer Anwendung angezeigt werden, die das Thema des Ressourcenüberwachungssystems ähnlich wie **amqsrua**subskribiert.

Zugehörige Konzepte

„Anwendungsaktivitätstrace“ auf Seite 212

Der Aktivitätstrace der Anwendung erzeugt detaillierte Informationen zum Verhalten von Anwendungen, die mit einem Warteschlangenmanager verbunden sind. Dabei wird das Verhalten einer Anwendung verfolgt und es wird eine detaillierte Ansicht der Parameter bereitgestellt, die von einer Anwendung während der Interaktion mit IBM MQ verwendet werden. Es zeigt auch die Reihenfolge der MQI-Aufrufe, die von einer Anwendung ausgegeben werden.

Systemressourcennutzung mit dem Befehl amqsrua überwachen

Sie können den Befehl **amqsrua** verwenden, um Leistungsdaten abzufragen, die sich auf die Systemressourcennutzung eines Warteschlangenmanagers beziehen.

Informationen zu diesem Vorgang

Die Beispielanwendung 'amqsrua' zeigt eine Möglichkeit, wie Veröffentlichungen zur IBM MQ-Überwachung gelesen und die von Warteschlangenmanagern veröffentlichten Leistungsdaten angezeigt werden können. Diese Daten können Informationen über die CPU-, Speicher- und Plattenbelegung enthalten. Sie können auch Daten anzeigen, die äquivalent zu den STATMQI-PCF-Statistikdaten sind. Die Daten werden alle 10 Sekunden veröffentlicht und werden während der Ausführung des Befehls gemeldet.

Sie können den Befehl nur mit dem Namen des Warteschlangenmanagers ausführen und interaktiv durchgehen, um die Parameter **CLASS**, dann **TYPE** und dann **object** auszuwählen, die für den Warteschlangenmanager bei jedem Schritt verfügbar sind. Wenn Sie die **CLASS**-, **TYPE**- und **object**-Namen kennen, für die Sie Informationen anzeigen wollen, können Sie sie angeben, wenn Sie den Befehl **amqsrua** ausführen.

Standardmäßig sucht die Anwendung 'amqsrua' nach Statistikdaten, die vom WS-Manager unter der Themenstruktur \$SYS/MQ/INFO/QMGR veröffentlicht werden. Andere Komponenten oder Anwendungen können einen ähnlichen Mechanismus verwenden, um unter einem anderen Themenstartpunkt zu ver-

öffentlichen. Ab IBM MQ 9.1.0 können Sie mit dem Parameter **-p** angeben, wo amqsrua nach den Statistikdaten für diese anderen Komponenten sowohl in Linux als auch in Windows sucht.

-m

Warteschlangenmanagername. Der WS-Manager muss aktiv sein. Wenn Sie keinen Warteschlangenmanagernamen angeben, werden die Statistiken für den Standardwarteschlangenmanager angezeigt.

-c

CLASS-Name. Veröffentlichungen zur IBM MQ-Ressourcennutzung sind einer Klasse zugeordnet. Die Klassen stellen die oberste Ebene in der Baumstruktur der Metadaten dar, die die verfügbaren Informationen zur Ressourcennutzung beschreibt.

CPU

Gibt Informationen zur CPU-Belastung zurück.

PLATTE

Gibt Informationen zur Plattenbelegung zurück.

STATMQI

Gibt Informationen zur MQI-Verwendung zurück.

STATQ

Gibt Informationen zur MQI-Verwendung pro Warteschlange zurück.

STATAPP

Gibt Informationen zu Nutzungsstatistiken für die angegebene Anwendung zurück.

Im Abschnitt „[Eigenes Ressourcenüberwachungsanwendung entwickeln](#)“ auf Seite 305 finden Sie Informationen zur Angabe der Themenstruktur für Metadaten und unter [Anwendungsnamen in unterstützten Programmiersprachen verwenden](#) finden Sie Informationen zu den Zeichen, die Sie in den Namen von Anwendungen verwenden können.

Anmerkung: Mit der Option **-o** geben Sie den Anwendungsnamen an, der für STATAPP überwacht wird, und den Warteschlangenamen für STATQ.

-t

TYPE name. Veröffentlichungen zur IBM MQ-Ressourcennutzung sind einem Typ innerhalb einer Klasse zugeordnet. Jede Veröffentlichung enthält die Klasse und den Typ, die es ermöglichen, die class/type/element-Definitionen zu finden und die resultierenden Veröffentlichungen zu bearbeiten. Die class/type/element-Beschreibungen werden als Metadaten beim Start des Warteschlangenmanagers veröffentlicht.

-o

Objektname Bei den Veröffentlichungen zur Ressourcennutzung handelt es sich um PCF-Nachrichten, die aus einer Folge von PCF-Elementen bestehen. Die PCF-Elemente, die für jedes Paar vom Typ/Typ veröffentlicht werden, werden in den Metadaten angezeigt. Ein Blatt wird in der Baumstruktur gespeichert, in der jedes Element beschrieben wird, wodurch die Elemente verarbeitet werden können.

-p

Metadatenpräfix. Geben Sie einen Ausgangspunkt für die Themenstruktur an, an dem **amqsrua** nach Statistikdaten suchen kann, die von Warteschlangenmanagern veröffentlicht werden. Die Standardtopicbaumstruktur ist \$SYS/MQ/INFO/QMGR, aber andere Komponenten oder Anwendungen veröffentlichen möglicherweise Statistikdaten unter einem anderen Ausgangspunkt der Topicbaumstruktur.

-n

Veröffentlichungszähler. Sie können angeben, wie viele Berichte zurückgegeben werden sollen, bevor der Befehl beendet wird. Die Daten werden ungefähr alle zehn Sekunden veröffentlicht. Wenn Sie also den Wert 50 eingeben, gibt der Befehl 50 Berichte über 500 Sekunden zurück. Wenn Sie diesen Parameter nicht angeben, wird der Befehl ausgeführt, bis entweder ein Fehler auftritt oder der WS-Manager heruntergefahren wird.

-s

Modellwarteschlange. Geben Sie optional eine zu verwendende Modellwarteschlange an. (Standardmäßig verwendet **amqsrua** das SYSTEM.DEFAULT.MODEL.QUEUE Warteschlange.)

-h

Verwendung

Vorgehensweise

1. Geben Sie im Verzeichnis "samples" den folgenden Befehl aus, um die verfügbaren Daten für den Warteschlangenmanager anzuzeigen:

- **Linux** Unter Linux, `MQ_INSTALLATION_PATH/samp/bin`:

```
./amqsrua -m QMgrName
```

- **Windows** Unter Windows, `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`:

```
amqsrua -m QMgrName
```

Dabei gibt `QMgrName` den Namen des Warteschlangenmanagers an, der abgefragt werden soll. Der WS-Manager muss aktiv sein. Wenn Sie keinen Warteschlangenmanagernamen angeben, wird der Standardwarteschlangenmanager verwendet.

Die folgenden Optionen sind verfügbar:

```
CPU : Platform central processing units
DISK : Platform persistent data stores
STATMQI : API usage statistics
STATQ : API per-queue usage statistics
Enter Class selection
==>
```

2. Geben Sie in der Liste der CLASS-Optionen `STATMQI` ein.

```
==> STATMQI
CONNDISC : MQCONN and MQDISC
OPENCLOSE : MQOPEN and MQCLOSE
INQSET : MQINQ and MQSET
PUT : MQPUT
GET : MQGET
SYNCPOINT : Commit and rollback
SUBSCRIBE : Subscribe
PUBLISH : Publish
Enter Type selection
==>
```

3. Geben Sie in der Liste der TYPE-Optionen `PUT` ein.

```
==>PUT
Publication received PutDate:20170329 PutTime:17045485 Interval:4 minutes,13.978 seconds
Interval total MQPUT/MQPUT1 count 22
Interval total MQPUT/MQPUT1 byte count 25284 100/sec
Non-persistent message MQPUT count 22
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 25284 100/sec
Put persistent messages - byte count 0
MQSTAT count 0

Publication received PutDate:20170329 PutTime:17050485 Interval:10.001 seconds
Interval total MQPUT/MQPUT1 count 1
Interval total MQPUT/MQPUT1 byte count 524 52/sec
Non-persistent message MQPUT count 1
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 524 52/sec
Put persistent messages - byte count 0
MQSTAT count 0
```

Ergebnisse

Sie haben die **amqsrua** -Beispielanwendung interaktiv verwendet, um Statistikdaten anzuzeigen, die Warteschlangenmanager in den Systemthemen unter dem Metadatenpräfix `$SYS/MQ/INFO/QMGR` veröffentlichten.

Anmerkung: Klassen und Typen von Ressourcenveröffentlichungen, die für Warteschlangenmanager verfügbar sind, können abhängig von ihrer Konfiguration, Version und Plattform unterschiedlich sein. Verwenden Sie **amqsrua** interaktiv, um die Klassen, Typen und Elemente zu suchen, die für Ihren speziellen Warteschlangenmanager verfügbar sind.

Nächste Schritte

Informationen zum Entwickeln Ihrer eigenen Überwachungsanwendung finden Sie unter [„Eigenes Ressourcenüberwachungsanwendung entwickeln“](#) auf Seite 305.

Zugehörige Tasks

[Vorbereiten und Ausführen von Beispielprogrammen unter Windows](#)

[Beispielprogramme unter UNIX und Linux vorbereiten und ausführen](#)

Eigenes Ressourcenüberwachungsanwendung entwickeln

Sie können eine eigene Anwendung entwickeln, um Systemressourcen zu überwachen.

Jeder WS-Manager veröffentlicht Ressourcennutzungsdaten in Themen. Diese Daten werden von Subskribenten zu diesen Themen verarbeitet. Wenn ein WS-Manager gestartet wird, veröffentlicht der Warteschlangenmanager eine Gruppe von Nachrichten zu Metathemen. In diesen Nachrichten wird beschrieben, welche Themen zur Ressourcennutzung vom Warteschlangenmanager unterstützt werden, und der Inhalt der Nachrichten, die zu diesen Themen veröffentlicht werden. Verwaltungstools können die Metadaten subskribieren, um zu ermitteln, welche Informationen zur Ressourcennutzung verfügbar sind, und zu welchen Themen und subskribieren Sie die beworbenen Themen.

Die Themenstruktur für die Metadaten hat die folgende Struktur:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/instance]/type]
```

Verwenden Sie die folgende Struktur für die Metadaten der Themenstruktur:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/resourceid]/type]
```

Damit verdeutlichen Sie die Angabe der Ressource, die über einen Anwendungsnamen mit einer Länge von maximal 28 Zeichen verfügt, um eine Verwechslung mit den Attributen des internen Metadatentyps zu vermeiden.

Eine Liste der möglichen Klassen finden Sie im Abschnitt [„Systemressourcennutzung mit dem Befehl amqsrua überwachen“](#) auf Seite 302.

Der Quelltext für die Anwendung 'amqsrua' wird als IBM MQ-Beispiel bereitgestellt. Sie können diese Musteranwendung als Leitfaden für die Erstellung einer eigenen Überwachungsanwendung verwenden. Sie können die Quelle für das Beispiel aus einer IBM MQ-Clientinstallation abrufen. Die Quellendatei heißt `amqsruaa.c` und befindet sich im Verzeichnis mit den Beispielen:

-   AIX and Linux-Plattformen:`MQ_INSTALLATION_PATH/samp/`
-  Windows-Plattformen:`MQ_INSTALLATION_PATH\tools\c\Samples\`

Das Programm 'amqsrua' subskribiert die Themen zur IBM MQ-Ressourcennutzung und formatiert die resultierenden veröffentlichten PCF-Daten. Die Anwendungsquelle stellt ein grundlegendes Beispiel dafür dar, wie diese Art von Verwaltungsdaten subskribieren und verarbeitet werden kann. Die Anwendung 'amqsrua' führt die folgenden Tasks aus:

- Erstellt eine nicht permanente Subskription für die Themen, die durch die Eingabeparameter angegeben werden.

- Ruft MQGET wiederholt auf, um Nachrichten aus den Themen abzurufen, und schreibt in stdout.
- Schreibt eine Nachricht für jeden MQI-Ursachencode (außer MQRC_NONE).
- Stoppt, wenn ein MQI-Beendigungscode von MQCC_FAILED vorhanden ist, oder wenn die angeforderte Anzahl von Veröffentlichungen zur Ressourcennutzung verbraucht wurde.

Zu den Systemthemen veröffentlichte Metriken

Metriken werden in Klassen und in Unterkategorien in Typen eingeteilt. Es gibt verschiedene Metriken, die unter jeder Metrikkategorie und jedem Metriktyp veröffentlicht werden.

Index

- „[CPU \(Zentrale Plattform-Verarbeitungseinheiten\)](#)“ auf Seite 306
- „[DISK \(persistente Datenspeicher der Plattform\)](#)“ auf Seite 307
- „[STATMQI \(API-Nutzungsstatistiken\)](#)“ auf Seite 308
- „[STATQ \(API-Nutzungsstatistik pro Warteschlange\)](#)“ auf Seite 310
- „[STATAPP \(Nutzungsstatistiken pro Anwendung\)](#)“ auf Seite 311
- „[NHAREPLICA \(native HA-Statistik pro Instanz\)](#)“ auf Seite 312

  Informationen zur Erfassung von Daten für die aufgelisteten Optionen mit Ausnahme von NHAREPLICA finden Sie in [„Systemressourcennutzung mit dem Befehl amqsrua überwachen“](#) auf Seite 302.

Sie können auch den Befehl ALTER QMGR verwenden, um STATMQI und STATQ auf Warteschlangenmanagerebene zu überwachen, oder das lokale Warteschlangenattribut STATQ, um einzelne Warteschlangen zu überwachen. Informationen zu dieser Option finden Sie unter [ALTER QUEUES](#).

CPU (Zentrale Plattform-Verarbeitungseinheiten)

Einführung

Wenn Statistikdaten auf das *aktuelle Intervall* verweisen, ist dies das Intervall, das in der veröffentlichten Nachricht durch den Parameter MQIAMO64_MONITOR_INTERVAL definiert ist.

Statistikdaten werden in der Regel alle 10 Sekunden veröffentlicht, das *Veröffentlichungsintervall*, solange mindestens ein aktiver Subskribent vorhanden ist. Das genaue Intervall sollte jedoch immer der Nachricht entnommen werden.

Wichtig: Sofern nicht anders angegeben, sind Metriken ansonsten absolute Werte für den Zeitpunkt, zu dem sie erfasst wurden.

SystemSummary (CPU-Leistung - plattformübergreifend)

Benutzer-CPU-Zeit in Prozent *X* %

Der durchschnittliche Prozentsatz der Zeit (belegt während des letzten 10-Sekunden-Intervalls), die von der CPU belegt wurde, als sie sich im nicht privilegierten Code befand.

System-CPU-Zeit in Prozent *X* %

Der durchschnittliche Prozentsatz der Zeit (im letzten 10-Sekunden-Intervall), die von der CPU belegt wurde, als sie sich im privilegierten Code befand.

CPU-Belastung - einminütiger Durchschnitt *X*

Die durchschnittliche Last in einer Minute. "Lastdurchschnitt" ist ein branchenweiter Begriff, aber der genaue gemeldete Wert kann plattformübergreifend abweichen.

CPU-Belastung - Fünf-Minuten-Durchschnitt *X*

Die durchschnittliche Last von fünf Minuten. "Lastdurchschnitt" ist ein branchenweiter Begriff, aber der genaue gemeldete Wert kann plattformübergreifend abweichen.

CPU-Belastung - Durchschnitt über fünfzehn Minuten X

Die durchschnittliche Last in 15 Minuten. "Lastdurchschnitt" ist ein branchenweiter Begriff, aber der genaue gemeldete Wert kann plattformübergreifend abweichen.

CPU-Systemzusammenfassung

RAM freier Prozentsatz X %

RAM-Gesamtbytes X MB

RAM freier Prozentsatz X %

RAM-Gesamtbytes X MB

QMGrSummary (CPU-Leistung - laufender Warteschlangenmanager)

Benutzer-CPU-Zeit - prozentuale Schätzung für Warteschlangenmanager X %

Der durchschnittliche Prozentsatz der Zeit (im Intervall der letzten 10 Sekunden), die von der CPU belegt wurde, als die Prozesse dieses Warteschlangenmanagers nicht privilegierten Code hatten.

System-CPU-Zeit - prozentuale Schätzung für Warteschlangenmanager X %

Der durchschnittliche Prozentsatz der Zeit (im Intervall der letzten 10 Sekunden), die von der CPU belegt wurde, als sich die Prozesse dieses Warteschlangenmanagers in privilegiertem Code befanden.

RAM-Gesamtbytes - Schätzung für Warteschlangenmanager X MB

Dies ist ein Näherungswert für den vom Warteschlangenmanager belegten Speicher.

DISK (persistente Datenspeicher der Plattform)

Die *SystemSummary* und *QMGrSummary* sind absolute Werte zum Zeitpunkt der Erfassung. Details zum *Veröffentlichungsintervall* finden Sie in der [Einführung](#).

SystemSummary (Plattenbelegung - plattformübergreifend)

MQ-Fehler Dateisystem - verwendete Bytes X MB

MQ-Fehler Dateisystem - freier Speicherbereich X %

Anzahl der MQ-FDC-Dateien X

MQ-Trace-Dateisystem - verwendete Bytes X MB

MQ-Trace-Dateisystem - freier Speicherbereich X %

QMGrSummary (Plattenbelegung - laufende Warteschlangenmanager)

Warteschlangenmanager-Dateisystem - verwendete Bytes X MB

Warteschlangenmanager-Dateisystem - freier Speicherbereich X %

Protokoll (Plattenbelegung - Warteschlangenmanager-Wiederherstellungsprotokoll)

Log - verwendete Bytes X

Protokoll - Bytes max. X

Die maximale Anzahl Byte, die in das Protokoll geschrieben werden können, wenn alle primären und sekundären Speicherbereiche voll waren Dies ist kleiner als die Größe des Protokolldateisystems.

Protokolldateisystem - verwendete Bytes X

Protokolldateisystem - Bytes max. X

Protokoll-Physische Byte, die für das aktuelle Intervall X geschrieben wurden

Die Definition des *aktuellen Intervalls* finden Sie in der [Einführung](#).

Protokoll-Logische Byte geschrieben für das *aktuelle Intervall* X

Log - Latenzzeit für Schreiboperationen X uSek

Ein gleitender Durchschnitt, der die Zeit darstellt, die ein einzelner Schreibvorgang auf Platte benötigt.

Dabei ist **LogWriteIntegrity=TripleWrite** die physische Anzahl der auf die Platte geschriebenen Byte größer als die Anzahl der geschriebenen logischen Byte.

Protokollschreibgröße X , auch rollierender Durchschnitt.

Protokollbelegung durch Bereiche, die auf die Archivierung warten X .

Nur veröffentlicht, wenn **logtype=linear** und **LogManagement=archive**. Weitere Informationen finden Sie unter [Zeilengruppe 'Log'](#) der Datei `qm.ini`.

Protokollspeicherbereich in MB, der für die Datenträgerwiederherstellung erforderlich ist X

Nur veröffentlicht, wenn **logtype=linear**.

Protokollspeicherbereich in MB, belegt durch wiederverwendbare Speicherbereiche X

Nur veröffentlicht, wenn **logtype=linear** und **LogManagement=automatic**. Weitere Informationen finden Sie unter [Zeilengruppe 'Log'](#) der Datei `qm.ini`.

Protokollaktueller primärer Speicherbereich im Gebrauch $X\%$.

Belegter Protokolldateispeicher als Prozentsatz der primären Protokolle. Dieser Wert kann größer als 100% sein.

Protokoll-Workload-Auslastung des primären Speicherbereichs $X\%$

Der Prozentsatz des belegten Protokolldateispeicherplatzes im Verlauf des letzten Protokolls als rollierender Durchschnitt.

STATMQI (API-Nutzungsstatistiken)

Alle API-Nutzungsstatistiken spiegeln Vorkommen und/oder Prozentsätze für das veröffentlichte Intervall wider. Die Definition des *Veröffentlichungsintervalls* finden Sie in der [Einführung](#).

Die Statistik gibt einen Zähler für die Anzahl der fehlgeschlagenen MQI-Aufrufe aus, aber nicht für jeden fehlgeschlagenen MQI-Aufruf wird in diesen Statistikdaten angezeigt. Tatsächlich werden die Statistikdaten nicht für jeden MQI-Aufruf aufgezeichnet. Dies liegt daran, dass viele Gründe für das Fehlschlagen von MQI-Aufrufen diagnostiziert werden, bevor der MQI-Aufruf die internen Daten des Warteschlangenmanagers erreicht hat, in dem die Statistikdaten aufgezeichnet werden.

Ein Beispiel hierfür ist `MQRC_HCONN_ERROR`, das an eine Clientanwendung zurückgegeben wurde. Wenn eine Clientanwendung einen fehlerhaften **hconn** übergibt, diagnostiziert der MQ-Client diesen Fehler und gibt `MQRC_HCONN_ERROR` zurück, ohne den MQI-Aufruf an den Warteschlangenmanager zu übergeben. Daher wird der fehlgeschlagene MQI-Aufruf nie in den vom Warteschlangenmanager aufgezeichneten Statistiken angezeigt.

Statistiken zu fehlgeschlagenen MQI-Aufrufen sind nützlich, weil sie es Kunden ermöglichen, schlecht geschriebene Anwendungen zu beheben, die unnötige fehlgeschlagene MQI-Aufrufe generieren, was sich auf die Leistung auswirkt. Beispiele für Fehlerursachen für verschiedene MQI-Aufrufe, die in den Statistiken aufgezeichnet werden:

- `MQCONN/MQCONN/MQOPEN` gibt 2035 `MQRC_NOT_AUTHORIZED` bei Diagnose durch den Warteschlangenmanager und nicht durch den Client zurück. Führen Sie beispielsweise **amqsput** als 'nobody' aus.
- `MQPUT/MQPUT1` gibt 2053 `MQRC_Q_FULL` zurück, da `MAXDEPTH` überschritten wurde.
- `MQGET` gibt 2033 `MQRC_NO_MSG_AVAILABLE` zurück, wenn eine leere Warteschlange durchsucht oder destruktiv abgerufen wird
- `MQSUBRQ` gibt 2437 `MQRC_NO_RETAINED_MSG` zurück, da keine aufbewahrte Nachricht vorhanden ist

CONNDISC (MQCONN und MQDISC)

`MQCONN/MQCONN`-Zähler X

Failed `MQCONN/MQCONN`-Zähler X

Gleichzeitig bestehende Verbindungen - obere Grenze X

`MQDISC`-Zähler X

OPENCLOSE (MQOPEN und MQCLOSE)

MQOPEN-Zähler X Y/sek
Zähler fehlgeschlagene MQOPEN X
MQCLOSE-Zähler X Y/sek
Zähler fehlgeschlagene MQCLOSE X

INQSET (MQINQ und MQSET)

Zähler MQINQ X
Zähler fehlgeschlagene MQINQ X
MQSET-Zähler X
Zähler fehlgeschlagene MQSET X

PUT (MQPUT)

Intervall Gesamtanzahl MQPUT/MQPUT1 X
Intervall Gesamtanzahl MQPUT/MQPUT1-Bytes X Y/sek
Nicht persistente Nachrichten MQPUT-Zähler X
Persistente Nachrichten MQPUT-Zähler X
Zähler fehlgeschlagene MQPUT X
Nicht persistente Nachrichten MQPUT1-Zähler X
Persistente Nachrichten MQPUT1-Zähler X
Zähler fehlgeschlagene MQPUT1 X
Eingereichte nicht persistente Nachrichten - Byte-Zähler X Y/sek
Eingereichte persistente Nachrichten - Byte-Zähler X
MQSTAT-Zähler X

GET (MQGET)

Intervall Gesamtabrufe mit Löschen - Zähler X
Intervall Gesamtabrufe mit Löschen - Byte-Zähler X Y/sek
Nicht persistente Nachrichten Abruf mit Löschen - Zähler X
Persistente Nachrichten Abruf mit Löschen - Zähler X
Zähler fehlgeschlagene MQGET X
Abgerufene nicht persistente Nachrichten - Byte-Zähler X Y/sek
Abgerufene persistente Nachrichten - Byte-Zähler X
Nicht persistente Nachrichten anzeigen - Zähler X
Anzeige persistenter Nachrichten - Zähler X
Zähler fehlgeschlagene Anzeigen X
Anzeige nicht persistenter Nachrichten - Zähler X Y/sek
Anzeige persistenter Nachrichten - Byte-Zähler X
Zähler abgelaufene Nachrichten X
Zähler gelöschte Warteschlange X
MQCB-Zähler X
Zähler fehlgeschlagene MQCB X
MQCTL-Zähler X

SYNCPOINT (Festschreiben und Rollback)

Commitzählung X
Rollbackzählung X

SUBSCRIBE (subskribieren)

Permanenten Subskriptionszähler erstellen X

Permanenten Subskriptionszähler ändern X
Permanenten Subskriptionszähler fortsetzen X
Nicht permanenten Subskriptionszähler erstellen X
Fehlgeschlagene Erstellung/Änderung/Fortsetzung des Subskriptionszählers X
Permanenten Subskriptionszähler löschen X
Nicht permanenten Subskriptionszähler löschen X
Zähler fehlgeschlagene Subskriptionslöschung X
MQSUBRQ-Zähler X
Zähler fehlgeschlagene MQSUBRQ X
Permanenter Subskribent - obere Grenze X
Permanenter Subskribent - unterer Grenzwert X
Nicht permanenter Subskribent - obere Grenze X
Nicht permanenter Subskribent - unterer Grenzwert X

PUBLISH (veröffentlichen)

Topic MQPUT/MQPUT1 Intervall gesamt X
Intervall Gesamtzahl der eingereichten Topic-Bytes X Y/sek.
Veröffentlicht an Subskribenten - Nachrichtenanzahl X
Veröffentlicht an Subskribenten - Byte-Zähler X
Nicht permanent - Topic MQPUT/MQPUT1-Zähler X
Permanent - Topic MQPUT/MQPUT1-Zähler X
Zähler fehlgeschlagene Topic MQPUT/MQPUT1 X

STATQ (API-Nutzungsstatistik pro Warteschlange)

GENERAL (Allgemein)

abgelaufene Nachrichten X (verschoben von GET für IBM MQ 9.3.0 und höhere CD-Versionen)
Zähler gelöschte Warteschlangen X (verschoben von GET für IBM MQ 9.3.0 und höhere CD-Versionen)
durchschnittliche Zeit in der Warteschlange X uSec (verschoben von GET für IBM MQ 9.3.0 und höhere CD-Versionen)
Warteschlangenlänge X uSec (verschoben von GET für IBM MQ 9.3.0 und höhere CD-Versionen)

OPENCLOSE (MQOPEN und MQCLOSE)

MQOPEN-Zähler X
MQCLOSE-Zähler X

INQSET (MQINQ und MQSET)

Zähler MQINQ X
MQSET-Zähler X

PUT (MQPUT und MQPUT1)

MQPUT/MQPUT1-Zähler X
MQPUT-Byte-Zähler X
MQPUT nicht persistente Nachrichten Zähler X
MQPUT persistente Nachrichten Zähler X
Zähler rückgängig gemachte MQPUT X
MQPUT1 nicht persistente Nachrichten Zähler X
MQPUT persistente Nachrichten Zähler X
Zähler nicht persistente Nachrichten X
Zähler permanente Byte X

Sperrkonflikt X %

Der Prozentsatz der Versuche, die Warteschlange zu sperren, die dazu geführt haben, dass auf die Freigabe der Sperre durch einen anderen Prozess gewartet wurde. Ein abnehmender Sperrkonflikt erhöht wahrscheinlich den maximalen Durchsatz Ihres Systems, da die Verwendung einer Sperre, die momentan nicht gesperrt ist, effizienter ist als das Warten auf die Freigabe einer Sperre.

von der Warteschlange vermiedene Einreihungen X %

Wenn eine Nachricht in eine Warteschlange eingereiht wird, während es einen wartenden Getter gibt, muss die Nachricht möglicherweise nicht in die Warteschlange gestellt werden, da sie möglicherweise sofort an den Getter übergeben werden kann. Diese Nachricht soll also die Warteschlange vermeiden und "Warteschlange vermiedene Einreihungen" ist die Anzahl solcher Nachrichten. Die Erhöhung der Warteschlangenvermeidung erhöht wahrscheinlich den maximalen Durchsatz Ihres Systems, da dadurch die Kosten für das Einreihen der Nachricht in die Warteschlange und das erneute Abrufen der Nachricht aus der Warteschlange vermieden werden.

von der Warteschlange vermiedene Bytes X %

Wenn eine Nachricht in eine Warteschlange eingereiht wird, während es einen wartenden Getter gibt, muss die Nachricht möglicherweise nicht in die Warteschlange gestellt werden, da sie möglicherweise sofort an den Getter übergeben werden kann. Diese Nachricht soll also die Warteschlange vermeiden haben, und "vermiedene Bytes in der Warteschlange" ist die Anzahl solcher Bytes. Die Erhöhung der Warteschlangenvermeidung erhöht wahrscheinlich den maximalen Durchsatz Ihres Systems, da dadurch die Kosten für das Einreihen der Nachricht in die Warteschlange und das erneute Abrufen der Nachricht aus der Warteschlange vermieden werden.

GET (MQGET)

MQGET-Zähler X

MQGET-Bytezähler X

Zähler gelöschte MQGET nicht persistente Nachricht X

Zähler gelöschte MQGET persistente Nachricht X

Zähler rückgängig gemachte MQGET X

Zähler gelöschte MQGET nicht permanente Byte X

Zähler gelöschte MQGET permanente Byte X

Zähler MQGET-Anzeigen nicht persistente Nachricht X

Zähler MQGET-Anzeigen persistente Nachricht X

Zähler MQGET-Anzeigen nicht permanente Byte X

Zähler MQGET-Anzeigen permanente Byte X

abgelaufene Nachrichten X (verschoben in [GENERAL](#) von IBM MQ 9.3)

Anzahl bereinigter Warteschlangen X (verschoben in [GENERAL](#) aus IBM MQ 9.3)

Durchschnittliche Warteschlangenzeit X uSec (von IBM MQ 9.3 in [GENERAL](#) versetzt)

Warteschlangenlänge X (von IBM MQ 9.3 nach [GENERAL](#) verschoben)

gescheiterte gelöschte MQGET X

gescheiterte gelöschte MQGET mit MQRC_NO_MSG_AVAILABLE X

gescheiterte gelöschte MQRC_TRUNCATED_MSG_FAILED X

Gescheiterte MQGET-Anzeigen X

Gescheiterte MQGET-Anzeigen mit MQRC_NO_MSG_AVAILABLE X

Gescheiterte MQGET-Anzeigen mit MQRC_TRUNCATED_MSG_FAILED X

STATAPP (Nutzungsstatistiken pro Anwendung)

INSTANCE (Instanzstatistik)

Instanzanzahl X absolut

Anzahl der verschiebbaren Instanzen X absolut
 Anzahl Instanzunterdeckung X absolut
 Gestartete Instanzen X Intervall
 Initiierte abgehende Instanzverschiebungen X Intervall
 Abgeschlossene abgehende Instanzverschiebungen x Intervall
 Instanzen, die während des Verbindungswiederherstellungsintervalls X beendet wurden
 Beendete Instanzen X Intervall

NHAREPLICA (native HA-Statistik pro Instanz)

REPLICATION (Replikationsstatistik)

Gesendete synchrone Protokollbyte X

Gesendete Catch-up-Protokollbyte X

 Gesendete synchrone komprimierte Protokollbyte X

 Komprimierte Catch-up-Protokollbyte gesendet X

 Gesendete synchrone nicht komprimierte Protokollbyte X

 Gesendete nicht komprimierte Protokollbyte aufholen X

Durchschnittliche Latenzzeit für Protokollschreiboperationen X μ Sec

Durchschnittliche Größe der Protokollschreibbestätigungen X

Aufgelaufene Byte X

Rückstand-Durchschnittliche Byte X

Zugehörige Informationen

Metriken, die bei Verwendung von IBM MQ Operator veröffentlicht werden

Real-time Monitoring

Die Real-time-Überwachung ist eine Technik, mit der Sie den aktuellen Status von Warteschlangen und Kanälen in einem Warteschlangenmanager ermitteln können. Die zurückgegebenen Informationen sind zum Zeitpunkt der Befehlsablaufinformationen korrekt.

Es stehen eine Reihe von Befehlen zur Verfügung, die beim Zurückgeben von Echtzeitinformationen zu Warteschlangen und Kanälen verfügbar sind. Informationen können für eine oder mehrere Warteschlangen oder Kanäle zurückgegeben werden und können in der Menge variieren. Die Echtzeit-Überwachung kann in den folgenden Tasks verwendet werden:

- Systemadministratoren den stabilen Zustand ihres IBM MQ-Systems erläutern. Dies hilft bei der Fehlerdiagnose, wenn ein Problem im System auftritt.
- Ermitteln Sie die Bedingung Ihres Warteschlangenmanagers zu jedem Zeitpunkt, selbst wenn kein bestimmtes Ereignis oder Problem festgestellt wurde.
- Unterstützung bei der Bestimmung der Ursache eines Problems in Ihrem System.

Bei der Echtzeitüberwachung können Informationen für Warteschlangen oder Kanäle zurückgegeben werden. Der Umfang der zurückgegebenen Echtzeitinformationen wird durch die Attribute des Warteschlangenmanagers, der Warteschlange und des Kanals gesteuert.

- Sie überwachen eine Warteschlange, indem Sie Befehle absetzen, um sicherzustellen, dass die Warteschlange ordnungsgemäß verarbeitet wird. Bevor Sie einige der Warteschlangenattribute verwenden können, müssen Sie sie für die Echtzeitüberwachung aktivieren.
- Sie können einen Kanal überwachen, indem Sie Befehle absetzen, um sicherzustellen, dass der Kanal ordnungsgemäß ausgeführt wird. Bevor Sie einige der Kanalattribute verwenden können, müssen Sie sie für die Echtzeitüberwachung aktivieren.

Die Echtzeit-Überwachung für Warteschlangen und Kanäle erfolgt zusätzlich zu und getrennt von der Leistungsüberwachung und der Kanalereignisüberwachung.

Attribute zur Steuerung der Echtzeitüberwachung

Einige Warteschlangen- und Kanalstatusattribute enthalten Überwachungsdaten, wenn die Echtzeitüberwachung aktiviert ist. Wenn die Echtzeitüberwachung nicht aktiviert ist, werden in diesen Überwachungsattributen keine Überwachungsdaten festgehalten. In den Beispielen wird gezeigt, wie Sie diese Warteschlangen- und Kanalstatusattribute verwenden können.

Sie können die Echtzeitüberwachung für einzelne Warteschlangen oder Kanäle oder für mehrere Warteschlangen oder Kanäle aktivieren oder inaktivieren. Wenn Sie einzelne Warteschlangen oder Kanäle steuern möchten, setzen Sie das Warteschlangenattribut MONQ oder das Kanalattribut MONCHL, um die Echtzeitüberwachung zu aktivieren oder zu inaktivieren. Um viele Warteschlangen oder Kanäle gemeinsam zu steuern, aktivieren oder inaktivieren Sie die Echtzeitüberwachung auf WS-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 IBM MQ-Objekte und weisen daher keine Attribute auf dieselbe Weise wie Kanalobjekte auf. Wenn Sie automatisch definierte Clustersenderkanäle steuern möchten, verwenden Sie das WS-Manager-Attribut MONACLS. Dieses Attribut bestimmt, ob automatisch definierte Clustersenderkanäle innerhalb eines Warteschlangenmanagers für die Kanalüberwachung aktiviert oder inaktiviert sind.

Für die Echtzeitüberwachung von Kanälen können Sie das Attribut MONCHL auf eine der drei Überwachungsstufen setzen: niedrig, mittel oder hoch. Sie können die Überwachungsstufe entweder auf der Objektebene oder auf der Ebene des Warteschlangenmanagers festlegen. Die Auswahl der Ebene hängt von Ihrem System ab. Für die Erfassung von Überwachungsdaten sind möglicherweise einige Anweisungen erforderlich, die relativ teuer sind, wie z. B. die Systemzeit. Um die Auswirkungen der Echtzeitüberwachung zu reduzieren, messen die Optionen "mittel" und "niedrig" eine Stichprobe der Daten in regelmäßigen Intervallen und erfassen nicht die gesamte Zeit. In [Tabelle 30 auf Seite 313](#) 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-Managerebene 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 312

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.

„Using IBM MQ online monitoring“ auf Seite 334

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the MONQ, MONCHL, and MONACLS attributes.

Zugehörige Tasks

„Warteschlangen- und Kanalüberwachungsdaten anzeigen“ auf Seite 315

Wenn Sie die Echtzeitüberwachungsdaten für eine Warteschlange oder einen Kanal anzeigen möchten, verwenden Sie den Befehl IBM MQ Explorer oder den entsprechenden MQSC-Befehl. In einigen Überwachungsfeldern wird ein durch Kommas getrennter Paar von Bezugswerten angezeigt, die Ihnen helfen, die Operation Ihres WS-Managers zu überwachen. In den Beispielen wird gezeigt, wie Überwachungsdaten angezeigt werden können.

[Warteschlangenmanagerattribute anzeigen und ändern](#)

[Überwachung \(MONCHL\)](#)

Warteschlangen-und Kanalüberwachungsdaten anzeigen

Wenn Sie die Echtzeitüberwachungsdaten für eine Warteschlange oder einen Kanal anzeigen möchten, verwenden Sie den Befehl IBM MQ Explorer oder den entsprechenden MQSC-Befehl. In einigen Überwachungsfeldern wird ein durch Kommas getrennter Paar von Bezugswerten angezeigt, die Ihnen helfen, die Operation Ihres WS-Managers zu überwachen. In den Beispielen wird gezeigt, wie Überwachungsdaten angezeigt werden können.

Informationen zu diesem Vorgang

Überwachungsfelder, die ein Paar von Werten anzeigen, die durch ein Komma voneinander getrennt sind, stellen kurz- und langfristige Indikatoren für die Zeit zur Verfügung, die seit der Überwachung für das Objekt oder von dem Zeitpunkt, an dem der Warteschlangenmanager gestartet wurde, aktiviert wurde:

- Der Kurzzeitindikator ist der erste Wert im Paar und wird so berechnet, dass aktuellere Messungen eine höhere Gewichtung erhalten und einen größeren Einfluss auf diesen Wert haben. Dies gibt einen Hinweis auf den jüngsten Trend bei den durchgeführten Messungen.
- Der Langzeitindikator in dem zweiten Wert im Paar und wird so berechnet, dass aktuellere Messungen nicht so hoch gewichtet werden. Dies gibt einen Hinweis auf die längerfristige Aktivität bei der Leistung einer Ressource.

Diese Indikatorwerte sind am nützlichsten, um Änderungen in der Operation Ihres WS-Managers zu erkennen. Dies erfordert die Kenntnis der Zeit, die diese Indikatoren zeigen, wenn sie im normalen Gebrauch ist, um zu erkennen, dass in diesen Zeiten zu erkennen ist. Indem Sie diese Werte regelmäßig erfassen und überprüfen, können Sie Schwankungen in der Operation Ihres Warteschlangenmanagers erkennen. Dies kann auf eine Änderung der Leistung hinweisen.

Gehen Sie wie folgt vor, um Echtzeitüberwachungsdaten zu erhalten:

Vorgehensweise

1. Wenn Sie die Echtzeitüberwachungsdaten für eine Warteschlange anzeigen möchten, verwenden Sie den IBM MQ Explorer oder den MQSC-Befehl `DISPLAY QSTATUS`, mit dem der optionale Parameter `MONITOR` angegeben wird.
2. Wenn Sie die Echtzeitüberwachungsdaten für einen Kanal anzeigen möchten, verwenden Sie den IBM MQ Explorer oder den MQSC-Befehl `DISPLAY CHSTATUS`, mit dem der optionale Parameter `MONITOR` angegeben wird.

Beispiel

In der Warteschlange `Q1` ist das Attribut `MONQ` auf den Standardwert `QMGR` gesetzt, und der Warteschlangenmanager, der Eigner der Warteschlange ist, hat das Attribut `MONQ` auf `MEDIUM` gesetzt. Verwenden Sie den folgenden Befehl, um die Überwachungsfelder anzuzeigen, die für diese Warteschlange erfasst wurden:

```
DISPLAY QSTATUS(Q1) MONITOR
```

Die Überwachungsfelder und die Überwachungsstufe der Warteschlange `Q1` werden wie folgt angezeigt:

```
QSTATUS(Q1)
TYPE(Queue)
MONQ(MEDIUM)
QTIME(11892157,24052785)
MSGAGE(37)
LPUTDATE(2005-03-02)
LPUTTIME(09.52.13)
LGETDATE(2005-03-02)
LGETTIME(09.51.02)
```

Der Senderkanal (`QM1.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.TO.QM2) MONITOR
```

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

```
CHSTATUS(QM1.TO.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 312

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

[ANZEIGEWARTESCHLANGENSTATUS](#)

Warteschlangen überwachen

Verwenden Sie diese Seite, um Tasks anzuzeigen, die Sie bei der Behebung eines Problems mit einer Warteschlange und der Anwendung, die diese Services in Warteschlange stellt, unterstützen. Es stehen verschiedene Überwachungsoptionen zur Verfügung, um das Problem zu ermitteln.

Häufig ist das erste Anzeichen eines Problems mit einer Warteschlange, die gewartet wird, dass die Anzahl der Nachrichten in der Warteschlange (CURDEPTH) zunimmt. Wenn Sie eine Erhöhung zu bestimmten Tageszeiten oder unter bestimmten Workloads erwarten, kann eine steigende Anzahl von Nachrichten nicht auf ein Problem hinweisen. Wenn Sie jedoch keine Erklärung für die steigende Anzahl von Nachrichten haben, können Sie die Ursache möglicherweise untersuchen.

Es kann eine Anwendungswarteschlange vorhanden sein, in der ein Problem mit der Anwendung auftritt, oder eine Übertragungswarteschlange, in der ein Problem mit dem Kanal aufgetreten ist. Es sind zusätzliche Überwachungsoptionen verfügbar, wenn die Anwendung, die die Warteschlange in die Warteschlange stellt, ein Kanal ist.

Die folgenden Beispiele untersuchen Probleme mit einer bestimmten Warteschlange (Q1) und beschreiben die Felder, die in der Ausgabe der verschiedenen Befehle angezeigt werden:

Feststellen, ob Ihre Anwendung die Warteschlange geöffnet hat

Wenn Sie ein Problem mit einer Warteschlange haben, überprüfen Sie, ob Ihre Anwendung die Warteschlange geöffnet hat.

Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um zu ermitteln, ob Ihre Anwendung die Warteschlange geöffnet hat:

Vorgehensweise

1. Stellen Sie sicher, dass die Anwendung, die für die Warteschlange ausgeführt wird, die Anwendung ist, die Sie erwarten. Geben Sie den folgenden Befehl für die Warteschlange aus:

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) ALL
```

Sehen Sie sich in der Ausgabe das Feld APPLTAG an, und überprüfen Sie, ob der Name Ihrer Anwendung angezeigt wird. Wenn der Name Ihrer Anwendung nicht angezeigt wird oder wenn überhaupt keine Ausgabe vorhanden ist, starten Sie Ihre Anwendung.

2. Wenn es sich bei der Warteschlange um eine Übertragungswarteschlange handelt, sehen Sie die Ausgabe im Feld CHANNEL an.

Wenn der Kanalname nicht im Feld CHANNEL angezeigt wird, stellen Sie fest, ob der Kanal aktiv ist.

3. Stellen Sie sicher, dass die Anwendung, die für die Warteschlange ausgeführt wird, die Warteschlange für die Eingabe geöffnet hat. Geben Sie den folgenden Befehl ein:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Sehen Sie sich in der Ausgabe das Feld IPPROCS an, um festzustellen, ob eine Anwendung die Warteschlange für die Eingabe geöffnet hat. Wenn der Wert 0 ist und es sich um eine Benutzeranwendungswarteschlange handelt, stellen Sie sicher, dass die Anwendung die Warteschlange für die Eingabe öffnet, damit die Nachrichten aus der Warteschlange abgerufen werden.

Überprüfen, ob Nachrichten in der Warteschlange verfügbar sind

Wenn Sie eine große Anzahl von Nachrichten in der Warteschlange haben und Ihre Anwendung keine dieser Nachrichten verarbeitet, überprüfen Sie, ob die Nachrichten in der Warteschlange für Ihre Anwendung verfügbar sind.

Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um zu untersuchen, warum Ihre Anwendung keine Nachrichten aus der Warteschlange verarbeitet:

Vorgehensweise

1. Stellen Sie sicher, dass Ihre Anwendung nicht nach einer bestimmten Nachrichten-ID oder Korrelations-ID fragt, wenn sie alle Nachrichten in der Warteschlange verarbeiten soll.
2. Obwohl die aktuelle Länge der Warteschlange zeigen kann, dass eine wachsende Anzahl von Nachrichten in der Warteschlange vorhanden ist, sind einige Nachrichten in der Warteschlange möglicherweise nicht für eine Anwendung verfügbar, da sie nicht festgeschrieben werden. Die aktuelle Tiefe schließt die Anzahl der nicht festgeschriebenen MQPUTs von Nachrichten in die Warteschlange ein. Geben Sie den folgenden Befehl ein:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Sehen Sie sich in der Ausgabe das UNCOM-Feld an, um festzustellen, ob nicht festgeschriebene Nachrichten in der Warteschlange vorhanden sind.

3. Wenn Ihre Anwendung versucht, Nachrichten aus der Warteschlange abzurufen, überprüfen Sie, ob die Putting-Anwendung die Nachrichten ordnungsgemäß festschreibt. Geben Sie den folgenden Befehl aus, um die Namen der Anwendungen zu ermitteln, die Nachrichten in diese Warteschlange einreihen:

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) OPENTYPE(OUTPUT)
```

4. Geben Sie dann den folgenden Befehl aus und setzen Sie in *appltag* den Wert APPLTAG aus der Ausgabe des vorherigen Befehls ein:

```
DISPLAY CONN(*) WHERE(APPLTAG EQ appltag) UOWSTDA UOWSTTI
```

Dies zeigt, wann die Arbeitseinheit gestartet wurde, und hilft Ihnen dabei zu ermitteln, ob die Anwendung eine lange Arbeitseinheit erstellt. Wenn es sich bei der Putting-Anwendung um einen Kanal handelt, möchten Sie möglicherweise untersuchen, warum eine Stapelverarbeitung eine lange Zeit in Anspruch nimmt.

Überprüfen, ob die Anwendung Nachrichten aus der Warteschlange erhält

Wenn Sie ein Problem mit einer Warteschlange und der Anwendung, die diese Warteschlange enthält, haben, überprüfen Sie, ob Ihre Anwendung Nachrichten aus der Warteschlange abstellt.

Informationen zu diesem Vorgang

Führen Sie die folgenden Prüfungen aus, um zu überprüfen, ob Ihre Anwendung Nachrichten aus der Warteschlange erhält:

Vorgehensweise

1. Stellen Sie sicher, dass die Anwendung, die für die Warteschlange ausgeführt wird, tatsächlich Nachrichten aus der Warteschlange verarbeitet. Geben Sie den folgenden Befehl ein:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

In der Ausgabe die Felder LGETDATE und LGETTIME anzeigen, die anzeigen, wann der letzte Versuch aus der Warteschlange ausgeführt wurde.

2. Wenn das letzte Abrufen aus dieser Warteschlange zurückliegt als erwartet, stellen Sie sicher, dass die Anwendung Nachrichten ordnungsgemäß verarbeitet.

Wenn es sich bei der Anwendung um einen Kanal handelt, überprüfen Sie, ob die Nachrichten durch diesen Kanal verschoben werden.

Feststellen, ob die Anwendung Nachrichten schnell genug verarbeiten kann

Wenn Nachrichten in der Warteschlange erstellt werden, die anderen Prüfungen jedoch keine Verarbeitungsfehler gefunden haben, überprüfen Sie, ob die Anwendung Nachrichten schnell genug verarbeiten kann. Wenn es sich bei der Anwendung um einen Kanal handelt, überprüfen Sie, ob der Kanal Nachrichten schnell genug verarbeiten kann.

Informationen zu diesem Vorgang

Führen Sie die folgenden Tests aus, um festzustellen, ob die Anwendung Nachrichten schnell genug verarbeitet:

Vorgehensweise

1. Geben Sie den folgenden Befehl in regelmäßigen Abständen aus, um Leistungsdaten zu der Warteschlange zu erfassen:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Wenn die Werte in den QTIME-Indikatoren hoch sind oder sich im Laufe des Zeitraums erhöhen und Sie bereits die Möglichkeit von Arbeitseinheiten mit langer Laufzeit ausgeschlossen haben, indem Sie überprüfen, ob Nachrichten in der Warteschlange verfügbar sind, wird die Anwendung unter Umständen nicht mit den Putting-Anwendungen in Einklang stehen.

2. Wenn Ihre Anwendung nicht mit den Eintragsanwendungen Schritt halten kann, fügen Sie eine weitere Anwendung zum Verarbeiten der Warteschlange hinzu.

Ob Sie eine weitere 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)
```

 Unter IBM MQ for z/OS können Sie ermitteln, ob ein Benutzer einen Kanal mithilfe von Befehlsereignisnachrichten stoppt.

6. Optional: Wenn der Wert des Felds STATUS RETRY lautet, führen Sie die folgenden Schritte aus:

- a) Überprüfen Sie die Fehlerprotokolle, um den Fehler zu identifizieren, und beheben Sie das Problem.

- b) Geben Sie den folgenden Befehl aus, um den Kanal erneut manuell zu starten:

```
START CHANNEL(QM1.TO.QM2)
```

oder Warten Sie, bis der Kanal bei der nächsten Wiederholung erfolgreich eine Verbindung hergestellt hat.

7. Optional: Wenn der Wert des Felds STATUS BINDING oder REQUESTING ist, ist der Kanal noch nicht erfolgreich mit dem Partner verbunden. Führen Sie die folgenden Schritte aus:

- a) Geben Sie den folgenden Befehl an beiden Enden des Kanals aus, um den Unterstatus des Kanals zu bestimmen:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

Anmerkung:

- i) In einigen Fällen kann es zu einem Substatus nur an einem Ende des Kanals kommen.

ii) Viele Unterzustände sind vorübergehend, daher setzen Sie den Befehl ein paar Mal ab, um festzustellen, ob ein Kanal in einem bestimmten Unterzustand blockiert ist.

b) Überprüfen Sie [Tabelle 31 auf Seite 321](#) , um festzustellen, welche Aktion zu ergreifen ist:

<i>Tabelle 31. 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 324.
	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 324.
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 324.
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 TLS-Handshake besteht aus einer Reihe von gesendeten und empfangenen Nachrichten. Wenn die Netzzeiten langsam sind oder die Verbindung zu den Lookup-CRLs langsam ist, wirkt sich dies auf die Zeit aus, die für den Handshake zur Hand genommen wird.  Unter IBM MQ for z/OS kann dieser Unterzustand auch anzeigen, dass der Wert von SSLTASKS nicht ausreicht.

Anmerkungen:

- i) Der einleitende MCA ist das Ende des Kanals, der den Dialog gestartet hat. Dabei kann es sich um Absender, Cluster-Absender, vollständig qualifizierte Server und Anforderer handeln. In einem Server-Requester-Paar ist es das Ende, von dem aus Sie den Kanal gestartet haben.
- ii) Der antwortende MCA ist das Ende des Kanals, das auf die Anforderung zum Starten des Datenaustauschs geantwortet hat. Dies kann Empfänger, Cluster-Empfänger, Requester (wenn der Server oder Sender gestartet wird), Server (wenn der Requester gestartet wird) und Absender (in einem Requester-Sender-Callback-Paar von Kanälen).

Überprüfen, ob der Kanal Nachrichten bewegt

Wenn Sie ein Problem mit einer Übertragungswarteschlange haben, überprüfen Sie, ob der Kanal Nachrichten bewegt.

Vorbereitende Schritte

Geben Sie den Befehl `DIS CHSTATUS(QM1.TO.QM2) ALL` aus. Wenn der Wert des Felds `STATUS` `RUNNING` ist, hat der Kanal erfolgreich eine Verbindung zum Partnersystem hergestellt.

Stellen Sie sicher, dass keine nicht festgeschriebenen Nachrichten in der Übertragungswarteschlange vorhanden sind, wie in [„Überprüfen, ob Nachrichten in der Warteschlange verfügbar sind“](#) auf Seite 317 beschrieben.

Informationen zu diesem Vorgang

Wenn es Nachrichten gibt, die für den Kanal verfügbar sind, um zu senden und zu empfangen, führen Sie die folgenden Prüfungen durch:

Vorgehensweise

1. Zeigen Sie in der Ausgabe des Statusbefehls für den Anzeigekanal (`DIS CHSTATUS(QM1.TO.QM2) ALL`) die folgenden Felder an:

MSGS

Die Anzahl der gesendeten oder empfangenen Nachrichten (oder für Serververbindungskanäle die Anzahl der verarbeiteten MQI-Aufrufe) während dieser Sitzung (seit dem Kanalstart).

BUFSENT

Anzahl der gesendeten Übertragungspuffer. Dazu gehören auch Übertragungen reiner Steuerinformationen.

BYTSENT

Gibt die Anzahl Bytes an, die in der aktuellen Sitzung (seit dem Kanalstart) bereits gesendet wurden. Dazu gehören auch Steuerinformationen, die von dem Nachrichtenkanal gesendet wurden.

LSTMSGDA

Gibt das Datum an, an dem die letzte Nachricht gesendet oder der letzte MQI-Aufruf ausgeführt wurde (siehe `LSTMSGTI`).

LSTMSGTI

Gibt den Zeitpunkt an, zu dem die letzte Nachricht gesendet oder der letzte MQI-Aufruf ausgeführt wurde. Für einen Sender- oder Serverkanal ist dies der Zeitpunkt, zu dem die letzte Nachricht (bzw. im Falle einer Segmentierung das letzte Nachrichtensegment) gesendet wurde. Für einen Requester- oder Empfängerkanal ist dies der Zeitpunkt, zu dem die letzte Nachricht in die entsprechende 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.

STATUS

Der Status des Kanals, der `Starting`, `Binding`, `Initializing`, `Running`, `Stopping`, `Retrying`, `Paused`, `Stopped` oder `Requesting` lauten kann.

SUBSTATE

Die Aktion, die vom Kanal derzeit ausgeführt wird.

INDOUBT

Gibt an, ob der Kanal momentan unbestätigt ist. Der Wert lautet nur dann `YES`, während der sendende Nachrichtenkanalagent auf eine Bestätigung dafür wartet, dass ein Stapel von ihm gesendeter Nachrichten erfolgreich empfangen wurde. Andernfalls wird `NO` zurückgegeben, auch für den Zeitraum, in dem Nachrichten gesendet wurden, jedoch noch keine Empfangsbestätigung angefordert wurde. Für einen empfangenden Kanal ist der Wert stets `NO`.

2. Stellen Sie fest, ob der Kanal seit dem Start alle Nachrichten gesendet hat. Wenn eine Nachricht gesendet wurde, stellen Sie fest, wann die letzte Nachricht gesendet wurde.

3. Der Kanal hat möglicherweise einen Stapel gestartet, der noch nicht abgeschlossen wurde, wie durch einen Wert ungleich null in CURMSGs angezeigt wird. Wenn INDOUBT den Wert YES hat, wartet der Kanal auf die Bestätigung, dass der Stapel am anderen Ende des Kanals empfangen wurde. Zeigen Sie das Feld SUBSTATE der Ausgabe an und lesen Sie die Informationen unter [Tabelle 32 auf Seite 323](#):

<i>Tabelle 32. Sender-und Empfänger-MCA-Unterzustände</i>		
AbsenderSUBSTATE	Empfänger-SUBSTATE	Anmerkungen
MQGET	RECEIVE	Normale Zustände eines Kanals in Ruhe.
SENDEN	RECEIVE	SEND ist in der Regel ein transitorischer Status. Wenn SEND angezeigt wird, zeigt dies an, dass die Kommunikationsprotokollpuffer gefüllt sind. Dies kann auf ein Netzproblem hinweisen.
RECEIVE		Wenn der Sender im Unterzustand RECEIVE für einen beliebigen Zeitraum angezeigt wird, wartet er auf eine Antwort, entweder auf einen Batch-Abschluss oder auf einen Heartbeat. Möglicherweise möchten Sie prüfen, warum eine Stapelverarbeitung lange dauert.

Anmerkung: Es kann auch festgestellt werden, ob der Kanal Nachrichten schnell genug verarbeiten kann, insbesondere wenn der Kanal einen Unterstatus hat, der der Exitverarbeitung zugeordnet ist.

Prüfen, warum ein Stapel lange dauert, bis er abgeschlossen ist

Gründe, warum ein Stapel lange Zeit in Anspruch nehmen kann, um ein langsames Netz zu schließen, oder ein Kanal verwendet die Nachrichtenwiederholungsverarbeitung.

Informationen zu diesem Vorgang

Wenn ein Senderkanal einen Stapel von Nachrichten gesendet hat, wartet er auf die Bestätigung des Stapels vom Empfänger, es sei denn, der Kanal ist pipelinert. Die in dieser Task beschriebenen Faktoren können sich darauf auswirken, wie lange der Senderkanal wartet.

Prozedur

- Überprüfen Sie, ob das Netz langsam ist.
Der NETTIME-Wert ist die Zeit, die in Mikrosekunden angezeigt wird, um ein Ende der Stapelanforderung an das ferne Ende des Kanals zu senden und eine Antwort abzüglich der Zeit zu empfangen, bis das Ende der Stapelanforderung verarbeitet wird. Dieser Wert kann aus einem der folgenden Gründe groß sein:
 - Das Netz ist langsam. Ein langsames Netz kann sich auf die Zeit auswirken, die benötigt wird, um einen Stapel zu beenden. Die Messungen, die zu den Indikatoren für das NETTIME-Feld führen, werden am Ende eines Stapels gemessen. Der erste von einer Verlangsamung im Netz betroffene Stapel wird jedoch nicht mit einer Änderung des NETTIME-Werts angezeigt, da er am Ende des Stapels gemessen wird.
 - Anforderungen werden am fernen Endpunkt in die Warteschlange gestellt. Beispielsweise kann ein Kanal eine Anforderung PUT wiederholen oder eine Anforderung PUT wird möglicherweise aufgrund einer Seitengruppenein-/ausgabe langsam ausgeführt. Sobald alle in der Warteschlange befindlichen Anforderungen vollständig ausgeführt worden sind, wird die Dauer für die Beendigung der Stapelanforderung gemessen. Wenn Sie also einen großen NETTIME-Wert erhalten, überprüfen Sie, ob am fernen Ende eine ungewöhnliche Verarbeitung vorhanden ist.
- Überprüfen Sie, ob der Kanal die Nachrichtenwiederholung verwendet.
Wenn es dem Empfängerkanal nicht gelingt, eine Nachricht in eine Zielwarteschlange zu stellen, kann er die Nachricht erneut verarbeiten, anstatt sie sofort in eine Warteschlange für nicht zustellbare Nachrichten einzureihen. Die Wiederholungsverarbeitung kann dazu führen, dass die Stapelverarbei-

tung 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 317 beschrieben, und überprüfen Sie anschließend das Feld XQTIME in der Ausgabe des Statusbefehls für den Anzeigenkanal. Wenn die Werte der XQTIME-Indikatoren konstant hoch sind oder sich über den Messzeitraum erhöhen, deutet dies darauf hin, dass der Kanal nicht mit den Putting-Anwendungen mithalten kann.

Führen Sie die folgenden Tests aus:

Vorgehensweise

1. Prüfen Sie, ob die Exits verarbeitet werden.

Wenn auf dem Kanal, der diese Nachrichten liefert, Exits verwendet werden, können sie zu den Nachrichten hinzugefügt werden, die für die Verarbeitung von Nachrichten benötigt werden. Führen Sie die folgenden Prüfungen durch, um zu ermitteln, ob dies der Fall ist:

a) Überprüfen Sie in der Ausgabe des Befehls DIS CHSTATUS(QM1.TO.QM2) ALL das Feld EXIT-TIME.

Wenn die in den Exits verbrachte Zeit höher ist als erwartet, überprüfen Sie die Verarbeitung in Ihren Exits auf unnötige Schleifen oder zusätzliche Verarbeitungszeiten, insbesondere in Nachrichten-, Sende- und Empfangsexits. Diese Verarbeitung wirkt sich auf alle Nachrichten aus, die über den Kanal übertragen werden.

b) Überprüfen Sie in der Ausgabe des Befehls DIS CHSTATUS(QM1.TO.QM2) ALL das Feld SUBSTATE.

Wenn der Kanal einen der folgenden Unterzustände für einen signifikanten Zeitraum aufweist, überprüfen Sie die Verarbeitung in Ihren Exits:

- SCYEXIT
- RCVEXIT
- SENDEXIT
- MSGEXIT
- MREXIT

Weitere Informationen zum Teilstatus eines Kanals finden Sie in der Tabelle [Tabelle 32 auf Seite 323](#).

2. Überprüfen Sie, ob das Netz langsam ist.

Wenn sich die Nachrichten nicht schnell genug über einen Kanal bewegen, kann es sein, dass das Netz langsam ist. Führen Sie die folgenden Prüfungen durch, um zu ermitteln, ob dies der Fall ist:

a) Überprüfen Sie in der Ausgabe des Befehls DIS CHSTATUS(QM1.TO.QM2) ALL das Feld NET-TIME.

Diese Indikatoren werden gemessen, wenn der sendende Kanal seinen Partner für eine Antwort auffordert. Dies geschieht am Ende jedes Stapels und, wenn ein Kanal während des Überwachungssignals im Leerlauf ist.

- b) Wenn dieser Indikator zeigt, dass Rundreisen länger als erwartet sind, verwenden Sie andere Netzüberwachungstools, um die Leistung Ihres Netzes zu untersuchen.
3. Überprüfen Sie, ob der Kanal die Komprimierung verwendet.
Wenn der Kanal die Komprimierung verwendet, wird dadurch die Zeit für die Verarbeitung von Nachrichten hinzugefügt. Wenn der Kanal nur einen Komprimierungsalgorithmus verwendet, führen Sie die folgenden Prüfungen durch:
 - a) Überprüfen Sie in der Ausgabe des Befehls `DIS CHSTATUS(QM1.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 319 beschriebenen Tasks, um die Probleme mit den gefundenen Kanälen zu diagnostizieren, die die Erstellung des Builds verursachen.

Cluster überwachen

In einem Cluster können Sie Anwendungsnachrichten, Steuernachrichten und Protokolle überwachen. Es gibt spezielle Überwachungsaspekte, wenn die Clusterauslastung zwischen zwei oder mehr Instanzen einer Warteschlange verteilt wird.

Anwendungsnachrichten im Cluster überwachen

In der Regel durchlaufen alle Clusternachrichten, die den Warteschlangenmanager verlassen, den `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, unabhängig davon, welcher Clustersenderkanal für die Übertragung der Nachricht verwendet wird. Jeder Kanal enthält Nachrichten, die für diesen Kanal parallel zu allen anderen Clustersenderkanälen ausgerichtet sind. Ein wachsender Aufbau von Nachrichten in dieser Warteschlange kann auf ein Problem mit einem oder mehreren Kanälen hinweisen und muss untersucht werden:

- Die Tiefe der Warteschlange muss entsprechend für das Clusterdesign überwacht werden.
- Der folgende Befehl gibt alle Kanäle zurück, die mehr als eine Nachricht haben, die auf die Übertragungswarteschlange wartet:

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

Bei allen Clusternachrichten in einer einzigen Warteschlange ist es nicht immer leicht zu sehen, welcher Kanal Probleme hat, wenn er beginnt, sich zu füllen. Die Verwendung dieses Befehls ist eine einfache Möglichkeit, zu sehen, welcher Kanal verantwortlich ist.

Sie können einen Cluster-WS-Manager so konfigurieren, dass er mehrere Übertragungswarteschlangen hat. Wenn Sie das Warteschlangenmanagerattribut `DEFCLXQ` in `CHANNEL` ändern, wird jeder Clustersenderkanal einer anderen Clusterübertragungswarteschlange zugeordnet. Alternativ können Sie separate Übertragungswarteschlangen manuell konfigurieren. Führen Sie den folgenden Befehl aus, um alle Cluster-Übertragungswarteschlangen anzuzeigen, die Clustersenderkanälen zugeordnet sind:

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

Definieren Sie die Clusterübertragungswarteschlangen so, dass sie dem Muster des festen Stamms des Warteschlangennamens auf der linken Seite folgen. Anschließend können Sie die Länge aller Clusterübertragungswarteschlangen abfragen, die vom Befehl **DISPLAY CLUSMGR** zurückgegeben wurden, indem Sie einen generischen Warteschlangennamen verwenden:

```
DISPLAY QUEUE (qname *) CURDEPTH
```

Überwachen von Steuernachrichten im Cluster

Die Warteschlange `SYSTEM.CLUSTER.COMMAND.QUEUE` wird für die Verarbeitung aller Clustersteuerungsnachrichten für einen Warteschlangenmanager verwendet, die entweder vom lokalen Warteschlangenmanager generiert oder von anderen Warteschlangenmanagern im Cluster an diesen Warteschlangenmanager gesendet wurden. Wenn ein WS-Manager seinen Clusterstatus korrekt aufrechterhält, neigt diese Warteschlange zu null. Es gibt Situationen, in denen die Tiefe der Nachrichten in dieser Warteschlange vorübergehend zunehmen kann:

- Wenn viele Nachrichten in der Warteschlange vorhanden sind, wird das Churn im Clusterstatus angezeigt.
- Wenn Sie wichtige Änderungen vornehmen, können Sie die Warteschlange zwischen diesen Änderungen einlagern. Wenn z. B. Repositorys verschoben werden, können Sie die Warteschlange vor dem Verschieben des zweiten Repositorys auf null setzen.

Während ein Nachrichtenrückstand in dieser Warteschlange vorhanden ist, werden Aktualisierungen für den Clusterstatus oder die clusterbezogenen Befehle nicht verarbeitet. Wenn Nachrichten über einen längeren Zeitraum nicht aus dieser Warteschlange entfernt werden, ist eine weitere Untersuchung erforderlich, die zunächst durch die Überprüfung der Fehlerprotokolle des Warteschlangenmanagers (oder der `CHINIT`-Protokolle unter `z/OS`) ausgeführt wird. Dadurch kann möglicherweise der Prozess ermittelt werden, durch den diese Situation verursacht wird.

Der `SYSTEM.CLUSTER.REPOSITORY.QUEUE` enthält die Informationen zum Cluster-Repository-Cache als Anzahl von Nachrichten. Es ist üblich, dass Nachrichten immer in dieser Warteschlange vorhanden

sind, und mehr für größere Cluster. Daher ist die Tiefe der Nachrichten in dieser Warteschlange kein Problem für die Problemmeldung.

Protokolle überwachen

Probleme, die im Cluster auftreten, zeigen möglicherweise keine externen Symptome an Anwendungen für viele Tage (und sogar Monate) an, nachdem das Problem aufgrund des Zwischenspeicherns von Informationen und der verteilten Clusterbildung ursprünglich aufgetreten ist. Das ursprüngliche Problem wird jedoch häufig in den IBM MQ-Fehlerprotokollen (und CHINIT-Protokollen unter z/OS) dokumentiert. Aus diesem Grund ist es wichtig, diese Protokolle für alle Nachrichten, die sich auf das Clustering beziehen, aktiv zu überwachen. Diese Nachrichten müssen gelesen und verstanden werden, wobei gegebenenfalls Maßnahmen ergriffen werden müssen.

Beispiel: Eine Unterbrechung der Kommunikation mit einem WS-Manager in einem Cluster kann dazu führen, dass bestimmte Clusterressourcen gelöscht werden, da Cluster die Clusterressourcen regelmäßig erneut überprüfen, indem sie die Informationen erneut veröffentlichen. Die Nachricht [AMQ9465](#) oder [CSQX465I](#) auf z/OS-Systemen enthält eine Warnung über ein solches Ereignis, das möglicherweise auftritt. Diese Nachricht weist darauf hin, dass das Problem untersucht werden muss.

Besondere Hinweise zum Lastausgleich

Wenn die Clusterauslastung zwischen zwei oder mehr Instanzen einer Warteschlange verteilt wird, müssen konsumierende Anwendungen die Nachrichten in jeder der Instanzen verarbeiten. Wenn eine oder mehrere der konsumierenden Anwendungen die Verarbeitung von Nachrichten beenden oder stoppt, ist es möglich, dass das Clustering weiterhin Nachrichten an diese Instanzen der Warteschlange sendet. In dieser Situation werden diese Nachrichten erst verarbeitet, wenn die Anwendungen wieder ordnungsgemäß funktionieren. Aus diesem Grund ist die Überwachung der Anwendungen ein wichtiger Bestandteil der Lösung, und es müssen Maßnahmen ergriffen werden, um die Nachrichten in dieser Situation erneut zu leiten. Ein Beispiel für einen Mechanismus zum Automatisieren einer solchen Überwachung finden Sie in diesem Beispiel: [The Cluster Queue Monitoring sample program \(AMQSCLM\)](#).

Zugehörige Konzepte

„Verteilte Publish/Subscribe-Netze optimieren“ auf Seite 390

Verwenden Sie die Optimierungstipps in diesem Abschnitt, um die Leistung der verteilten Publish/Subscribe-Cluster und -Hierarchien von IBM MQ zu verbessern.

„Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen“ auf Seite 396

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

Übertragungswarteschlangenwechsel überwachen

Es ist wichtig, dass Sie den Prozess der Clustersenderkanäle überwachen, die die Übertragungswarteschlangen wechseln, damit die Auswirkungen auf Ihr Unternehmen minimiert werden. Sie sollten diesen Prozess beispielsweise nicht versuchen, wenn die Auslastung hoch ist oder wenn Sie viele Kanäle gleichzeitig wechseln.

Der Prozess zum Wechseln von Kanälen

Der Prozess, der zum Wechseln von Kanälen verwendet wird:

1. Der Kanal öffnet die neue Übertragungswarteschlange für die Eingabe und beginnt mit dem Abrufen von Nachrichten aus ihr (mithilfe des Abrufvorgangs nach Korrelations-ID).
2. Ein Hintergrundprozess wird vom Warteschlangenmanager eingeleitet, um alle Nachrichten, die für den Kanal eingereicht wurden, aus seiner alten Übertragungswarteschlange in seine neue Übertragungswarteschlange zu verschieben. Beim Verschieben von Nachrichten werden alle neuen Nachrichten für den Kanal in die alte Übertragungswarteschlange eingereicht, um die Reihenfolgeplanung beizubehalten. Dieser Prozess kann einige Zeit dauern, wenn eine große Anzahl von Nachrichten

für den Kanal in der alten Übertragungswarteschlange vorhanden ist oder neue Nachrichten schnell ankommen.

3. Wenn keine festgeschriebenen oder nicht festgeschriebenen Nachrichten für den Kanal in der alten Übertragungswarteschlange verbleiben, wird der Wechsel abgeschlossen. Neue Nachrichten werden jetzt direkt in die neue Übertragungswarteschlange eingereicht.

Um den Fall zu vermeiden, dass mehrere Kanäle gleichzeitig wechseln, bietet IBM MQ die Möglichkeit, die Übertragungswarteschlange eines oder mehrerer Kanäle zu wechseln, die nicht aktiv sind. Unter:

- IBM MQ for Multiplatforms heißt der Befehl **runswch1**
- IBM MQ for z/OS Das Dienstprogramm CSQUTIL kann verwendet werden, um einen Befehl SWITCH CHANNEL zu verarbeiten.

Status von Switchoperationen überwachen

Um den Status von Switchoperationen zu verstehen, können Administratoren die folgenden Aktionen ausführen:

- Überwachen Sie das Fehlerprotokoll des Warteschlangenmanagers (AMQERR01.LOG), in dem Nachrichten ausgegeben werden, um die folgenden Phasen während der Operation anzugeben:
 - Die Switchoperation wurde gestartet.
 - Das Verschieben von Nachrichten wurde gestartet.
 - Regelmäßige Aktualisierungen zur Anzahl der noch zu verschiebenden Nachrichten (wenn die Switchoperation nicht schnell abgeschlossen wird)
 - Das Verschieben von Nachrichten ist abgeschlossen.
 - Die Switchoperation ist abgeschlossen

Unter z/OS werden diese Nachrichten im Jobprotokoll des Warteschlangenmanagers ausgegeben, nicht im Jobprotokoll des Kanalinitiators, obwohl eine einzelne Nachricht von einem Kanal in das Jobprotokoll des Kanalinitiators ausgegeben wird, wenn beim Start ein Wechsel eingeleitet wird.

- Mit dem Befehl DISPLAY CLUSQMGR können Sie die Übertragungswarteschlange abfragen, die momentan von jedem Clustersenderkanal verwendet wird.
- Führen Sie den Befehl **runswch1** (oder CSQUTIL unter z/OS) im Abfragemodus aus, um den Wechselstatus eines oder mehrerer Kanäle zu ermitteln. Die Ausgabe dieses Befehls gibt Folgendes für jeden Kanal an:
 - Gibt an, ob für den Kanal eine Switchoperation ansteht
 - Von und zu welcher Übertragungswarteschlange der Kanal wechselt
 - Wie viele Nachrichten in der alten Übertragungswarteschlange verbleiben

Jeder Befehl ist wirklich nützlich, da Sie in einem Aufruf den Status jedes Kanals, die Auswirkung einer Konfigurationsänderung und die Beendigung aller Switchoperationen ermitteln können.

Potenzielle Probleme, die auftreten können

Im Abschnitt Mögliche Probleme beim Wechseln von Übertragungswarteschlangen finden Sie eine Liste einiger Probleme, die beim Wechseln der Übertragungswarteschlange auftreten können, ihre Ursachen und die wahrscheinlichsten Lösungen.

Zugehörige Konzepte

„Verteilte Publish/Subscribe-Netze optimieren“ auf Seite 390

Verwenden Sie die Optimierungstipps in diesem Abschnitt, um die Leistung der verteilten Publish/Subscribe-Cluster und -Hierarchien von IBM MQ zu verbessern.

„Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen“ auf Seite 396

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rück-

stand nicht konsumieren Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

Multi Lastausgleich von Anwendungen überwachen

Mit dem Befehl **DISPLAY APSTATUS** können Sie den Lastausgleichsstatus von Anwendungen in einem Uniform-Cluster überwachen und ermitteln, warum eine Anwendung wider Erwartung nicht in den Lastausgleich einbezogen ist.

Aktuellen Status von Anwendungen über mehrere Warteschlangenmanager hinweg in einem Cluster überwachen

Sie können sich von jedem Warteschlangenmanager in einem Uniform-Cluster aus einen Überblick über den aktuellen Status der Anwendungen über alle Warteschlangenmanager eines Clusters hinweg verschaffen, indem Sie den Befehl `DIS APSTATUS` ausführen.

Ab IBM MQ 9.2.0 enthält die Ausgabe auch das Feld **TYPE**.

Nachdem ein Warteschlangenmanager kurz zuvor gestartet wurde, ist beispielsweise eine Ausgabe mit ungefähr folgendem Text möglich:

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCOUNT(8)
BALANCED(UNKNOWN)
TYPE (APPL)
```

Dies zeigt an, dass der Uniform-Cluster eine Anwendung namens MYAPP enthält und es derzeit acht Instanzen gibt, von denen alle acht als im Uniform-Cluster verschiebbar gelten. Der Lastausgleichswert UNKNOWN ist ein temporärer Wert, der angibt, dass der Warteschlangenmanager noch nicht versucht hat, die Anwendung im Bedarfsfall neu zu verteilen.

Nach einer kurzen Zeit ist folgende Ausgabe wahrscheinlicher:

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCOUNT(8)
BALANCED(NO)
TYPE (APPL)
```

Diese Ausgabe zeigt, dass die Anwendung acht Instanzen hat, zwischen denen im Uniform-Cluster jedoch kein Lastausgleich erfolgt. An diesem Punkt lohnt es sich, einen Blick auf die Verteilung der Anwendungen im Cluster zu werfen.

Führen Sie hierzu erneut den Befehl **DIS APSTATUS** aus. Beachten Sie, dass Sie diesen Befehl für jeden beliebigen Warteschlangenmanager im Uniform-Cluster ausführen können:

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(6)                     MOVCOUNT(6)
BALSTATE(HIGH)                 LMSGDATE(2019-05-24)
LMSGTIME(13:11:10)            QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                     MOVCOUNT(1)
BALSTATE(LOW)                 LMSGDATE(2019-05-24)
LMSGTIME(13:11:03)            QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                     MOVCOUNT(1)
BALSTATE(LOW)                 LMSGDATE(2019-05-24)
```

```
LMSGTIME(13:11:07) QMNAME (UNID003)
QMID (UNID003_2019-05-24_13.09.43)
TYPE (QMGR)
```

Daraus wird zu diesem Zeitpunkt ersichtlich, dass der Warteschlangenmanager UNID001 über sechs Instanzen verfügt, die Warteschlangenmanager UNID0002 und UNID0003 jedoch nur über jeweils eine Instanz. Die BALSTATE-Ausgabe zeigt den Laustausgleichsstatus an, der zuletzt vom Warteschlangenmanager gemeldet wurde. Beachten Sie jedoch, dass die Instanzanzahl möglicherweise aktueller als das Feld BALSTATE ist.

Diese Ausgabe gibt auch einen hilfreichen Hinweis darauf, dass der Uniform-Cluster Informationen zu dieser Anwendung verteilt. Die Anwendung ist allen Warteschlangenmanagerinstanzen in im Uniform-Cluster bekannt, und das Datum und die Uhrzeit der letzten Nachricht sind sehr aktuell.

Außerdem gibt das Feld AKTIV an, dass alle Warteschlangenmanager im Cluster als kommunizierend miteinander betrachtet werden; wenn ACTIVE auf NO auf einem beliebigen Warteschlangenmanager gesetzt ist, zeigt dies an, dass die reguläre Kommunikation mit dem Warteschlangenmanager unterbrochen wurde.

Wenn der Lastausgleich sich selbst überlassen wird, würde die Ausgabe letztendlich etwa folgendermaßen aussehen:

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP) ACTIVE (YES)
COUNT(3) MOVCCOUNT (3)
BALSTATE(OK) LMSGDATE(2019-05-24)
LMSGTIME(13:14:22) QMNAME (UNID001)
QMID (UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP) ACTIVE (YES)
COUNT(3) MOVCCOUNT (3)
BALSTATE(OK) LMSGDATE(2019-05-24)
LMSGTIME(13:13:53) QMNAME (UNID002)
QMID (UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP) ACTIVE (YES)
COUNT(2) MOVCCOUNT (2)
BALSTATE(OK) LMSGDATE(2019-05-24)
LMSGTIME(13:13:47) QMNAME (UNID003)
QMID (UNID003_2019-05-24_13.09.43)
TYPE (QMGR)
```

Der BALSTATE der Warteschlangenmanager ist jetzt OK, was darauf hinweist, dass ein stabiler Zustand erreicht wurde.

Einzelne Anwendungsinstanzen überwachen

Sie können die einzelnen Anwendungsinstanzen betrachten, tun dies allerdings für jeden Warteschlangenmanager. Sehen Sie sich die Ausgabe unter UNID001 an:

```
1 : DIS APSTATUS(*) type(LOCAL)
AMQ8932I: Display application status details.
APPLNAME (MYAPP)
CONNTAG (MQCT02DFE75C02EA0A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS (1) IMMREASN (NONE)
IMMCOUNT (0) IMMDATE ( )
IMMTIME ( ) MOVABLE (YES)
AMQ8932I: Display application status details.
APPLNAME (MYAPP)
CONNTAG (MQCT02DFE75C02E50A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS (1) IMMREASN (NONE)
IMMCOUNT (0) IMMDATE ( )
IMMTIME ( ) MOVABLE (YES)
AMQ8932I: Display application status details.
APPLNAME (MYAPP)
CONNTAG (MQCT02DFE75C02E60A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS (1) IMMREASN (NONE)
IMMCOUNT (0) IMMDATE ( )
IMMTIME ( ) MOVABLE (YES)
```

```
TYPE (LOCAL)
One MQSC command read.
```

Das Vorhandensein von MOVABLE(YES) zeigt an, dass diese Instanz bei Bedarf in einen anderen Warteschlangenmanager im Cluster verschoben werden kann. Im folgenden Beispiel ist die Anwendungsinstanz nicht verschiebbar, da sie sich nicht als Client verbindet:

```
3 : DISPLAY APSTATUS('ServerApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ServerApp)
CONNTAG(MQCT02DFE75C01800B20UNID001_2019-05-24_13.09.35ServerApp)
CONNS(1) IMMREASN(NOTCLIENT)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)
```

Im Feld **CONNTAG** können Sie die einzelnen Warteschlangenmanagerverbindungen dieser betreffenden Instanz sehen. Dies kann hilfreich sein, wenn für die Anwendungsinstanz IMMREASN(NOTRECONN) angegeben ist. Im folgenden Beispiel ist die Clientanwendung nicht verschiebbar, und die Untersuchung der Verbindungsoptionen zeigt, dass MQCNO_RECONNECT im Feld **CONNOPTS** nicht vorhanden ist.

```
1 : DISPLAY APSTATUS('ClientApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ClientApp)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
CONNS(1) IMMREASN(NOTRECONN)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)

2 : DISPLAY CONN(*) TYPE(CONN) WHERE(CONNTAG eq
'MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp') ALL
AMQ8276I: Display Connection details.
CONN(02DFE75C01CB0B20)
EXTCONN(414D5143554E49443030312020202020)
TYPE(CONN)
PID(14656) TID(20)
APPLDESC(IBM MQ Channel) APPLTAG(ClientApp)
APPLTYPE(USER) ASTATE(NONE)
CHANNEL(SYSTEM.DEF.SVRCONN) CLIENTID( )
CONNNAME(127.0.0.1)
CONNOPTS(MQCNO_HANDLE_SHARE_BLOCK,MQCNO_SHARED_BINDING)
USERID(MyUserid) UOWLOG( )
UOWSTDA( ) UOWSTTI( )
UOWLOGDA( ) UOWLOGTI( )
URTYPE(QMGR)
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])
QMURID(0.0) UOWSTATE(NONE)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
TYPE (CONN)
```

Zugehörige Konzepte

[Automatischer Ausgleich von Anwendungen](#)

z/OS

Monitoring performance and resource usage on z/OS

Use this topic to understand the facilities available to monitor the performance, and resource usage of your IBM MQ for z/OS subsystems.

Related tasks

[Configuring queue managers on z/OS](#)

[Administering IBM MQ for z/OS](#)

z/OS

Introduction to monitoring IBM MQ for z/OS

Use this topic as an overview of the monitoring facilities available for IBM MQ for z/OS. For example, obtaining snapshots, using IBM MQ trace, online monitoring, and events.

This topic describes how to monitor the performance and resource usage of IBM MQ.

- It outlines some of the information that you can retrieve and briefly describes a general approach to investigating performance problems. See [“Investigating performance problems” on page 338](#) for more information.
- It describes how you can collect statistics about the performance of IBM MQ by using SMF records.
- It describes how to gather accounting data to enable you to charge your customers for their use of your IBM MQ systems.
- It describes how to use IBM MQ events (alerts) to monitor your systems.

Here are some of the tools you might use to monitor IBM MQ; they are described in the sections that follow:

- Tools provided by IBM MQ:
 - [Using DISPLAY commands](#)
 - [“Using CICS adapter statistics” on page 333](#)
 - [“Using IBM MQ events” on page 335](#)
- z/OS service aids:
 - [“Using System Management Facility” on page 335](#)
- Other IBM licensed programs:
 - [Using the Resource Measurement Facility](#)
 - [Using Tivoli Decision Support for z/OS](#)
 - [Using the CICS monitoring facility](#)

Information about interpreting the data gathered by the performance statistics trace is given in [“Interpreting IBM MQ for z/OS performance statistics” on page 339](#).

Information about interpreting the data gathered by the accounting trace is given in [“Interpreting IBM MQ for z/OS accounting data” on page 376](#).

Getting snapshots of IBM MQ using the DISPLAY commands

IBM MQ provides the MQSC facility which can give a snapshot of the performance, and resource usage using the DISPLAY commands.

You can get an idea of the current state of IBM MQ by using the DISPLAY commands and, for the CICS adapter, the CICS adapter panels.

Using DISPLAY commands

You can use the IBM MQ MQSC DISPLAY or PCF Inquire commands to obtain information about the current state of IBM MQ. They provide information about the status of the command server, process definitions, queues, the queue manager, and its associated components. These commands are:

MQSC command	PCF command
DISPLAY ARCHIVE	Inquire Archive
DISPLAY AUTHINFO	Inquire Authentication Information Object
DISPLAY CFSTATUS	Inquire CF Structure Status
DISPLAY CFSTRUCT	Inquire CF Structure
DISPLAY CHANNEL	Inquire Channel
DISPLAY CHINIT	Inquire Channel Initiator
DISPLAY CHSTATUS	Inquire Channel Status
DISPLAY CMDSERV	

MQSC command	PCF command
DISPLAY CLUSQMgr	Inquire Cluster Queue Manager
DISPLAY CONN	Inquire Connection
DISPLAY GROUP	Inquire Group
DISPLAY LOG	Inquire Log
DISPLAY PROCESS	Inquire Process
DISPLAY QMgr	Inquire Queue Manager
DISPLAY QSTATUS	Inquire Queue Status
DISPLAY QUEUE	Inquire Queue
DISPLAY SECURITY	Inquire Security
DISPLAY STGCLASS	Inquire Storage Class
DISPLAY SYSTEM	Inquire System
DISPLAY TRACE	
DISPLAY USAGE	Inquire Usage

For the detailed syntax of each command, see MQSC commands or PCF commands. All of the functions of these commands (except DISPLAY CMDSERV and DISPLAY TRACE) are also available through the operations and control panels.

These commands provide a snapshot of the system only at the moment the command was processed. If you want to examine trends in the system, you must start an IBM MQ trace and analyze the results over a period of time.

Using CICS adapter statistics

If you are an authorized CICS user, you can use the CICS adapter control panels to display CICS adapter statistics dynamically.

These statistics provide a snapshot of information related to CICS thread usage and situations when all threads are busy. The display connection panel can be refreshed by pressing the Enter key.

For more information about configuring the IBM MQ CICS adapter, see the [Configuring connections to MQ](#) section of the CICS documentation.

Using IBM MQ trace

You can record performance statistics and accounting data for IBM MQ by using the IBM MQ trace facility. Use this topic to understand how to control IBM MQ trace.

The data generated by IBM MQ is sent to:

- The System Management Facility (SMF), specifically as SMF record type 115, subtypes 1 and 2 for the performance statistics trace
- The SMF, specifically as SMF record type 116, subtypes zero, 1, and 2 for the accounting trace.

If you prefer, the data generated by the IBM MQ accounting trace can also be sent to the generalized trace facility (GTF).

Starting IBM MQ trace

You can start the IBM MQ trace facility at any time by issuing the IBM MQ [START TRACE](#) command.

Accounting data can be lost if the accounting trace is started or stopped while applications are running. To collect accounting data successfully, the following conditions must apply:

- The accounting trace must be active when an application starts, and it must still be active when the application finishes.
- If the accounting trace is stopped, any accounting data collection that was active stops.

You can also start collecting some trace information automatically if you specify YES on the SMFSTAT (SMF STATISTICS) and SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro.  These parameters are described in [Using CSQ6SYSP](#).

Before starting an IBM MQ trace, read [“Using System Management Facility”](#) on page 335.

Controlling IBM MQ trace

To control the IBM MQ trace data collection at start-up, specify values for the parameters in the CSQ6SYSP macro when you customize IBM MQ.  See [Using CSQ6SYSP](#) for details.

You can control IBM MQ tracing when the queue manager is running with these commands:

- START TRACE
- ALTER TRACE
- STOP TRACE

You can choose the destination to which trace data is sent. Possible destinations are:

SMF

System Management Facility

GTF

Generalized Trace Facility (accounting trace only)

SRV

Serviceability routine for diagnostic use by IBM service personnel

For daily monitoring, information is sent to SMF (the default destination). SMF data sets typically contain information from other systems; this information is not available for reporting until the SMF data set is dumped.

You can also send accounting trace information to the GTF. This information has an event identifier of 5EE.

 [The MQI call and user parameter, and z/OS generalized trace facility \(GTF\)](#) describes how to deal with IBM MQ trace information sent to the GTF.

For information about IBM MQ commands, see [MQSC commands](#).

Effect of trace on IBM MQ performance

Using the IBM MQ trace facility can have a significant effect on IBM MQ and transaction performance. For example, if you start a global trace for class 1 or for all classes, it is likely to increase processor usage and transaction response times by approximately 50%. However, if you start a global trace for classes 2 - 4 alone, the increase in processor usage and transaction response times is likely to be less than 1% additional processor cost to the cost of IBM MQ calls. The same applies for a statistics or accounting trace.

Using IBM MQ online monitoring

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the MONQ, MONCHL, and MONACLS attributes.

Table 33 on page 335 summarizes the commands to set these attributes at different levels and to display the monitoring information.

Table 33. Setting and displaying attributes to control online monitoring

Attribute	Applicable at this level	Set using command	Display monitoring information using command
MONQ	Queue	DEFINE QLOCAL DEFINE QMODEL ALTER QLOCAL ALTER QMODEL	DISPLAY QSTATUS
	Queue manager	ALTER QMGR	
MONCHL	Channel	DEFINE CHANNEL ALTER CHANNEL	DISPLAY CHSTATUS
	Queue manager	ALTER QMGR	
MONACLS	Queue manager	ALTER QMGR	

For full details of these commands, see [MQSC commands](#). For more information about online monitoring, see “IBM MQ-Netz überwachen” on page 5.

Using IBM MQ events

IBM MQ instrumentation events provide information about errors, warnings, and other significant occurrences in a queue manager. You can monitor the operation of all your queue managers by incorporating these events into your own system management application.

IBM MQ instrumentation events fall into the following categories:

Queue manager events

These events are related to the definitions of resources within queue managers. For example, an application attempts to put a message to a queue that does not exist.

Performance events

These events are notifications that a threshold condition has been reached by a resource. For example, a queue depth limit has been reached, or the queue was not serviced within a predefined time limit.

Channel events

These events are reported by channels as a result of conditions detected during their operation. For example, a channel instance is stopped.

Configuration events

These events are notifications that an object has been created, changed, or deleted.

When an event occurs, the queue manager puts an *event message* on the appropriate *event queue*, if defined. The event message contains information about the event that can be retrieved by a suitable IBM MQ application.

IBM MQ events can be enabled using the IBM MQ commands or the operations and control panels.

See “Ereignistypen” on page 12 for information about the IBM MQ events that generate messages, and for information about the format of these messages. See [Event message reference](#) for information about enabling the events.

Using System Management Facility

You can use SMF to collect statistics and accounting information. To use SMF, certain parameters must be set in z/OS and in IBM MQ.

System management facility (SMF) is a z/OS service aid used to collect information from various z/OS subsystems. This information is dumped and reported periodically, for example, hourly. You can use SMF with the IBM MQ trace facility to collect data from IBM MQ. In this way you can monitor *trends*, for example, in system utilization and performance, and collect accounting information about each user ID using IBM MQ.

To record performance statistics (record type 115) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(115))
```

To record accounting information (record type 116) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(116))
```

To use the z/OS command SETSMF, either PROMPT(ALL) or PROMPT(LIST) must be specified in the SMFPRM xx member. See [SMFPRMxx \(system management facilities \(SMF\) parameters\)](#) for more information.

You can start collecting some trace information automatically if you specify YES on the SMFSTAT (SMF STATISTICS) and SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro; this is described in [Using CSQ6SYSP](#).

Specifying YES on the SMFSTAT and SMFACCT parameters enables you to collect trace information as a queue manager starts.

You can also start collection of the data when the queue manager is running with the **START TRACE** command, specifying START TRACE (A) or START TRACE (S).

You can turn on or off the recording of accounting information at the queue or queue manager level using the ACCTQ parameter of the **DEFINE QLOCAL**, **DEFINE QMODEL**, **ALTER QLOCAL**, **ALTER QMODEL**, or **ALTER QMGR** commands. See [MQSC commands](#) for details of these commands.

You can control the collection of channel accounting data at the channel or queue manager level using the **STATCHL** parameter of the **DEFINE CHANNEL**, **ALTER CHANNEL** or **ALTER QMGR** commands.

You can specify the interval at which IBM MQ collects statistics and accounting data in one of these ways:

- You can collect statistics data and accounting data at different intervals, using STATIME (statistics data) and ACCTIME (accounting data) in your system parameters (described in [Using CSQ6SYSP](#)).
- You can collect statistics data and accounting data at the same interval by specifying a value for STATIME in your system parameters (described in [Using CSQ6SYSP](#)).
- You can collect statistics data and accounting data by specifying zero for STATIME.

SMF must be running before you can send data to it. For more information about SMF, see the [z/OS MVS System Management Facilities \(SMF\) manual](#).

For the statistics and accounting data to be reset, at least one MQI call must be issued during the accounting interval.

Allocating additional SMF buffers

When you start a trace, you must ensure that you allocate adequate SMF buffers. Specify SMF buffering on the VSAM BUFSP parameter of the access method services DEFINE CLUSTER statement. Specify CISZ(4096) and BUFSP(81920) on the **DEFINE CLUSTER** statement for each SMF VSAM data set.

If an SMF buffer shortage occurs, SMF rejects any trace records sent to it. IBM MQ sends a CSQW133I message to the z/OS console when this occurs. IBM MQ treats the error as temporary and remains

active even though SMF data can be lost. When the shortage has been alleviated and trace recording has resumed, IBM MQ sends a CSQW123I message to the z/OS console.

Reporting data in SMF

You can use the SMF program IFASMFDP (or IFASMF DL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed.

There are several ways to report on this data, for example:

- Write an application program to read and report information from the SMF data set. You can then tailor the report to fit your exact needs.
- Use Performance Reporter to process the records. For more information, see [“Using other products with IBM MQ”](#) on page 337.

No interval CLASS(03) SMF accounting records produced during long running processes

You are collecting CLASS(3) SMF116 accounting records for IBM MQ, but are getting no records produced while a long running process runs.

The CLASS(3) SMF116 accounting records normally are produced only when a process ends. For long running processes, for example CICS, this might not produce a sufficient number of records, as the process can run for a month or longer. However, you might want to gather SMF116 records at set time intervals while a process is running.

To gather CLASS(3) SMF116 accounting records you must set the following:

SMFACCT

=YES

SMFSTAT

=YES or NO, where

YES

Causes records to be produced if a collection broadcast is received.

No

Causes you to get a CLASS(3) SMF116 record produced only when a process ends

and issue the following command:

```
START TRACE(ACCTG) DEST(SMF) CLASS(03)
```

If you have set SMFSTAT=YES and a collection broadcast occurs, an interval CLASS(3) SMF116 accounting record is produced for any process currently running that was also running at the time of the previous collection broadcast.

You can set the collection broadcast to occur on a regular time interval by setting STATIME in [CSQ6SYSP](#) as follows:

- If your STATIME has been set to a value greater than 0, that is your broadcast interval in minutes.
- If your STATIME = 0 the SMF broadcast of your system is used (SMF INTVAL)
- If your STATIME = 0 and your SMF INTVAL is not set, no broadcast occurs and no interval records are produced

Using other products with IBM MQ

You can use other products to help you to improve the presentation of, or to augment statistics related to, performance and accounting. For example, Resource Measurement Facility, Tivoli Decision Support, and CICS monitoring.

Using the Resource Measurement Facility

Resource Measurement Facility (RMF) is an IBM licensed program (program number 5685-029) that provides system-wide information about processor utilization, I/O activity, storage, and paging. You can use RMF to monitor the utilization of physical resources across the whole system dynamically. For more information, see the [z/OS Resource Measurement Facility User's Guide](#).

Using Tivoli Decision Support for z/OS

You can use Tivoli Decision Support for z/OS to interpret RMF and SMF records.

Tivoli Decision Support for z/OS is an IBM licensed program (program number 5698-B06) that enables you to manage the performance of your system by collecting performance data in a Db2® database and presenting the data in various formats for use in systems management. Tivoli Decision Support for can generate graphic and tabular reports using systems management data it stores in its Db2 database. It includes an administration dialog, a reporting dialog, and a log collector, all of which interact with a standard Db2 database.

This is described in the [IBM Tivoli Decision Support for z/OS: Administration Guide and Reference](#).

Using the CICS monitoring facility

The CICS monitoring facility provides performance information about each CICS transaction running. It can be used to investigate the resources used and the time spent processing transactions. For background information, see the [CICS Performance Guide](#) and [Developing CICS System Programs](#), together with the two companion reference manuals, formerly called the [CICS Customization Guide](#).

Investigating performance problems

Performance problems can arise from various factors. For example, incorrect resource allocation, poor application design, and I/O restraints. Use this topic to investigate some of the possible causes of performance problems.

Performance can be adversely affected by:

- Buffer pools that are an incorrect size
- Lack of real storage
- I/O contention for page sets or logs
- Log buffer thresholds that are set incorrectly
- Incorrect setting of the number of log buffers
- Large messages
- Units of recovery that last a long time, incorporating many messages for each sync point
- Messages that remain on a queue for a long time
- RACF® auditing
- Unnecessary security checks
- Inefficient program design

When you analyze performance data, always start by looking at the overall system before you decide that you have a specific IBM MQ problem. Remember that almost all symptoms of reduced performance are magnified when there is contention. For example, if there is contention for DASD, transaction response times can increase. Also, the more transactions there are in the system, the greater the processor usage and greater the demand for both virtual and real storage.

In such situations, the system shows heavy use of *all* its resources. However, the system is actually experiencing normal system stress, and this stress might be hiding the cause of a performance reduction. To find the cause of such a loss of performance, you must consider all items that might be affecting your active tasks.

Investigating the overall system

Within IBM MQ, the performance problem is either increased response time or an unexpected and unexplained heavy use of resources. First check factors such as total processor usage, DASD activity, and paging. An IBM tool for checking total processor usage is resource management facility (RMF). In general, you must look at the system in some detail to see why tasks are progressing slowly, or why a specific resource is being heavily used.

Start by looking at general task activity, then focus on particular activities, such as specific tasks or a specific time interval.

Another possibility is that the system has limited real storage; therefore, because of paging interrupts, the tasks progress more slowly than expected.

Investigating individual tasks

You can use the accounting trace to gather information about IBM MQ tasks. These trace records tell you a great deal about the activity that the task has performed, and about how much time the task spent suspended, waiting for latches. The trace record also includes information about how much Db2 and coupling facility activity were performed by the task.

Interpreting IBM MQ accounting data is described in [“Interpreting IBM MQ for z/OS accounting data”](#) on page 376.

Long running units of work can be identified by the presence of message CSQR026I in the job log. This message indicates that a task has existed for more than three queue manager checkpoints and its log records have been shunted.  For a description of log record shunting, see [The log files](#).

Interpreting IBM MQ for z/OS performance statistics

Use this topic as an index to the different SMF records created by IBM MQ for z/OS.

IBM MQ for z/OS performance statistics are written as SMF type 115 records. Statistics records are produced periodically at a time interval specified by the **STATIME** parameter of the CSQ6SYSP system parameter module, or at the SMF global recording interval if you specify zero for **STATIME**. The information provided in the SMF records comes from the following components of IBM MQ:

Buffer manager	Manages the buffer pools in virtual storage and the writing of pages to page sets as the buffer pools become full. Also manages the reading of pages from page sets.
Coupling facility manager	Manages the interface with the coupling facility.
Data manager	Manages the links between messages and queues. It calls the buffer manager to process the pages with messages on them.
Db2 manager	Manages the interface with the Db2 database that is used as the shared repository.
Lock manager	Manages locks
Log manager	Manages the writing of log records, which are essential for maintaining the integrity of the system if there is a back out request, or for recovery, if there is a system or media failure.
Message manager	Processes all IBM MQ API requests.
Storage manager	Manages storage, for example, storage pool allocation, expansion, and deallocation.
Topic manager	Manages the topic and subscription information

Coupling facility SMDS manager Manages the shared message data sets (SMDS) for large messages stored in the coupling facility.

IBM MQ statistics are written to SMF as SMF type 115 records. The following subtypes can be present:

- 1**
System information, for example, related to the logs and storage.
- 2**
Information about number of messages and paging information. Queue sharing group information related to the coupling facility and Db2.
- 5 and 6**
Detailed information about internal storage usage in the queue manager address space. While you can view this information, some of it is intended only for IBM use.
- 7**
Storage manager summary information. While you can view this information, some of it is intended only for IBM use.
- 201**
Page set input/output information
- 215**
Buffer pool information
- 216**
Queue information
- 231**
System information for the channel initiator address space.

Note that:

- Subtype 1, 2, 201, and 215 records are created with statistics trace class 1.
- Subtype 5, 6, and 7 records are created with statistics trace class 3.
- Subtype 231 records are created with statistics trace class 4.
- Subtype 216 records are created with statistics trace class 5.

The subtype is specified in the SM115STF field (shown in [Table 34 on page 343](#)).

Erforderliche Befehle zum Erfassen von SMF-Datensätzen vom Typ 115 und 116

Verwenden Sie dieses Thema als Referenz für die Befehle, die zum Erfassen von SMF-Datensätzen vom Typ 115 und 116 erforderlich sind.

Verwenden des START TRACE-Befehls

Verwenden Sie die dynamische Version des [TRACE STARTEN](#) Befehl mit den folgenden Optionen zum Sammeln der Datensätze:

- `START TRACE(STAT) DEST SMF CLASS(*)` und `START TRACE(ACCTG) DEST SMF CLASS(*)` startet Spuren für die Klassen 1 bis 3
- `START TRACE(STAT) DEST SMF CLASS(4)` und `START TRACE(ACCTG) DEST SMF CLASS(4)` startet die Kanalinitiatorstatistik bzw. die Kanalabrechnungsdaten.

Sehen [SMF-Daten des Kanalinitiators planen](#) für weitere Einzelheiten zu den Informationen zum Kanalinitiator der Klasse 4.

- `START TRACE(STAT) DEST SMF CLASS(5)` startet Warteschlangenstatistiken.



Achtung: Wenn Sie die Konsolenversion des Befehls `add` verwenden `cp` zum Anfang des von Ihnen ausgegebenen Befehls.

Für die CLASS(4)-Kanalstatistiken müssen Sie das STATCHL-Attribut in der CHANNEL-Definition festlegen. Weitere Informationen finden Sie unter „Erfassung von Kanalstatistikdaten steuern“ auf Seite 152.

Für die Warteschlangenstatistiken von CLASS(5) müssen Sie das STATQ-Attribut in der QUEUE- und/oder QMGR-Definition festlegen. Weitere Informationen finden Sie unter „Erfassung von Warteschlangenstatistikdaten steuern“ auf Seite 151.

SMF-Typ 115-Datensätze -IBM MQ Statistiken

Um die aktuelle Statistikerfassung zu überprüfen, führen Sie den [TRACE ANZEIGEN](#) Befehl `DISPLAY TRACE (STAT)`. Daraufhin sollte Folgendes angezeigt werden:

```
RESPONSE=MPX1
CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
TNO TYPE CLASS DEST USERID RMID
02 STAT 01,02,03,04 SMF * *
END OF TRACE REPORT
```

Anmerkung: Wenn Sie das Attribut ZPARAM SMFSTAT oder ZPARAM SMFACCT auf ein Sternchen setzen, werden jetzt nur noch die Klassen 1, 2 und 3 gesteuert. Sie müssen die Klassen 4 und 5 mit dem Befehl `START TRACE` aktivieren.

SMF Typ 116 Klasse 3 & 4 Daten –IBM MQ Aufgaben- und Kanalabrechnungsdatensätze

Um die aktuelle Statistikerfassung zu überprüfen, führen Sie den [TRACE ANZEIGEN](#) Befehl `DISPLAY TRACE (ACCT)`. Daraufhin sollte Folgendes angezeigt werden:

```
RESPONSE=MPX1
CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
TNO TYPE CLASS DEST USERID RMID
03 ACCTG * SMF * *
END OF TRACE REPORT
CSQ9022I QML1 CSQWVCM1 'DISPLAY TRACE' NORMAL COMPLETION
```

Sie können sich wenden `SMF116`. Die Datenerfassung lässt sich mithilfe der folgenden Befehle dynamisch ein- und ausschalten, da kein Warteschlangenmanagerzyklus erforderlich ist:

```
START TRACE(ACCTG) DEST(SMF) CLASS(3)
START TRACE(ACCTG) DEST(SMF) CLASS(4)

STOP TRACE(ACCTG) DEST(SMF) CLASS(3)
STOP TRACE(ACCTG) DEST(SMF) CLASS(4)
```

Zur Erstellung der SMF sind für die Kanalabrechnungsdaten zusätzliche Einstellungen erforderlich. Im Folgenden sind die einzelnen Schritte aufgeführt:

- Geben Sie den Befehl ein, um das Sammeln der Informationen für alle Sender-Empfänger- und Client-Verbindungen zu aktivieren:

```
ALTER QMGR STATCHL(HIGH)
```

- Geben Sie den Befehl ein, um das Sammeln von Informationen für automatisch definierte Clusterkanäle zu aktivieren, wenn Clustering verwendet wird:

```
ALTER QMGR STATCLS(HIGH)
```

Anmerkung: In den vorhergehenden Anweisungen haben die Werte HIGH, MED oder LOW dieselbe Wirkung.

- Überprüfen Sie, ob bei allen Kanälen STATCHL auf QMGR eingestellt ist:

Überprüfen der STATIME



Achtung: Um den neuesten Tools zu entsprechen, sollten Sie das Intervall auf höchstens fünf Minuten einstellen. Verwenden Sie dazu SYSTEM EINSTELLEN Befehl, obwohl der Standardwert 30 Minuten beträgt.

Wenn Sie eine Queue Sharing Group (QSG) auswerten, stellen Sie sicher, dass STATIME in der gesamten QSG konsistent ist.

- Zeigen Sie die Systemeinstellungen mit den folgenden ANZEIGESYSTEM Befehl:

```
DISPLAY SYSTEM
```

- Wenn der STATIME-Wert Null ist, was bedeutet, dass das Statistikintervall auf den LPAR-Standardwert eingestellt ist, ist dies normalerweise in Ordnung.

Es gab Fälle, in denen LPARs mit sehr großem Volumen zu unterschiedlichen Dauern der SMF-Datenproduktion führten. Wenn bei der Auswertung der Daten sehr unterschiedliche Zeiträume aufgetreten sind, legen Sie STATIME für alle Warteschlangenmanager auf einen Wert ungleich null fest.

- Die folgenden Beispiele legen das STATIME-Intervall auf fünf Minuten fest:

```
SET SYSTEM STATIME(05)
```

oder wenn bei IBM MQ 9.2.4 oder höher:

```
SET SYSTEM STATIME(05.00)
```

Anmerkung: Das neue Intervall wird erst nach Ablauf des aktuellen Intervalls wirksam. Sie müssen die Änderung daher vor Beginn der Datenerfassung vornehmen.

ACCTIME-Wert für Warteschlangenmanager auf Release-Ebene prüfen IBM MQ 9.2.4 oder höher:

- Wenn ACCTIME auf -1 gesetzt ist, ist es dasselbe wie das STATIME-Attribut.
- Wenn diese Auswertung für eine QSG erfolgt, stellen Sie sicher, dass die ACCTIME in der gesamten QSG konsistent ist. Wenn dies nicht der Fall ist, legen Sie die Werte auf das gleiche Intervall fest, bevor Sie mit der Datenerfassung beginnen.
- Wenn die ACCTIME auf 30 oder höher eingestellt ist, ändern Sie sie auf höchstens 15 mit dem SYSTEM EINSTELLEN Befehl. Die folgenden Beispiele legen das ACCTIME-Intervall auf fünf Minuten fest:

```
SET SYSTEM ACCTIME(05.00)
```

Anmerkung: Das neue Intervall wird erst nach Ablauf des aktuellen Intervalls wirksam. Sie müssen die Änderung daher vor Beginn der Datenerfassung vornehmen.

ACCTQ-Einstellung auf den Warteschlangenmanagern prüfen

- Zeigen Sie die ACCTQ-Einstellung für die Warteschlangenmanager an, die Sie untersuchen, indem Sie den folgenden Befehl verwenden:

```
DISPLAY QMGR ACCTQ
```

- Wenn der Wert ACCTQ(ON) lautet, ist keine weitere Aktion erforderlich. Andernfalls geben Sie den folgenden Befehl ein:

```
ALTER QMGR ACCTQ(ON)
```

- Wenn Clustering verwendet wird, stellen Sie sicher, dass die Abrechnung für den SYSTEM.CLUSTER.TRANSMIT.QUEUE und alle anderen benannten Cluster-Übertragungswarteschlangen, die vom Warteschlangenmanager gehostet werden. So ermitteln Sie, ob die Abrechnung für die Cluster-Übertragungswarteschlangen aktiviert ist:
 - Zeigen Sie die ACCTQ-Einstellung für die Warteschlange(n) mit dem folgenden Befehl an:

```
DISPLAY QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ
```

- Wenn der Wert ACCTQ(ON) lautet, ist keine weitere Aktion erforderlich. Andernfalls geben Sie den folgenden Befehl ein:

```
ALTER QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ(ON)
```

- Überprüfen Sie bei allen anderen Warteschlangen mit hohem Volumen, ob der Wert von ACCTQ auf ON steht, oder setzen Sie ACCTG auf QMGR.

Layout of an SMF type 115 record

You can use this section as a reference for the format of an SMF type 115 record.

The standard layout for SMF records involves three parts:

SMF header

Provides format, identification, and time and date information about the record itself.

Self-defining section

Defines the location and size of the individual data records within the SMF record.

Data records

The actual data from IBM MQ that you want to analyze.

For more information about SMF record formats, see [z/OS MVS System Management Facilities \(SMF\)](#).

Related reference

“The SMF header” on page 343

Use this topic as a reference for the format of the SMF header.

“Self-defining sections” on page 344

Use this topic as a reference for format of the self-defining sections of the SMF record.

“Examples of SMF statistics records” on page 345

Use this topic to understand some example SMF records.

The SMF header

Use this topic as a reference for the format of the SMF header.

Table 34 on page 343 shows the format of SMF record header (SM115).

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM115	SMF record header.	
0	0	Integer	2	SM115LEN	SMF record length.	14A0
2	2		2		Reserved.	
4	4	Integer	1	SM115FLG	System indicator.	5E
5	5	Integer	1	SM115RTY	Record type. The SMF record type, for IBM MQ statistics records this is always 115 (X'73').	73

Table 34. SMF record 115 header description (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
6	6	Integer	4	SM115TME	Time when SMF moved record.	00355575
10	A	Integer	4	SM115DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM115SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)
18	12	Character	4	SM115SSI	IBM MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM115STF	Record subtype.	0002
24	18	Character	3	SM115REL	IBM MQ version.	F6F0F0 (600)
27	1B		1		Reserved	
28	1C	Character	0	SM115END	End of SMF header and start of self-defining section.	

Self-defining sections

Use this topic as a reference for format of the self-defining sections of the SMF record.

A self-defining section of a type 115 SMF record tells you where to find a statistics record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow the header, at fixed offsets from the start of the SMF record. Each statistics record can be identified by an eye-catcher string.

The following types of self-defining section are available to users for type 115 records. Each self-defining section points to statistics data related to one of the IBM MQ components. [Table 35 on page 344](#) summarizes the sources of the statistics, the eye-catcher strings, and the offsets of the self-defining sections from the start of the SMF record header.

Table 35. Offsets to self-defining sections

Source of statistics	Record subtype (SM115STF)	Offset of self-defining section		Eye-catcher of data
		Dec	Hex	
Storage manager	1	100	X'64'	QSST
Log manager	1	116	X'74'	QJST
Message manager	2	36	X'24'	QMST
Data manager	2	44	X'2C'	QIST
No longer used. The self-defining section will be binary zeros.	2	52	X'34'	
Lock manager	2	60	X'3C'	QLST
Db2 manager	2	68	X'44'	Q5ST
Coupling Facility manager	2	76	X'4C'	QEST
Topic manager	2	84	X'54'	QTST
SMDS usage	2	92	X'5C'	QESD

Table 35. Offsets to self-defining sections (continued)

Source of statistics	Record subtype (SM115STF)	Offset of self-defining section		Eye-catcher of data
		Dec	Hex	
Buffer manager - one for each buffer pool	215	36	X'24'	QPST
Channel initiator	231			QWSX
Data manager page set - one for each page set	201	36	X'24'	QIS1
Storage manager	5	36	X'24'	QSPH
Storage manager	6	36	X'24'	QSGM
Storage manager	7	36	X'24'	QSRS
Queues	216	36	X'24'	QQST

Note: Some of the storage manager information in subtype 5, 6 and 7 records is intended only for IBM use. Other self-defining sections that are not listed contain data for IBM use only.

Each self-defining section is two fullwords long and has this format:

```
ssssssssllllnnnn
```

where:

- ssssssss is a fullword containing the offset from the start of the SMF record.
- llll is a halfword giving the length of this data record.
- nnnn is a halfword giving the number of data records in this SMF record.

For more information see, “Examples of SMF statistics records” on page 345.

Note: Always use offsets in the self-defining sections to locate the statistics records.

Examples of SMF statistics records

Use this topic to understand some example SMF records.

Figure 20 on page 346 shows an example of part of the SMF record for subtype 1. Subtype 1 includes the storage manager and log manager statistics records. The SMF record header is shown underlined.

The self-defining section at offset X'64' refers to storage manager statistics and the self-defining section at offset X'74' refers to log manager statistics, both shown in **bold**.

The storage manager statistics record is located at offset X'0000011C' from the start of the header and is X'48' bytes long. There is one set of storage manager statistics, identified by the eye-catcher string QSST. The start of this statistics record is also shown in the example.

The log manager statistics record is located at offset X'00000164' from the start of the header and is X'78' bytes long. There is one set of log manager statistics, identified by the eye-catcher string QJST.

```

000000 02000000 5E730035 55750100 223FD4E5 *...;.....MV*
000010 F4F1D4D8 F0F70001 F6F0F000 000001DC *41MQ07..600...*
000020 00240001 00000000 00000000 00000000 *.....*
000030 00000000 00000000 00000000 0000007C *.....@*
000040 00400001 000000BC 00600001 00000000 *.....*
000050 00000000 00000000 00000000 00000000 *.....*
000060 00000000 0000011C 00480001 00000000 *.....*
000070 00000000 00000164 00780001 00000000 *.....*
000080 00000000 00000000 00000000 00000000 *.....*
.
000110 00000000 00000000 00000000 003C0048 *.....*
000120 D8E2E2E3 0000004F 00000003 00000002 *QSST...|.....*

```

Figure 20. SMF record 115, subtype 1

Figure 21 on page 347 shows an example of part of the SMF record for subtype 2. Subtype 2 includes the statistics records for the message, data, lock, coupling facility, topic, and Db2 managers. The SMF record header is shown underlined; the self-defining sections are shown alternately **bold** and *italic*.

- The self-defining section at offset X'24' refers to message manager statistics. The message manager statistics record is located at offset X'00000064' from the start of the header and is X'48' bytes long. There is one set of these statistics, identified by the eye-catcher string QMST.
- The self-defining section at offset X'2C' refers to data manager statistics. The data manager statistics record is located at offset X'000000AC' from the start of the header and is X'50' bytes long. There is one set of these statistics, identified by the eye-catcher string QIST.
- In releases prior to IBM MQ 9.1.0, the self-defining section at offset X'34' referred to buffer manager statistics. As this SMF record was taken from an IBM MQ 9.1.0 queue manager, the buffer manager self-defining section is set to zeros to indicate that there are no buffer manager statistics. Instead, these statistics are in SMF 115 subtype 215 records.
- The self-defining section at offset X'3C' refers to lock manager statistics. The lock manager statistics record is located at offset X'000000FC' from the start of the header and is X'20' bytes long. There is one set of these statistics, identified by the eye-catcher string QLST.
- The self-defining section at offset X'44' refers to Db2 manager statistics. The Db2 manager statistics record is located at offset X'0000011C' from the start of the header and is X'2A0' bytes long. There is one set of these statistics, identified by the eye-catcher string Q5ST.
- The self-defining section at offset X'4C' refers to coupling facility manager statistics. The coupling facility manager statistics record is located at offset X'000003BC' from the start of the header and is X'1008' bytes long. There is one set of these statistics, identified by the eye-catcher string QEST.
- The self-defining section at offset X'54' refers to topic manager statistics. The topic manager statistics record is located at offset X'000013C4' from the start of the header and is X'64' bytes long. There is one set of these statistics, identified by the eye-catcher string QTST.
- The self-defining section at offset X'5C' is for SMDS statistics. This self-defining section is set to zeros indicating that SMDS is not being used.

```

000000 09F40000 5E730033 4DBE0113 142FD4E5 *.4..;...(. ....MV*
000010 F4F1D4D8 F2F10002 F9F1F000 00001428 *41MQ21..910.....*
000020 00240001 00000064 00480001 000000AC *.....*
000030 00500001 00000000 00000000 000000FC *.....*
000040 00200001 0000011C 02A00001 000003BC *.....*
000050 10080001 000013C4 00640001 00000000 *.....D.....*
000060 00000000 D40F0048 D8D4E2E3 00000000 *...M...QMST...*
000080 00000000 00000000 00000000 00000000 *.....*
000090 00000000 00000000 00000000 00000000 *.....*
0000A0 00000000 00000000 00000000 C90F0050 *.....I..&*
0000B0 D8C9E2E3 00000000 00000000 00000000 *QIST.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 00000000 00000000 00000000 *.....*
0000E0 00000000 00000000 00000000 00000000 *.....*
0000F0 00000000 00000000 00000000 D30F0020 *.....L...*
000100 D8D3E2E3 00000000 00000000 00000000 *QLST.....*
000110 00000000 00000000 00000000 F50F02A0 *.....5...*
000120 D8F5E2E3 00000008 00000000 00000000 *Q5ST.....*
.

```

Figure 21. SMF record 115, subtype 2

Processing type 115 SMF records

Use this topic as a reference for processing type 115 SMF records.

You must process any data you collect from SMF to extract useful information. When you process the data, verify that the records are from IBM MQ and that they are the records you are expecting.

Validate the values of the following fields:

- SM115RTY, the SMF record number, must be X'73' (115)
- SM115STF, the record subtype, must be 0001, 0002, 0005, 0006, 0007, 0201, 0215, or 0231

Reading from the active SMF data sets (or SMF logstreams) is not supported. You must use the SMF program IFASMFDP (or IFASMFDL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed. For more information see [“Using System Management Facility”](#) on page 335.

Details of the structures and fields can be found in IBM MQ SupportPac [MP1B](#).

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records from the sequential data set. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC(CSQ4SMFJ).

Storage manager data records

Use this topic as a reference for storage manager data records.

The format of the storage manager statistics record is described in assembler macro thlqual.SCSQMACS(CSQDQSST).

The data contains information about the number of fixed and variable storage pools that the queue manager has allocated, expanded, contracted, and deleted during the statistics interval, plus the number of GETMAIN, FREEMAIN, and STORAGE requests to z/OS, including a count of those requests that were unsuccessful. Additional information includes a count of the number of times the short-on-storage condition was detected and a count of the number of abends that occurred as a result of that condition.

Additional data about storage usage in the queue manager is produced by class 2 and class 3 statistics trace. While you can view this information, some of it is intended only for IBM use.

- The format of the storage manager pool header statistics record, which is present in subtype 5 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSPH).
- The format of the storage manager getmain statistics record, which is present in subtype 6 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSGM).

- The format of the storage manager region summary record, which is present in subtype 7 records, is described in assembler macro thlqua1.SCSQMACS(CSQDQSRs).

Protokollmanagerdatensätze

Verwenden Sie dieses Thema als Referenz für das Format von Protokollmanagerdatensätzen.

Das Format des Protokollmanagerstatistikdatensatzes wird im Assemblermakro thlqua1.SCSQMACS(CSQDQJST) beschrieben.

In den Statistiken sind diese Zahlen wichtig:

1. Die Gesamtzahl der Protokollschreibanforderungen:

$$N_{\text{logwrite}} = \text{QJSTWRNW} + \text{QJSTWRF}$$

2. Die Gesamtzahl der Protokolleseanforderungen:

$$N_{\text{logread}} = \text{QJSTRBUF} + \text{QJSTRACT} + \text{QJSTRARH}$$

Die Fehlersymptome, die anhand der Protokollmanagerstatistik untersucht werden können, werden in der folgenden Tabelle beschrieben.

<p>Symptom 1 QJSTWTB ist ungleich null.</p> <p>Grund Tasks werden ausgesetzt, während der In-Storage-Puffer in das aktive Protokoll geschrieben wird. Möglicherweise gibt es Probleme beim Schreiben in das aktive Protokoll. Der Parameter OUTBUFF in CSQ6LOGP ist zu klein.</p> <p>Action Schreiben Sie die Probleme auf, die in das aktive Protokoll geschrieben werden. Erhöhen Sie den Wert des Parameters OUTBUFF innerhalb von CSQ6LOGP.</p>
<p>Symptom 2 Das Verhältnis: $\text{QJSTWTL} / N_{\text{logread}}$ ist größer als 1%.</p> <p>Grund Es wurden Protokollesevorgänge eingeleitet, bei denen aus einem Archivprotokoll gelesen werden muss. IBM MQ konnte allerdings kein Dataset zuordnen, da MAXRTU-Datasets bereits zugeordnet waren.</p> <p>Action Erhöhen Sie MAXRTU.</p>

Symptom 3

Das Verhältnis: $QJSTRARH/N_{logread}$ ist größer als normal.

Grund

Die meisten Protokollesesanforderungen sollten aus dem Ausgabepuffer oder dem aktiven Protokoll stammen. Um Anforderungen für ein Zurückstellen zu erfüllen, werden die Arbeitseinheitenwiederherstellungsdatensätze aus dem speicheraktiven Puffer, dem aktiven Protokoll und den archivierten Protokollen gelesen.

Eine lange Arbeitseinheit mit Wiederherstellung, die sich über einen Zeitraum von mehreren Minuten erstreckt, enthält möglicherweise Protokollsätze, die sich über viele verschiedene Protokolle erstrecken. Dies beeinträchtigt die Leistung, da zusätzliche Arbeiten zur Wiederherstellung der Protokollsätze ausgeführt werden müssen.

Action

Ändern Sie die Anwendung, um die Länge einer Arbeitseinheit mit Wiederherstellung zu reduzieren. Darüber hinaus sollten Sie die Größe des aktiven Protokolls erhöhen, um die Möglichkeit einer einzelnen Arbeitseinheit zu reduzieren, die sich auf mehr als ein Protokoll ausbreitet.

Andere Zeiger

Das Verhältnis $N_{logread} / N_{logwrite}$ gibt einen Hinweis darauf, wie viel Arbeit zurückgesetzt werden muss.

Symptom 4

$QJSTLLCP$ ist mehr als 10 Stunden pro Stunde.

Grund

Auf einem ausgelasteten System erwarten Sie in der Regel 10 Prüfpunkte pro Stunde. Wenn der Wert für $QJSTLLCP$ größer als dieser Wert ist, weist er auf ein Problem in der Konfiguration des Warteschlangenmanagers hin.

Der wahrscheinlichste Grund hierfür ist, dass der Parameter $LOGLOAD$ in $CSQ6SYSP$ zu klein ist. Das andere Ereignis, das einen Prüfpunkt verursacht, ist, wenn ein aktives Protokoll voll ist, und wechselt zu der nächsten aktiven Protokolldatei. Wenn Ihre Protokolle zu klein sind, kann dies zu häufigen Prüfpunkten führen.

$QJSTLLCP$ ist die Anzahl der Gesamtanzahl der Prüfpunkte.

Action

Erhöhen Sie den Wert für den Parameter $LOGLOAD$ oder erhöhen Sie die Größe der Protokoll Datensätze nach Bedarf.

Symptom 5

$QJSTCmpFail > 0$ oder $QJSTCmpComp$ nicht viel kleiner als $QJSTCmpUncmp$

Grund

Der Warteschlangenmanager kann die Protokollsätze nicht wesentlich komprimieren.

$QJSTCmpFail$ ist die Anzahl der Male, die der Warteschlangenmanager nicht in der Lage war, eine Reduzierung der Satzlänge zu erreichen. Vergleichen Sie die Zahl mit $QJSTCmpReq$ (Anzahl der Komprimierungsanforderungen), um festzustellen, ob die Anzahl der Fehler signifikant ist.

$QJSTCmpComp$ ist die Summe der komprimierten Byte, die in das Protokoll geschrieben werden, und $QJSTCmpUncmp$ ist die Summe der Byte vor der Komprimierung. Die Summe enthält keine Byte, die für Protokollsätze geschrieben wurden, die nicht für die Komprimierung auswählbar waren. Wenn die Zahlen ähnlich sind, hat die Komprimierung wenig Nutzen erzielt.

Action

Schalten Sie die Protokollkomprimierung aus. Setzen Sie den Befehl `SET LOG COMPLOG (NONE)` ab. Weitere Informationen finden Sie im Befehl `SET LOG`.

Anmerkung: In der ersten Gruppe von Statistiken, die nach dem Systemstart erstellt wurden, kann es aufgrund der Auflösung von In-Flight-Einheiten der Wiederherstellung zu einer erheblichen Protokollaktivität kommen.

Änderungen an zHyperWrite

V 9.4.0

Ab IBM MQ 9.3.5 ändert die zHyper-Schreibverarbeitung das Verhalten der QJSTHWC- und QJSTHWE-Statistiken.

Vor IBM MQ 9.3.5:

- QJSTHWC ist die Anzahl der im SMF-Intervall verwendeten Protokolldateien, die sich auf zHyper-schreibfähigen Datenträgern befinden. Die zHyper-Schreibfunktionalität eines Datenträgers wird beim Start des Warteschlangenmanagers abgerufen und kann sich im Laufe der Zeit ändern, sodass diese Informationen möglicherweise veraltet sind.
- QJSTHWE ist die Anzahl der im SMF-Intervall verwendeten Protokolldateien, in die geschrieben wird, wenn zHyperWrite aktiviert ist. Dies geschieht, wenn sich die Protokolldatei auf einem zHyper-schreibfähigen Datenträger befindet und wenn der Warteschlangenmanager für zHyperWrite aktiviert wurde, indem ZHYWRITE (YES) festgelegt wurde.

Ab IBM MQ 9.3.5:

- QJSTHWC ist die Anzahl der im SMF-Intervall verwendeten Protokolldateien, die sich auf zHyper-schreibfähigen Datenträgern befinden. Die zHyper-Schreibfunktionalität eines Datenträgers wird beim Start des Warteschlangenmanagers abgerufen und kann sich im Laufe der Zeit ändern, sodass diese Informationen möglicherweise veraltet sind.
- QJSTHWE ist die Anzahl der im SMF-Intervall verwendeten Protokolldateien, in die geschrieben wird, wenn zHyperWrite aktiviert ist. Dies ist der Fall, wenn der Warteschlangenmanager für zHyperWrite durch Festlegen von ZHYWRITE (YES) aktiviert wurde.



Achtung: Die zHyper-Schreibfunktion eines Datenträgers kann sich im Laufe der Zeit ändern. Ab IBM MQ 9.3.5 kann dies zu Szenarios führen, in denen QJSTHWE größer als QJSTHWC ist.

zHyperLink-Statistik

V 9.4.0

Ab IBM MQ 9.4.0 wurden Protokollmanagerstatistiken hinzugefügt, um die Leistung von zHyperLink zu überwachen.

Die folgenden Statistiken wurden zur QJST hinzugefügt:

Name	Typ	Länge Byte	Beschreibung
QJSTHLSCIW	Integer ohne Vorzeichen	4	Anzahl der Einzelseitenschreibvorgänge, die zHyperLink angefordert haben.
QJSTHLMCIW	Integer ohne Vorzeichen	4	Anzahl der mehrseitigen Schreibvorgänge, die zHyperLink angefordert haben.
QJSTHLCICNTW	Integer ohne Vorzeichen	4	Die Anzahl der Seiten, die geschrieben wurden, als der Schreibvorgang zHyperLink angefordert hat.
QJSTHLIS (QJSTHLSCIS)	Integer ohne Vorzeichen	4	Anzahl der Seitenschreibvorgänge, die zHyperLink erfolgreich verwendet haben. Das heißt, der Schreibvorgang wurde synchron ausgeführt.
QJSTHLMCIS	Integer ohne Vorzeichen	4	Anzahl der mehrseitigen Schreibvorgänge, die zHyperLink erfolgreich verwendet haben. Das heißt, der Schreibvorgang wurde synchron ausgeführt.

Name	Typ	Länge Byte	Beschreibung
QJSTHLICINTS	Integer ohne Vorzeichen	4	Die Anzahl der Seiten, die erfolgreich mit zHyperLink geschrieben wurden. Das heißt, der Schreibvorgang wurde synchron ausgeführt.
QJSTHLSCIF	Integer ohne Vorzeichen	4	Die Anzahl der Einzelseitenschreibvorgänge, die versucht haben, zHyperLink zu verwenden, aber der Schreibvorgang konnte nicht synchron ausgeführt werden. Stattdessen wurde der Schreibvorgang erfolgreich asynchron ausgeführt.
QJSTHLMCIF	Integer ohne Vorzeichen	4	Die Anzahl der mehrseitigen Schreibvorgänge, die versucht haben, zHyperLink zu verwenden, aber der Schreibvorgang konnte nicht synchron ausgeführt werden. Stattdessen wurde der Schreibvorgang erfolgreich asynchron ausgeführt.
QJSTHLICINTF	Integer ohne Vorzeichen	4	Die Anzahl der geschriebenen Seiten, bei denen der Schreibvorgang versucht hat, zHyperLink zu verwenden, der Schreibvorgang jedoch nicht synchron ausgeführt werden konnte. Stattdessen wurde der Schreibvorgang erfolgreich asynchron ausgeführt.
QJSTHLSCI- CONCNT	Integer ohne Vorzeichen	4	Anzahl der Schreibvorgänge für einzelne Seiten, bevor eine zHyperLink-Schreibsitzung aufgebaut wurde. Siehe Anmerkung „1” auf Seite 352.
QJSTHLMCI- KONCNT	Integer ohne Vorzeichen	4	Anzahl der mehrseitigen Schreibvorgänge, bevor eine zHyperLink-Schreibsitzung aufgebaut wurde. Siehe Anmerkung „1” auf Seite 352.
QJSTHLKON- CICNT	Integer ohne Vorzeichen	4	Anzahl der Seiten, die geschrieben wurden, bevor eine zHyperLink-Verbindung hergestellt wurde. Siehe Anmerkung „1” auf Seite 352.
QJSTHLSCIWT- MAX	Integer ohne Vorzeichen	8	Längste zHyperLink-Schreibzeit im Intervall, einzelne Seite.
QJSTHLSCIWT- MIN	Integer ohne Vorzeichen	8	Kürzeste zHyperLink-Schreibzeit im Intervall, einzelne Seite.
QJSTHLSCIWT- TOT	Integer ohne Vorzeichen	8	Gesamte zHyperLink-Schreibzeit im Intervall, einzelne Seite.
QJSTHLMCIWT- MAX	Integer ohne Vorzeichen	8	Längste zHyperLink-Schreibzeit im Intervall, mehrere Seiten.
QJSTHLMCIWT- MIN	Integer ohne Vorzeichen	8	Kürzeste zHyperLink-Schreibzeit im Intervall, mehrere Seiten.
QJSTHLMCIWT- TOT	Integer ohne Vorzeichen	8	Die gesamte zHyperLink-Schreibzeit im Intervall, mehrere Seiten.
QJSTHLIOSQU	Integer ohne Vorzeichen	16	Quadratsumme der zHyperLink-Schreibzeiten für Einzelseitenschreibvorgänge.
QJSTHLC	Integer ohne Vorzeichen	4	Die Anzahl der neuen Protokolle, die in diesem SMF-Intervall verwendet werden und zHyperLink-fähig sind.

Name	Typ	Länge Byte	Beschreibung
QJSTHLE	Integer ohne Vorzeichen	4	Die Anzahl der neuen Protokolle, die in diesem SMF-Intervall verwendet werden und für zHyper-Link aktiviert sind.

Anmerkung:

1. Beim Wechsel zu einer neuen Kopie des aktiven Protokolls kann es einen bestimmten Zeitraum geben, bevor eine zHyperLink-Schreibsitzung mit der DASD-Einheit aufgebaut wird.

z/OS Message manager data records

Use this topic as a reference for message manager data records.

The format of the message manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQMST)`.

The data gives you counts of different IBM MQ API requests.

z/OS Data manager data records

Use this topic as a reference for the format of the Data Manager data records.

The format of the data manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQIST)`.

The data gives you counts of different object requests.

z/OS Data manager page set data records

Use this section as a reference for the format of the data manager page set data records

The format of the data manager page set statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQIS1)`.

The page set usage information helps to facilitate better management of local queues within a queue manager by recording information such as page set input-output rates and highest usage.

The data provides the same basic page set information as output by the MQSC **DISPLAY USAGE TYPE (PAGESET)** command, or the PCF Inquire Usage (**MQCMD_INQUIRE_USAGE**) command.

For example:

- The total pages
- The current used pages
- Unused persistent and nonpersistent pages
- Expansion method
- Number of extends
- Number of stripes

The data also provides some performance indicators, together with other performance information. For example:

- How many times deferred write, immediate write, and read page I/O requests happened during the SMF interval,
- The number of pages moved, elapsed time and number of read and write operations.
- How many pages written in checkpoints.
- Has expansion occurred during the SMF interval?
- How many times the page set became full.
- An indication of where new space is being allocated within the page set.

From the information displayed, you should be able to understand the general status of each page set, and consider whether you need to retune the system.

Related reference

[“The SMF header” on page 343](#)

Use this topic as a reference for the format of the SMF header.

[“Self-defining sections” on page 344](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 345](#)

Use this topic to understand some example SMF records.

Buffer manager data records

Use this topic as a reference for the format of buffer manager data records.

The format of the buffer manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQPST)`.

Note: Buffer manager statistics records will only be created for buffer pools that are defined. If a buffer pool is defined but not used then no values will be set and its buffer manager statistics record will not contain any data.

For information about efficiently managing your buffer pools, see [“Managing your buffer pools” on page 354](#).

When interpreting the statistics, you are recommended to consider the following factors because the values of these fields can be used to improve the performance of your system:

1. If QPSTSOS, QPSTDMC, or QPSTIMW is greater than zero, you should either increase the size of the buffer pool or reallocate the page sets to different buffer pools.
 - QPSTSOS is the number of times that there were no buffers available for page get requests. If QPSTSOS ever becomes nonzero, it shows that IBM MQ is under severe stress. The buffer pool size should be increased. If increasing the buffer pool size does not make the value of QPSTSOS zero, there might be I/O contention on the DASD page sets.
 - QPSTDMC is the number of updates that were performed synchronously because there was either more than 95% of the pages in the buffer pool waiting for write I/O, or there was less than 5% of the buffer pool available for read requests. If this number is not zero, the buffer pool might be too small and should be enlarged. If increasing the buffer pool size does not reduce QPSTDMC to zero, there might be I/O contention on the DASD page sets.
 - QPSTIMW is a count of the number of times pages were written out synchronously. If QPSTDMC is zero, QPSTIMW is the number of times pages were found on the queue waiting for write I/O that had been there for at least two checkpoints.
2. For buffer pool zero and buffer pools that contain short-lived messages:
 - QPSTDWT should be zero, and the percentage QPSTCBSL/QPSTNBUF should be greater than 15%.
QPSTDWT is the number of times the asynchronous write processor was started because there was either more than 85% of the pages in the buffer pool waiting for write I/O, or there was less than 15% of the buffer pool available for read requests. Increasing the buffer pool size should reduce this value. If it does not, the pattern of access is one of long delays between puts and gets.
 - QPSTTPW might be greater than zero due to checkpointing activity.
 - QPSTRIO should be zero unless messages are being read from a page set after the queue manager is restarted.

The ratio of QPSTRIO to QPSTGETP shows the efficiency of page retrieval within the buffer pool. Increasing the buffer pool size should decrease this ratio and, therefore, increase the page retrieval efficiency. If this does not happen, it indicates that pages are not being frequently reaccessed. This implies a transaction pattern where there is a long delay between messages being put and then later retrieved.

The ratio of QPSTGETN to QPSTGETP indicates the number of times an empty page, as opposed to a non-empty page, has been requested. This ratio is more an indication of transaction pattern, than a value that can be used to tune the system.

- If QPSTSTL has a value greater than zero, this indicates that pages that have not been used before are now being used. This might be caused by an increased message rate, messages not being processed as fast as they were previously (leading to a buildup of messages), or larger messages being used.

QPSTSTL is a count of the number of times a page access request did not find the page already in the buffer pool. Again, the lower the ratio of QPSTSTL to (QPSTGETP + QPSTGETN) is, the higher the page retrieval efficiency. Increasing the buffer pool size should decrease this ratio but, if it does not, it is an indication that there are long delays between puts and gets.

- You are recommended to have sufficient buffers to handle your peak message rate.
3. For buffer pools with long-lived messages, where there are more messages than can fit into the buffer pool:

- $(QPSTRIO + QPSTWIO) / \text{Statistics interval}$ is the I/O rate to page sets. If this value is high, you should consider using multiple page sets on different volumes to allow I/O to be carried out in parallel.
- Over the period of time that the messages are processed (for example, if messages are written to a queue during the day and processed overnight) the number of read I/Os (QPSTRIO) should be approximately the total number of pages written (QPSTTPW). This shows that one page is read for every page written.

If QPSTRIO is much larger than QPSTTPW, this shows that pages are being read in multiple times. This might be a result of the application using MQGET by *MsgId* or *CorrelId* when the queue is not indexed, or browsing messages on the queue using get next.

The following actions might relieve this problem:

- a. Increase the size of the buffer pool so that there are enough pages to hold the queue, in addition to any changed pages.
- b. Use the INDXTYPE queue attribute, which allows a queue to be indexed by *MsgId* or *CorrelId* and eliminates the need for a sequential scan of the queue.
- c. Change the design of the application to eliminate the use of MQGET with *MsgId* or *CorrelId*, or the get next with browse option.

Note: Applications using long-lived messages typically process the first available message and do not use MQGET with *MsgId* or *CorrelId*, and they might browse only the first available message.

- d. Move page sets to a different buffer pool to reduce contention between messages from different applications.

Managing your buffer pools

To manage your buffer pools efficiently, you must consider the factors that affect the buffer pool I/O operations and also the statistics associated with the buffer pools.

The following factors affect buffer pool I/O operations.

- If a page containing the required data is not found in the buffer pool, it is read in synchronously to an available buffer from its DASD page set.
- Whenever a page is updated, it is put on an internal queue of pages to be (potentially) written out to DASD. This means that the buffer used by that page is unavailable for use by any other page until the buffer has been written to DASD.
- If the number of pages queued to be written to DASD exceeds 85% of the total number of buffers in the pool, an asynchronous write processor is started to put the buffers to DASD.

Similarly, if the number of buffers available for page get requests become less than 15% of the total number of buffers in the pool, the asynchronous write processor is started to perform the write I/O operations.

The write processor stops when the number of pages queued to be written to DASD has fallen to 75% of the total number of buffers in the pool.

- If the number of pages queued for writing to DASD exceeds 95% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

Similarly, if the number of buffers available for page get requests becomes less than 5% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

- If the number of buffers available for page get requests ever reaches zero, a transaction that encounters this condition is suspended until the asynchronous write processor has finished.
- If a page is frequently updated, the page spends most of its time on the queue of pages waiting to be written to DASD. Because this queue is in least recently used order, it is possible that a frequently updated page placed on this least recently used queue is never written out to DASD. For this reason, at the time of update, if the page is found to have been waiting on the write operation to DASD queue for at least two checkpoints, it is synchronously written to DASD. Updating occurs at checkpoint time and is suspended until the asynchronous write processor has finished.

The aim of this algorithm is to maximize the time pages spend in buffer pool memory while allowing the system to function if the system load puts the buffer pool usage under stress.

Lock manager data records

Use this topic as a reference to the format of the lock manager data records.

The format of the lock manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQLST)`.

The records contain data about the following information:

- The number of lock get requests and lock release requests.
- The number of times a lock get request determined that the requested lock was already held.

Db2 manager data records

Use this topic as a reference to the format of the Db2 manager data records.

The format of the Db2 manager statistics record is described in the following table and in assembler macro `thlqual.SCSQMACS(CSQDQ5ST)` and C header file `thlqual.SCSQC370(CSQDSMFC)`. The field names in C are all in lowercase, for example `q5st`, `q5stid`.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	668	Q5ST	Db2 manager statistics
0	0	Bitstring	2	Q5STID	Control block identifier
2	2	Integer	2	Q5STLL	Control block length
4	4	Character	4	Q5STEYEC	Control block eye catcher
8	8	Character	660	Q5STZERO	QMST part cleared on occasion
8	8	Integer	4	NUMTASK	Number of server tasks
12	C	Integer	4	ACTTASK	Number of active server tasks
16	10	Integer	4	CONNCNT	Number of connect requests
20	14	Integer	4	DISCCNT	Number of disconnect requests

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
24	18	Integer	4	DHIGMAX	Max. request queue depth
28	1C	Integer	4	ABNDCNT	Number of Db2SRV task abends
32	20	Integer	4	REQUCNT	Number of requests queued
36	24	Integer	4	DEADCNT	Number of deadlock timeouts
40	28	Integer	4	DELECNT	Number of delete requests
44	2C	Integer	4	LISTCNT	Number of list requests
48	30	Integer	4	READCNT	Number of read requests
52	34	Integer	4	UPDTCNT	Number of update requests
56	38	Integer	4	WRITCNT	Number of write requests
60	3C	Integer	4	SCSSEL	SCST (shared-channel-status) selects
64	40	Integer	4	SCSINS	SCST inserts
68	44	Integer	4	SCSUPD	SCST updates
72	48	Integer	4	SCSDEL	SCST deletes
76	4C	Integer	4	SSKSEL	SSKT (shared-sync-key) selects
80	50	Integer	4	SSKINS	SSKT inserts
84	54	Integer	4	SSKDEL	SSKT deletes
88	58	Integer	4	SCSBFTS	SCST number of times buffer too small
92	5C	Integer	4	SCSMAXR	SCST maximum rows on query
96	60	Integer	4	* (2)	Reserved
104	68	Character	8	DELETCUW	Cumulative STCK difference - Thread delete
112	70	Character	8	DELETMXW	Maximum STCK difference - Thread delete
120	78	Character	8	DELESCUW	Cumulative STCK difference - SQL delete
128	80	Character	8	DELESMXW	Maximum STCK difference - SQL delete
136	88	Character	8	LISTTCUW	Cumulative STCK difference - Thread list
144	90	Character	8	LISTTMXW	Maximum STCK difference - Thread list
152	98	Character	8	LISTSCUW	Cumulative STCK difference - SQL list
160	A0	Character	8	LISTSMXW	Maximum STCK difference - SQL list
168	A8	Character	8	READTCUW	Cumulative STCK difference - Thread read
176	B0	Character	8	READTMXW	Maximum STCK difference - Thread read
184	B8	Character	8	READSCUW	Cumulative STCK difference - SQL read
192	C0	Character	8	READSMXW	Maximum STCK difference - SQL read
200	C8	Character	8	UPDTTCUW	Cumulative STCK difference - Thread update
208	D0	Character	8	UPDTTMXW	Maximum STCK difference - Thread update
216	D8	Character	8	UPDTSCUW	Cumulative STCK difference - SQL update

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
224	E0	Character	8	UPDTSMXW	Maximum STCK difference - SQL update
232	E8	Character	8	WRITTCUW	Cumulative STCK difference - Thread write
240	F0	Character	8	WRITTMXW	Maximum STCK difference - Thread write
248	F8	Character	8	WRITSCUW	Cumulative STCK difference - SQL write
256	100	Character	8	WRITSMXW	Maximum STCK difference - SQL write
264	108	Character	8	SCSSTCUW	Cumulative STCK difference - Thread select
272	110	Character	8	SCSSTMXW	Maximum STCK difference - Thread select
280	118	Character	8	SCSSSCUW	Cumulative STCK difference - SQL select
288	120	Character	8	SCSSSMXW	Maximum STCK difference - SQL select
296	128	Character	8	SCSITCUW	Cumulative STCK difference - Thread insert
304	130	Character	8	SCSITMXW	Maximum STCK difference - Thread insert
312	138	Character	8	SCSISCUW	Cumulative STCK difference - SQL insert
320	140	Character	8	SCSISMXW	Maximum STCK difference - SQL insert
328	148	Character	8	SCSUTCUW	Cumulative STCK difference - Thread update
336	150	Character	8	SCSUTMXW	Maximum STCK difference - Thread update
344	158	Character	8	SCSUSCUW	Cumulative STCK difference - SQL update
352	160	Character	8	SCSUSMXW	Maximum STCK difference - SQL update
360	168	Character	8	SCSDTCUW	Cumulative STCK difference - Thread delete
368	170	Character	8	SCSDTMXW	Maximum STCK difference - Thread delete
376	178	Character	8	SCSDSCUW	Cumulative STCK difference - SQL delete
384	180	Character	8	SCSDSMXW	Maximum STCK difference - SQL delete
392	188	Character	8	SSKSTCUW	Cumulative STCK difference - Thread select
400	190	Character	8	SSKSTMXW	Maximum STCK difference - Thread select
408	198	Character	8	SSKSSCUW	Cumulative STCK difference - SQL select
416	1A0	Character	8	SSKSSMXW	Maximum STCK difference - SQL select
424	1A8	Character	8	SSKITCUW	Cumulative STCK difference - Thread insert
432	1B0	Character	8	SSKITMXW	Maximum STCK difference - Thread insert
440	1B8	Character	8	SSKISCUW	Cumulative STCK difference - SQL insert
448	1C0	Character	8	SSKISMXW	Maximum STCK difference - SQL insert
456	1C8	Character	8	SSKDTCUW	Cumulative STCK difference - Thread delete
464	1D0	Character	8	SSKDTMXW	Maximum STCK difference - Thread delete
472	1D8	Character	8	SSKDSCUW	Cumulative STCK difference - SQL delete
480	1E0	Character	8	SSKDSMXW	Maximum STCK difference - SQL delete
488	1E8	Integer	4	LMSSEL	Number of Db2 BLOB read requests

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
492	1EC	Integer	4	LMSINS	Number of Db2 BLOB insert requests
496	1F0	Integer	4	LMSUPD	Number of Db2 BLOB update requests
500	1F4	Integer	4	LMSDEL	Number of Db2 BLOB delete requests
504	1F8	Integer	4	LMSLIS	Number of Db2 BLOB list requests
508	1FC	64 bit integer	8	LMSSTCUW	Total elapsed time for all thread read BLOB requests
516	204	64 bit integer	8	LMSSTMXW	Maximum elapsed time for a thread read BLOB request
524	20C	64 bit integer	8	LMSSSCUW	Total elapsed time for all SQL read BLOB requests
532	214	64 bit integer	8	LMSSSMXW	Maximum elapsed time for an SQL read BLOB request
540	21C	64 bit integer	8	LMSITCUW	Total elapsed time for all thread insert BLOB requests
548	224	64 bit integer	8	LMSITMXW	Maximum elapsed time for a thread insert BLOB request
556	22C	64 bit integer	8	LMSISCUW	Total elapsed time for all SQL insert BLOB requests
564	234	64 bit integer	8	LMSISMXW	Maximum elapsed time for an SQL insert BLOB request
572	23C	64 bit integer	8	LMSUTCW	Total elapsed time for all thread update BLOB requests
580	244	64 bit integer	8	LMSUTMXW	Maximum elapsed time for a thread update BLOB request
588	24C	64 bit integer	8	LMSUSCUW	Total elapsed time for all SQL update BLOB requests
596	254	64 bit integer	8	LMSUSMXW	Maximum elapsed time for an SQL update BLOB request
604	25C	64 bit integer	8	LMSDTCW	Total elapsed time for all thread delete BLOB requests
612	264	64 bit integer	8	LMSDTMXW	Maximum elapsed time for a thread delete BLOB request
620	26C	64 bit integer	8	LMSDSCW	Total elapsed time for all SQL delete BLOB requests
628	274	64 bit integer	8	LMSDSMXW	Maximum elapsed time for an SQL delete BLOB request
636	27C	64 bit integer	8	LMSLTCW	Total elapsed time for all thread list BLOB requests
644	284	64 bit integer	8	LMSLTMXW	Maximum elapsed time for a thread list BLOB request

Offset: Dec	Offset: Hex	Type	Len	Name	Description
652	28C	64 bit integer	8	LMSLSCUW	Total elapsed time for all SQL list BLOB requests
660	294	64 bit integer	8	LMSLSMXW	Maximum elapsed time for an SQL list BLOB request

The data contains counts for each request type that the Db2 resource manager supports. For these request types, maximum and cumulative elapse times are kept for the following:

- The time spent in the Db2 resource manager as a whole (called the thread time).
- The time that was spent performing the RRSF and SQL parts of the request (a subset of the thread time called the SQL time).

Information is also provided for:

- The number of server tasks attached.
- The maximum overall request depth against any of the server tasks.
- The number of times any of the server task requests terminated abnormally.

If the abnormal termination count is not zero, a requeue count is provided indicating the number of queued requests that were requeued to other server tasks as a result of the abnormal termination.

If the average thread time is significantly greater than the average SQL time, this might indicate that thread requests are spending an excessive amount of time waiting for a server task to process the SQL part of the request. If this is the case, examine the DHIGMAX field and, if the value is greater than one, consider increasing the number of Db2 server tasks specified in the QSGDATA parameter of the CSQ6SYSP system parameter macro.

Coupling facility manager data records

Use this topic as a reference to the format of the coupling facility manager data records.

The format of the coupling facility manager statistics record is described in the following table and in assembler macro thlqual.SCSQMACS(CSQDQEST) and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qest, qestid.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	4104	QEST	CF manager statistics
0	0	Bitstring	2	QESTID	Control block identifier
2	2	Integer	2	QESTLL	Control block length
4	4	Character	4	QESTEYEC	Control block eye catcher
8	8	Character	4096	QESTZERO	QEST part cleared on occasion
8	8	Character	64	QESTSTUC (0:63)	Array (one entry per structure)
8	8	Character	12	QESTSTR	Structure name
20	14	Integer	4	QESTSTRN	Structure number
24	18	Integer	4	QESTCSEC	Number of IXLLSTE calls

Offset: Dec	Offset: Hex	Type	Len	Name	Description
28	1C	Integer	4	QESTCMEC	Number of IXLLSTM calls
32	20	Character	8	QESTSSTC	Time spent doing IXLLSTE calls
40	28	Character	8	QESTMSTC	Time spent doing IXLLSTM calls
48	30	Integer	4	QESTRSEC	Number of IXLLSTE redrives
52	34	Integer	4	QESTRMEC	Number of IXLLSTM redrives
56	38	Integer	4	QESTSFUL	Number of structure fulls
60	3C	Integer	4	QESTMNUS	Maximum number of entries in use
64	40	Integer	4	QESTMLUS	Maximum number of elements in use
68	44	Character	4	*	Reserved
4104	1008	Character	0	*	End of control block

The data contains information for each coupling facility list structure, including the CSQ_ADMIN structure, that the queue manager could connect to during the statistics interval. The information for each structure includes the following:

- The number of and cumulative elapsed times for IXLLSTE and IXLLSTM requests.
- The number of times a request had to be retried because of a timeout.
- The number of times a 'structure full' condition occurred.

Topic manager data records

Use this topic as a reference to the format of the topic manager data records.

The format of the topic manager statistics record is described in the following table and in assembler macro thlqual.SCSQMACS(CSQDQTST) and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qtst, qtstid.

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	96	QTST	Topic manager statistics
0	0	Bitstring	2	QTSTID	Control block identifier
2	2	Integer	2	QTSTLL	Control block length
4	4	Character	4	TESTEYEC	Control block eye catcher
8	8	Character	88	QTSTZERO	QTST part cleared on occasion
8	8	Integer	4	QTSTSTOT	Total subscription requests
12	0C	Integer	4	QTSTSDUR	Durable subscription requests
16	10	Integer	4	QTSTSHIG (1:3)	Subscription high water mark array (API, ADMIN, PROXY)
28	1C	Integer	4	QTSTSLOW (1:3)	Subscription low water mark array (API, ADMIN, PROXY)

Table 38. Topic manager statistics record (QTST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
40	28	Integer	4	QTSTSEXP	Subscriptions expired
44	2C	Integer	4	QTSTTMSG	Total messages put to Sub queue
48	30	Integer	4	QTSTSPHW	Single publish subscriber high water mark
52	34	Integer	4	QTSTPTOT (1:3)	Total Publication requests (API, ADMIN, PROXY)
64	40	Integer	4	QTSTPTHI	Total publish high water mark
68	44	Integer	4	QTSTPTLO	Total publish low water mark
72	48	Integer	4	QTSTPNOS	Count of publishes to no subscriber
76	4C	Integer	4	*	Reserved
80	50	Bitstring	8	QTSTETHW	Elapse time HW on publish
88	58	Bitstring	8	QTSTETTO	Elapse time total on publish

Coupling facility manager SMDS data records

Use this topic as a reference to the format of the coupling facility manager shared message data set (SMDS) data records.

The format of the coupling facility manager shared message data set (SMDS) statistics record is described in assembler macro `thlqual.SCSQMACS (CSQDQESD)` and in C header file `thlqual.SCSQC370 (CSQDSMFC)`.

The statistics provide information about the utilization of the owned shared message data set, I/O activity for the group of shared message data sets, and SMDS buffer utilization.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Layout of channel initiator SMF type 115 records

The layout of channel initiator statistics data (SMF type 115, subtype 231) records is described in this topic.

Self-defining section

The self-defining section for the channel initiator statistics data follows the standard SMF header. It is structured in the standard triplet format. The format of the triplets is described in structure `qwsx` in the C programming language header file `thlqual.SCSQC370 (CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS (CSQDQWSX)`.

Table 39 on page 361 shows the format of the self-defining section.

Table 39. Structure of the channel initiator statistics self-defining section

Offset: Dec	Offset: Hex	Type	Length	Name	Description
0	0	Integer	4	QWSX0PSO	Offset from the start of the SMF record to the first instrumentation standard header (QWHS)

Table 39. Structure of the channel initiator statistics self-defining section (continued)

Offset: Dec	Off- set: Hex	Type	Length	Name	Description
4	4	Integer	2	QWSX0PSL	Length of the QWHS
6	6	Integer	2	QWSX0PSN	Number of instances of QWHS
8	8	Integer	4	QWSX0R10	Offset from the start of the SMF record to the first channel initiator control information block (QCCT)
12	C	Integer	2	QWSX0R1L	Length of the QCCT
14	E	Integer	2	QWSX0R1N	Number of instances of QCCT
16	10	Integer	4	QWSX0R20	Offset from the start of the SMF record to the first dispatcher task block (QCT_DSP)
20	14	Integer	2	QWSX0R2L	Length of the QCT_DSP
22	16	Integer	2	QWSX0R2N	Number of instances of QCT_DSP
24	18	Integer	4	QWSX0R30	Offset from the start of the SMF record to the first adapter task block (QCT_ADP)
28	1C	Integer	2	QWSX0R3L	Length of the QCT_ADP
30	1E	Integer	2	QWSX0R3N	Number of instances of QCT_ADP
32	20	Integer	4	QWSX0R40	Offset from the start of the SMF record to the first SSL task block (QCT_SSL)
36	24	Integer	2	QWSX0R4L	Length of the QCT_SSL
38	26	Integer	2	QWSX0R4N	Number of instances of QCT_SSL
40	28	Integer	4	QWSX0R50	Offset from the start of the SMF record to the first DNS task block (QCT_DNS)
44	2C	Integer	2	QWSX0R5L	Length of the QCT_DNS
46	2E	Integer	2	QWSX0R5N	Number of instances of QCT_DNS

Typically one record contains all the data. If there are a large number of dispatchers, adapters, or SSL tasks, the data is split over more than one record.

If this happens, the count of instances of some type of tasks can be zero, and information about a group of tasks can be spread across multiple records. The channel initiator control information block (QCCT) is only present in the first record. For example the data could be split between two SMF records like this:

Table 40. Example data

Count	First record	Last record
QWHS	1	1
QCCT	1	0
QCT_DSP	50	5
QCT_ADP	0	10
QCT_SSL	0	3
QCT_DNS	0	1

This example shows that there were 55 dispatcher TCBs running during the SMF interval.

Instrumentation standard header (QWHS)

The format of the QWHS is described in structure `qwhs` in the C programming language header file `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.SCSQMACS(CSQDQWHS)`. It contains the following key fields that are relevant to channel initiator SMF 115 records:

Name	Length	Description
QWHSNDA	1 byte	Number of self-defining sections
QWHSSSID	4 bytes	Subsystem name
QWHSSMFC	1 bit	Indicates whether there are multiple SMF records containing information for this interval. If this bit is on, information for this interval is continued in further SMF records. If this bit is off, this is the last or only record. The subsystem ID in QWHSSSID, and the SMF interval start time in QWHSTIME, can be used to group multiple records for the same interval.
QWHSTIME	8 bytes	Local time of the start of the interval in STCK format
QWHSDURN	8 bytes	Duration from the start of the interval to the end of the interval in STCK format
QWHSSTCK	8 bytes	End of the interval in UTC in STCK format

Channel initiator statistics data records

Use this topic as a reference for channel initiator statistics data records.

The format of the channel initiator statistics data record contains two parts:

- The first part is the channel initiator control information block, described in assembler macro `th1qua1.SCSQMACS(CSQDQCCT)`. For further information, see [“Channel initiator control information block” on page 364](#).
- The second part is the channel initiator task block, described in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

The channel initiator task block contains information about the four types of task within the CHINIT. For further information, see:

- [“Dispatcher tasks” on page 365](#)
- [“Adapter tasks” on page 366](#)
- [“Domain Name Server \(DNS\) task” on page 367](#)
- [“SSL tasks” on page 368](#)

Each task includes:

- The elapsed time that the task spent processing requests in the interval (*qcteltm*)
- The CPU time used by the task in the interval, which is made up of CPU used while processing requests and CPU used between requests (*qctcptm*)
- The total wait time of this task in the interval (*qctwttm*)
- The number of requests in the interval (*qctreqn*)

You can use this information to see how busy the task was, and determine whether you need to add more tasks based on the analysis.

For TLS and DNS tasks, the duration of the longest request (*qctlgdu*, *qctlsdu*) and the time of day when this occurred (*qctlgdm*, *qctlsdm*) are also included.

These can be useful to identify when channel requests took a long time. For example, a DNS lookup request going to a server outside of your enterprise taking seconds rather than milliseconds.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time.

The example accounting data in the following tasks has been formatted using IBM MQ SupportPac MP1B.

Both of the parts are also described in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`. Note that the field names in C are all in lowercase, for example, *qcct*, *qct_adp*.

Channel initiator control information block

Use this topic as a reference for the channel initiator control information block.

The channel initiator control information block contains basic information for this CHINIT, including:

- CHINIT job name (*qcctjobn*)
- QSG name if it is in a queue sharing group (*qcctqsgn*)
- Peak number used of current channels (*qcctnocc*)
- Peak number used of active channels (*qcctnoac*)
- MAXCHL - maximum permitted current channels (*qcctmxcc*)
- ACTCHL - maximum permitted active channels (*qcctmxac*)
- TCPCHL - maximum permitted TCP/IP channels (*qcctmxtp*)
- LU62CHL - maximum permitted LU62 channels (*qcctmxlu*)
-  31-bit storage used by CHINIT in the extended private region (*qcctstus*). This information is also provided by the `CSQX004I` message in the CHINIT job log.
-  64-bit storage limit available to the CHINIT (*qcctslim*)
-  64-bit storage used by CHINIT (*qcctstab*). This information is also provided by the `CSQX004I` message in the CHINIT job log.

The format of the channel initiator control information block is described in structure *qcct* in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQCCS)`.

You can use this information to see if the number of active channels is approaching the configured maximum value. Note that the number of current and active channels are the values when the record was created. So, between the two intervals there might have been more than this number of channels active.

Channel information from SMF data

Here is an example of channel information from SMF data:

```
 MV4A, MQ27, 2023/10/02, 11:53:02, VRM:934,  
From 2023/10/02, 11:52:52 to 2023/10/02, 11:53:02, duration 10 seconds.  
Peak number used of current channels..... 1  
Peak number used of active channels ..... 1  
MAXCHL. Max allowed current channels..... 9999  
ACTCHL. Max allowed active channels..... 9999  
TCPCHL. Max allowed TCP/IP channels..... 9999
```

```

LU62CHL. Max allowed LU62 channels..... 200
31-bit storage used..... 436 MB
64-bit storage limit.....16384 PB
64-bit storage used..... 187 MB
64-bit storage free.....16384 PB

```

You can monitor the storage usage and see whether the value is trending upwards. If the total used is approaching the total storage available, you might be running out of storage, and so might not be able to support many more channels.

If the numbers of active current channels are tending towards the maximum number of channels, you might need to increase the maximum number of channels.

Dispatcher tasks

This topic contains example data for the dispatcher tasks statistics, and information about how to interpret the data.

The format of the dispatcher task block is described in structure `qct_dsp` in the C programming language header file `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

Example data

Task	Type	Requests	Busy %	CPU used, Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	DISP	26587	0.4	0.592463	0.1	22	127
1	DISP	26963	0.3	0.588092	0.1	22	112
2	DISP	864329	2.7	2.545668	0.3	3	28
3	DISP	26875	0.4	0.590825	0.1	22	120
4	DISP	26874	0.4	0.603285	0.1	22	123
Summ	DISP	971628	0.8	4.920332	0.1	5	38

The example data shows that there were five dispatchers. A channel is associated with a dispatcher when it starts. The channel initiator tries to distribute work across all the dispatchers when allocating a channel to a dispatcher. This example shows that one dispatcher is processing more requests than other dispatchers. This is normal, as some channels might stop, so the dispatcher is processing fewer channels, and some channels can be busier than others.

- 4.9 seconds of CPU were used by the dispatchers.
- The average request used 5 microseconds of CPU and took 38 microseconds elapsed time.
- A dispatcher is used to send and receive data over a communications network, and this is not usually dependent on external events. The average elapsed time should, therefore, be close to the average CPU time used. The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests.

The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time. If the CHINIT is delayed due to lack of CPU, then the ratio of average elapsed time to average CPU time is much larger, compared to when the CHINIT is not delayed for CPU.

- The average CPU used per request depends on the message traffic. For example, bigger messages use more CPU than smaller messages.

The fields are calculated from:

- Duration: `qwhs.qwhsdurn`
- Requests : `qctreqn`
- Busy %: `qcteltm` and duration
- CPU used: `qctcptm`
- CPU %: `qctcptm` and duration
- Average CPU: `qctcptm` and `qctreqn`

- Average ET: *qcteltn* and *qctreqn*

Usually, the number of dispatchers should be less than, or equal to, the number of processors in the LPAR. If you have more dispatchers than processors in the LPAR they might compete for CPU resources. For more information about tuning your system, see [SupportPac MP16](#).

Channels have an affinity to a dispatcher, so you might find that some dispatchers process many more requests than another dispatcher.

You can use the ALTER QMGR CHIDISPS() command to change the number of dispatchers used. Any change comes into effect the next time the channel initiator is started.

Adapter tasks

This topic contains example data for the adapter tasks statistics, and information about how to interpret the data.

The format of the adapter task block is described in structure *qct_adp* in the C programming language header file *th1qua1.SCSQC370* (CSQDSMFC), and in assembler macro *th1qua1.SCSQMACS* (CSQDQCTA).

Example data

Task	Type	Requests	Busy %	CPU used, Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	ADAP	470297	10.2	41.290670	4.6	88	194
1	ADAP	13907	0.6	1.589428	0.2	114	365
2	ADAP	2517	0.2	0.185325	0.0	74	746
3	ADAP	1095	0.1	0.085774	0.0	78	907
4	ADAP	535	0.1	0.040743	0.0	76	947
5	ADAP	220	0.0	0.016228	0.0	74	1175
6	ADAP	82	0.0	0.005521	0.0	67	1786
7	ADAP	80	0.0	0.004248	0.0	53	1160
Summ	ADAP	488733	1.4	43.217938	0.6	88	205

The fields are calculated from:

- Duration: *qwhs.qwhsdurn*
- Requests: *qctreqn*
- Busy %: *qcteltn* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn* average
- ET: *qcteltn* and *qctreqn*

This example shows that there were eight adapter tasks.

Adapter number 0

- Processed the majority of the requests (470297 out of 488733)
- Was busy 10.2% of the interval
- Used 41.3 seconds of CPU

Overall

The average CPU per request was 88 microseconds of CPU and took 205 microseconds

The adapters process IBM MQ requests. Some of these requests might wait, for example, for log I/O during a commit, so the average Elapsed Time per request has little meaning.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qcteltn*) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time.

When an IBM MQ request is made the first free adapter task is used.

- If there is at least one adapter that has been little used (less than 1%) busy, you have enough adapters.
- If at least one adapter was not used, you have enough adapters defined.
- If all the adapters were used, you might need to allocate more adapters.
- If all of the adapters were used, and they were all busy for most of the interval, you need to allocate more adapters.

You can use the ALTER QMGR CHIADAPS() command to change the number of adapters used. Any changes come into effect the next time the channel initiator is started.



Attention: If there are too many adapters acting on a small set of queues, you might get contention within the queue manager.

Related reference

[ALTER QMGR](#)

Domain Name Server (DNS) task

This topic contains example data for the DNS tasks statistics, and information about how to interpret the data.

The format of the DNS task block is described in structure `qct_dns` in the C programming language header file `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

Example data

```
Task, Type, Requests, Busy %, CPU used, CPU %, "avg CPU", "avg ET", longest,
date, time
0, DNS, 14002, 0.0, 0.122578, 0.0, 9, 11, 463, 2014/03/18,
12:56:33.987671
Summ, DNS, 14002, 0.0, 0.122578, 0.0, 9, 11, 463, 2014/03/18,
12:56:33.987671
```

The channel initiator uses a single DNS task. The example shows that the task processed 14002 requests and on average the request used 9 microseconds of CPU and took 11 microseconds of elapsed time.

The longest DNS request took 463 microseconds elapsed time, and this occurred at 12:56:33 local time.

The fields are calculated from:

- Duration: *qwhs.qwhsdurn*
- Requests : *qctreqn*
- Busy %: *qcteltm* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn*
- Average ET: *qcteltm* and *qctreqn*
- Longest: *qctlgdu*
- Longest at: *qctlgtm*

The DNS task can go out of your enterprise to look up the IP address associated with a name. If the average Elapsed time is significantly more than the average CPU time used, you might have some long requests.

If the value of the longest request time is unacceptable you should work with your network team to investigate why you are having long requests. It might be that you have an invalid name in your connections.

If the DNS task is busy for 25% of the duration, consider investigating the cause further.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time might be greater than the elapsed time.

Note: There are requests to the DNS task that are not DNS lookups, so you might have the number of requests being greater than zero - but no longest request information.

SSL tasks

This topic contains example data for the SSL tasks statistics, and information about how to interpret the data.

The format of the SSL task block is described in structure `qct_ssl` in the C programming language header file `th1qua1.SCSQC370` (CSQDSMFC), and in assembler macro `th1qua1.SCSQMACS` (CSQDQCTA).

Example data

Task, date,	Type,	Requests, time	Busy %,	CPU used, Seconds,	CPU %,	"avg CPU", uSeconds,	"avg ET", uSeconds,	longest, uSeconds,	
0, 12:46:40.237697	SSL,	3112,	1.2,	0.248538,	0.3,	80,	362,	8864,	2014/03/18,
1, 12:46:18.938022	SSL,	3070,	1.2,	0.245433,	0.3,	80,	359,	4714,	2014/03/18,
2, 12:46:35.358145	SSL,	3170,	1.2,	0.255557,	0.3,	81,	362,	7273,	2014/03/18,
3, 12:46:44.514045	SSL,	3060,	1.2,	0.246542,	0.3,	81,	365,	13164,	2014/03/18,
4, 12:46:22.134123	SSL,	3120,	1.3,	0.251927,	0.3,	81,	373,	22438,	2014/03/18,
Summ, 12:46:22.134123	SSL,	15532,	1.2,	1.247998,	0.3,	80,	364,	22438,	2014/03/18,

This example data shows that the average request took 364 microseconds. The longest request was for SSL task 4, took 22,438 microseconds, and occurred at 12:46:22.134123 local time.

The fields are calculated from:

- Duration: *qwhs.qwhsdurn*
- Requests : *qctreqn*
- Busy %: *qcteltm* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn*
- Average ET: *qcteltm* and *qctreqn*
- Longest: *qctlstdu* longest at: *qctlstm*

A running channel is associated with an SSL task, in a similar way that a channel is associated with a dispatcher. The SSL tasks can use the cryptographic coprocessors available to the LPAR. So, the elapsed time can include time spent on a coprocessor. You should monitor the average elapsed time throughout the day. If this time increases significantly during peak periods you should work with your z/OS systems programmers, as your coprocessors might be over-used.

If the SSL tasks are busy for a significant proportion of the interval, increasing the number of SSL tasks might help. If the SSL tasks are waiting for external resources such as a coprocessor, increasing the number of SSL tasks has little effect.

You can use the `ALTER QMGR SSLTASKS()` command to change the number of SSL tasks used. Any changes come into effect the next time the channel initiator is started.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time might be greater than the elapsed time.

Related reference

[ALTER QMGR](#)

Queue data records

Use this topic as a reference for queue (SMF type 115, subtype 216) data records. The statistics are designed to make it easier for you to monitor usage and performance of your queue over time, and give an insight into what happened with your queue during the last SMF interval. This includes all the [DISPLAY QSTATUS](#) information and information on message flow, expiry, high and low watermarks and more.

The format of the queue statistics data record is described in assembler macro `th1qual.SCSQMACS(CSQDQST)`.

The queue statistics record contains information on the performance of selected queues and includes the following fields:

QQSTID – Control block identifier

The identifier for the queue statistics control block; is always x' D80F '.

QQSTLL – Length of control block

The length of a queue statistics record.

QQSTEYEC – Control block eyecatcher

The eyecatcher used to make identification of the control block easier; is always ' QQST '

QQSTQNAM – Queue name

The name of the queue.

QQSTFLAG

An array of bits containing the following information about the queue:

QQSTDISP – Queue disposition

This bit identifies whether the queue is of private or shared disposition. If the bit is on, then it is a shared queue.

QQSTPART – Partial record identifier

This bit identifies whether the record is a full or partial record. If the bit is on, then it is a partial record. When this flag is set there was an issue accessing the information on the queue, for example if there is a CF structure failure.

In a shared queue partial record, the accuracy of **qqstdpth**, **qqstmage**, and **qqstuncm** cannot be guaranteed. Therefore, the fields **qqstdpth** and **qqstmage** are populated with x' 00 ' and the **qqstuncm** flag is not set.

In a private queue partial record, the accuracy of **qqstmage** cannot be guaranteed, therefore the field is populated with x' 00 '.

QQSTUNCM – Uncommitted changes pending

This bit indicates whether there are any uncommitted changes (puts and gets) pending for the queue. If the bit is on, there are uncommitted changes.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would if run at the time the SMF record was generated.

If the queue is a shared queue and QQSTPART is set, this bit is always off as there might have been an issue obtaining the correct value.

QQSTPSID – Page set ID

The page set ID where the queue is located, if allocated and a private queue. If the queue is a shared queue, or a private queue that does not have a page set assigned to it, this field is set to -1 (x' FFFF ').

This value is correct at the time the SMF record was generated. It is possible that the page set changed during the SMF interval, in which case, the value reflected in the next SMF record will be the new page set.

QQSTBPID – Buffer pool ID

The buffer pool ID used by the queue, if allocated and a private queue. If the queue is a shared queue, or a private queue that does not have a buffer pool assigned to it, this field is set to -1 (x' FFFF ').

This value is correct at the time the SMF record was generated. It is possible that the buffer pool changed during the SMF interval. In this case, the value reflected in the SMF record is the new buffer pool.

QQSTQSGN – QSG name

The Queue Sharing Group name that the queue manager is a member of, if it is a shared queue. If the queue is a private queue this field is blank.

QQSTCFST – CF Structure name

The coupling facility (CF) structure name the queue uses if it is a shared queue. If the queue is a private queue this field is blank.

QQSTDPTH – Current depth

The depth of the queue at the time the SMF data was captured.

If the queue is a shared queue and QQSTPART is set, this value is always zero as there might have been an issue obtaining the correct value.

QQSTOPCT – Current open for output count

The number of handles that are currently open for output for the queue at the time when the SMF data was captured. For shared queues, the number returned applies only to the queue manager generating the record. The number is not the total for all the queue managers in the queue sharing group.

This is the same as OPPROCS from a DISPLAY QSTATUS command.

This is checked and set at the time of the SMF data collection and provides the same result as a DISPLAY QSTATUS command would, if run at the time the SMF record was generated.

QQSTIPCT – Current open for input count

The number of handles that are currently open for input for the queue at the time when the SMF data was captured. For shared queues, the number returned applies only to the queue manager generating the record. The number is not the total for all the queue managers in the queue sharing group.

This is the same as IPPROCS from a DISPLAY QSTATUS command.

This is checked and set at the time of the SMF data collection and provides the same result as a DISPLAY QSTATUS command would, if run at the time the SMF record was generated.

QQSTMAGE – Oldest message age

The age, in seconds, of the oldest message on the queue.

This is checked and set at the time of the SMF data collection and provides the same result as a DISPLAY QSTATUS command would, if run at the time the SMF record was generated.

If QQSTPART is set, this value is always zero as there might have been an issue obtaining the correct value.

QQSTQTST – Short term QTIME

The interval, in microseconds, between messages being put on the queue and then being destructively read. Value based on the last few messages processed. For shared queues, the values shown are for measurements collected on this queue manager only.

This is the same as the first value in QTIME from a DISPLAY QSTATUS command.

This is checked and set at the time of the SMF data collection and provides the same result as a DISPLAY QSTATUS command would, if run at the time the SMF record was generated.

QQSTQTLT – Long term QTIME

The interval, in microseconds, between messages being put on the queue and then being destructively read. The value is based on a larger sample of the recently processed messages. For shared queues, the values shown are for measurements collected on this queue manager only.

This is the same as the second value in QTIME from a DISPLAY QSTATUS command.

This is checked and set at the time of the SMF data collection and provides the same result as a DISPLAY QSTATUS command would, if run at the time the SMF record was generated.

QQSTLPUT – Last put date/time

The time, in store clock format, at which the last message was put to the queue since the queue manager started. For shared queues, the value shown is for messages put by this queue manager only.

This is the same as LPUTDATE and LPUTTIME from a DISPLAY QSTATUS command.

This is checked and set at the time of the SMF data collection and provides the same result as a DISPLAY QSTATUS command would, if run at the time the SMF record was generated.

QQSTLGET – Last get date/time

The time, in store clock format, at which the last message was retrieved from the queue since the queue manager started. For shared queues, the value shown is for messages put by this queue manager only.

A message being browsed does not count as a message being retrieved.

This is the same as LGETDATE and LGETTIME from a DISPLAY QSTATUS command.

This is checked and set at the time of the SMF data collection and provides the same result as a DISPLAY QSTATUS command would, if run at the time the SMF record was generated.

QQSTDPHI – Highest depth

The highest depth reached by the queue during the SMF interval.

For shared queues, queue managers only have partial information about the change in depth of the queue over time. The QQSTDPHI value is based off this partial information as follows:

- At the start of the interval the value of QQSTDPHI is set to zero.
- When an application puts a message to the queue in the interval the queue manager checks the depth of the queue, including the message just being put. If this value is higher than the current value of QQSTDPHI, then it is used as the new value of QQSTDPHI.
- When SMF data for the queue is collected, the queue manager will check if the current queue depth is higher than QQSTDPHI, if so the current queue depth is used as the new value of QQSTDPHI.

This approach means that the value of QQSTDPHI does not take into account messages put by other queue managers in the queue sharing group, unless those messages contributed to the queue depth at the point where SMF data is collected.

QQSTDPLO – Lowest depth

The lowest depth reached by the queue during the SMF interval.

For shared queues, queue managers only have partial information about the change in depth of the queue over time. The QQSTDPLO value is based off this partial information as follows:

- At the start of the interval the value of QQSTDPLO is set to a special value.
- The first time during the interval the queue manager obtains the depth of the queue, QQSTDPLO to that value.
- When an application puts a message to the queue in the interval the queue manager checks the depth of the queue, including the message just being put. If this value is lower than the current value of QQSTDPLO, then it is used as the new value of QQSTDPLO.
- When SMF data for the queue is collected, the queue manager will check if the current queue depth is lower than QQSTDPLO, if so the current queue depth is used as the new value of QQSTDPLO.

This approach means that the value of QQSTDPLO does not take into account messages got by other queue managers in the queue sharing group, unless those messages contributed to the queue depth at the point where SMF data is collected.

QQSTPUTS – MQPUT count

The number of messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPUT1 – MQPUT1 count

The number of messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNPPT – Non-persistent MQPUT count

The number of non-persistent messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPPT – Persistent MQPUT count

The number of persistent messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNPP1 – Non-persistent MQPUT1 count

The number of non-persistent messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPP1 – Persistent MQPUT1 count

The number of persistent messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPUTB – MQPUT bytes

The number of bytes of message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPT1B – MQPUT1 bytes

The number of bytes of message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNPPB – Non-persistent MQPUT bytes

The number of bytes of non-persistent message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTPPB – Persistent MQPUT bytes

The number of bytes of persistent message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTNP1B – Non-persistent MQPUT1 bytes

The number of bytes of non-persistent message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTP1B – Persistent MQPUT1 bytes

The number of bytes of persistent message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

QQSTFLPT – Failed MQPUT count

The number of MQPUT calls targeting the queue, which failed with a completion code of MQCC_FAILED, during the SMF interval.

For shared queues, the count only includes failed puts attempted through the queue manager that generated the SMF record.

QQSTFLP1 – Failed MQPUT1 count

The number of MQPUT1 calls targeting the queue, which failed with a completion code of MQCC_FAILED, during the SMF interval.

For shared queues, the count only includes failed puts attempted through the queue manager that generated the SMF record.

QQSTFPTC – Fast puts to a waiting getter count

The number of MQPUT and MQPUT1 calls targeting the queue, that were fast put to a waiting getter during the SMF interval.

Note: *Put to a waiting getter* is a technique whereby a message might not actually be put onto a queue if there is an application already waiting to get the message. Certain conditions must be satisfied for this to occur, in particular the message must be non-persistent and the putting and getting application must be processing the message outside syncpoint control.

If these conditions are met, then the message is transferred from the putting application's buffer into the getting application's buffer without actually touching the IBM MQ queue. This removes a lot of processing involved in putting the message on the queue and therefore leads to increased throughput and lower CPU costs.

QQSTFPTB – Fast puts to a waiting getter bytes

The number of message and properties bytes from MQPUT and MQPUT1 calls targeting the queue, that were fast put to a waiting getter during the SMF interval. This does not include message headers in the calculation of the size.

QQSTSTRM – Streamed message count

The number of messages that were successfully streamed from the queue during the interval. This is always zero if STREAMQ is not set for the queue.

QQSTMSMI – Minimum message size put

The minimum message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

QQSTMSMA – Maximum message size put

The maximum message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

QQSTMSAV – Average message size put

The average message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

QQSTGETS – Destructive MQGET count

The number of messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTNPDG – Non-persistent destructive MQGET count

The number of non-persistent messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTPDG – Persistent destructive MQGET count

The number of persistent messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTGETB – Destructive MQGET byte count

The number of message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTNPDB – Non-persistent destructive MQGET bytes

The number of non-persistent message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTPDB – Persistent destructive MQGET bytes

The number of persistent message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

QQSTBRWS – Non-destructive MQGET count

The number of messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTNPBR – Non-persistent non-destructive MQGET count

The number of non-persistent messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTPBR – Persistent non-destructive MQGET count

The number of persistent messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTBRWB – Non-destructive MQGET bytes

The number of message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTNPBB – Non-persistent non-destructive MQGET bytes

The number of non-persistent message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTPBB – Persistent non-destructive MQGET bytes

The number of persistent message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

QQSTFLGT – Failed destructive MQGET count

The number of destructive MQGET calls targeting the queue, that failed with a completion code of MQCC_FAILED, during the SMF interval.

Also included in the count is MQCC_WARNING when accompanied by a return code of MQRC_TRUNCATED_MSG_FAILED. However, not included in this count is any MQGET with a wait that receives MQRC_NO_MSG_AVAILABLE.

For shared queues, the count only includes failed MQGET attempts through the queue manager that generated the SMF record.

QQSTNMAG – Failed destructive MQGET with MQRC_NO_MSG_AVAILABLE count

The number of destructive MQGET calls, without wait, targeting the queue, that failed with both a completion code of MQCC_FAILED and a return code of MQRC_NO_MSG_AVAILABLE, during the SMF interval.

This value is a subset of QQSTFLGT.

For shared queues, the count only includes failed MQGET, without wait, attempts through the queue manager that generated the SMF record.

QQSTTMFB – Failed destructive MQGET with MQRC_TRUNCATED_MSG_FAILED count

The number of destructive MQGET calls targeting the queue, that failed with both a completion code of MQCC_WARNING and a return code of MQRC_TRUNCATED_MSG_FAILED, during the SMF interval.

This value is a subset of QQSTFLGT.

For shared queues, the count only includes failed MQGET attempts through the queue manager that generated the SMF record.

QQSTFLGW – No message available for destructive MQGET with a wait count

The number of times that there is no message available for destructive MQGET calls with a wait, targeting the queue, causing the MQGET to continue waiting, during the SMF interval.

QQSTRDGW – Re-driven destructive MQGET with a wait count

The number of times that destructive MQGET calls with a wait are re-driven to check if there is a message on the queue that matches their criteria, during the SMF interval.

When a new message arrives on the queue, all eligible waiting MQGET calls are woken up to attempt to get the message. Each MQGET with a wait that is woken up to check for a valid message increments this count by one. If any of these MQGET with wait fails to get the message, QQSTFLGW is incremented by one and the MQGET goes back into a waiting state.

QQSTFLBW – No message available for non-destructive MQGET with a wait count

The number of times that there is no message available for non-destructive MQGET calls with a wait, targeting the queue, causing the MQGET to continue waiting, during the SMF interval.

QQSTRDBW – Re-driven non-destructive MQGET with a wait count

The number of times that non-destructive MQGET calls with a wait are re-driven to check if there is a message on the queue that matches their criteria, during the SMF interval.

When a new message arrives on the queue, all eligible waiting MQGET calls are woken up to attempt to browse the message. Each MQGET with a wait that is woken up to check for a valid message increments this count by one. If any of these MQGET with wait fails to browse the message, QQSTFLBW is incremented by one and the MQGET goes back into a waiting state.

QQSTSAGT – Destructive MQGET with MQRC_SIGNAL_REQUEST_ACCEPTED count

The number of destructive MQGET calls targeting the queue, that complete with both a completion code of MQCC_WARNING and a return code of MQRC_SIGNAL_REQUEST_ACCEPTED, during the SMF interval.

QQSTSABR – Non-destructive MQGET with MQRC_SIGNAL_REQUEST_ACCEPTED count

The number of non-destructive MQGET calls targeting the queue, that complete with both a completion code of MQCC_WARNING and a return code of MQRC_SIGNAL_REQUEST_ACCEPTED, during the SMF interval.

QQSTIPHI – High watermark for IPPROC

The highest number of concurrent input handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTIPL0 – Low watermark for IPPROC

The lowest number of concurrent input handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTOPHI – High watermark for OPPROC

The highest number of concurrent output handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTOPLO – Low watermark for OPPROC

The lowest number of concurrent output handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

QQSTOPEN – Successful MQOPEN count

The number of times the queue was successfully opened during the SMF interval. This does not include opens performed as part of an MQPUT1 call.

For shared queues, the count only includes when the queue is opened through the queue manager that generated the SMF record.

QQSTCLOS – MQCLOSE count

The number of times the queue was successfully closed using MQCLOSE, during the SMF interval.

For shared queues, the count only includes when the queue is closed through the queue manager that generated the SMF record.

QQSTINQR – MQINQ count

The number of MQINQ calls that completed with a completion code of MQCC_OK or MQCC_WARNING, during the SMF interval.

QQSTSET – MQSET count

The number of MQSET calls that completed with a completion code of MQCC_OK during the SMF interval.

QQSTEXPR – Expired messages count

The number of expired messages cleared from the queue during the SMF interval.

This includes messages expired by an application issuing an MQGET, by the expired message scanning task or by a REFRESH QMGR TYPE(EXPIRY) command.

QQSTRBPT – Rolled back MQPUT counts

The number of messages that were put to the queue, that have been rolled back off the queue, during the SMF interval.

QQSTRBGT – Rolled back MQGET counts

The number of messages destructively read from the queue, that have been rolled back onto the queue, during the SMF interval.

Interpreting IBM MQ for z/OS accounting data

IBM MQ for z/OS accounting data is written as SMF type 116 records. Use this topic as a reference to the different types of accounting data records.

IBM MQ accounting information can be collected for the following subtypes:

0

Message manager accounting records (how much processor time was spent processing IBM MQ API calls and the number of MQPUT and MQGET calls). This information is produced when a named task disconnects from IBM MQ, and so the information contained within the record might cover many hours.

1

Accounting data for each task, at thread and queue level.

2

Additional queue-level accounting data (if the task used more queues than could fit in the subtype 1 record).

10

Accounting data for channels.

Note: Accounting information for specific channels can be enabled or suppressed by the **STATCHL** channel attribute, and the **STATACLS** queue manager attribute.

Note that:

- Subtype 0 records are produced with accounting trace class 1.
- Subtype 1 and 2 records are produced with accounting trace class 3.
- Subtype 10 records are produced with accounting trace class 4.

Layout of an SMF type 116 record

Use this topic as a reference to the format of an SMF type record.

The standard layout for SMF records involves three parts:

SMF header

Provides format, identification, and time and date information about the record itself.

Self-defining section

Defines the location and size of the individual data records within the SMF record.

Data records

The actual data from IBM MQ that you want to analyze.

For more information about SMF record formats, see [z/OS MVS System Management Facilities \(SMF\)](#).

The SMF header

Table 42 on page 377 shows the format of SMF record header (SM116).

Off-set: Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM116	SMF record header.	
0	0	Integer	2	SM116LEN	SMF record length.	01A4
2	2		2		Reserved.	
4	4	Integer	1	SM116FLG	System indicator.	5E
5	5	Integer	1	SM116RTY	Record type. The SMF record type, for IBM MQ accounting records this is always 116 (X'74').	74
6	6	Integer	4	SM116TME	Time when SMF moved record.	00356124
10	A	Integer	4	SM116DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM116SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)
18	12	Character	4	SM116SSI	IBM MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM116STF	Record subtype.	0000
24	18	Character	3	SM116REL	IBM MQ version.	F9F3F0 (930)
27	1B		1		Reserved.	
28	1C	Character	0	SM116END	End of SMF header and start of self-defining section.	

Note: The (hexadecimal) values in the right-hand column relate to [Figure 22 on page 379](#).

Self-defining sections

A self-defining section of an SMF record tells you where to find an accounting record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow the header, at a fixed offset from the start of the SMF record.

Each self-defining section points to accounting related data. [Table 43 on page 378](#) summarizes the offsets from the start of the SMF record header.

Table 43. Offsets to self-defining sections

Record subtype (SMF116STF)	Source of accounting data	Offset of self-defining section		See...
		Dec	Hex	
All	Common header	28	X'1C'	“Common IBM MQ SMF header” on page 380
0	Message manager	44	X'2C'	“Message manager data records” on page 382
1	Thread identification record	36	X'24'	“Thread-level and queue-level data records” on page 383
1	Thread-level accounting	44	X'2C'	“Thread-level and queue-level data records” on page 383
1	Queue-level accounting	52	X'34'	“Thread-level and queue-level data records” on page 383. This section is present only if the WTASWQCT field in the task-related information (WTAS) structure is non-zero.
2	Thread identification record	36	X'24'	“Thread-level and queue-level data records” on page 383
2	Queue-level accounting	44	X'2C'	“Thread-level and queue-level data records” on page 383
10	Channel accounting			“Channel accounting data records” on page 386

Note: Other self-defining sections refer to data for IBM use only.

Each self-defining section is two fullwords long and has this format:

```
sssssssl111nnnn
```

where:

SSSSSSSS

Fullword containing the offset from start of the SMF record.

LLLL

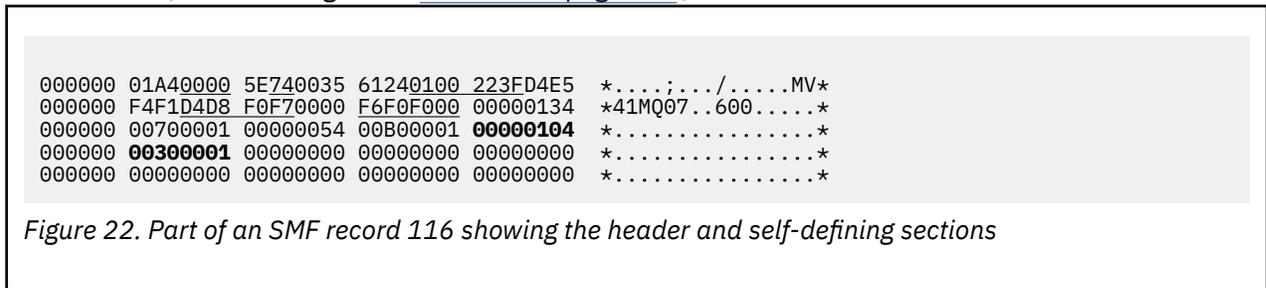
Halfword giving the length of this data record.

NNNN

Halfword giving the number of data records in this SMF record.

Figure 22 on page 379 shows an example of part of an SMF type 116 record. The numbers in the left-hand column represent the offset, in hexadecimal, from the start of the record. Each line corresponds to sixteen bytes of data, where each byte is two hexadecimal characters, for example 0C. The characters in the right-hand column represent the printable characters for each byte. Non-printable characters are shown by a period (.) character.

In this example, alternate fields in the SMF header are underlined to help you to see them; refer to Table 42 on page 377 to identify them. The self defining section for one of the message manager accounting data records (at the offset given in Table 43 on page 378) is shown in **bold**.



The self-defining section for the type of message manager accounting data is located at offset X'2C' from the start of the SMF record and contains this information:

- The offset of the message manager accounting data is located X'00000104' bytes from the start of the SMF record.
- This message manager record is X'0030' bytes long.
- There is one record (X'0001').

Note: Always use offsets in the self-defining sections to locate the accounting records.

z/OS Processing type 116 SMF records

Use this topic as a reference to the format of the processing type accounting record.

Any accounting data you collect from SMF must be processed to extract useful information. When you process the data, verify that the records are from IBM MQ and that they are the records you are expecting.

Validate the value of the following fields:

- SM116RTY, the SMF record number = X'74' (116)
- SM116STF, the record subtype, must be 0000, 0001, 0002, or 0010

Reading from the active SMF data sets (or SMF logstreams) is not supported. You must use the SMF program IFASMFDP (or IFASMF DL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed. For more information see [“Using System Management Facility” on page 335](#).

Details of the structures and fields can be found in IBM MQ SupportPac [MP1B](#).

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records from the sequential data set. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC(CSQ4SMFJ).

You need to update the SMFIN DD card with the name of the SMF data set. Use the z/OS command '/D SMF' to show the name of the data set, and you need to update the DUMPOUT DD card with the name for the output data set.

You also need to specify the START and END times that you require.

The following sample JCL extracts SMF records from SMF data sets and dumps them to the SMFOUT data set:

```
//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=
//SMFIN DD DSN=xxxxxx.MANA,DISP=SHR
//SMFOUT DD DSN=xxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
INDD(SMFIN,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

The following sample JCL extracts SMF records from the SMF log stream named in LSNAME and dumps them to the SMFOUT data set:

```
//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=*
//SMFOUT DD DSN=xxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
LSNAME(IFASMF.MQ,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

Common IBM MQ SMF header

Use this topic as a reference to the common IBM MQ SMF header type accounting record.

The format of this record is described in Table 44 on page 380 and in assembler macros thlqual.SCSQMACS(CSQDQWHS) and thlqual.SCSQMACS(CSQDQWHC), and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qwhs, qwhsnsda.

The QWHS data includes the subsystem name. For subtype 1 records, it also shows whether there are queue-level accounting records present. If the QWHSNSDA field is 3 or less, there are not, and the corresponding self-defining section (at offset X'34') is not set.

The QWHC data gives you information about the user (for example, the user ID (QWHCAID) and the type of application (QWHCATYP)). The QWHC section is completed only for subtype 0 records. The equivalent information is present in the thread identification record for subtype 1 and 2 records.

<i>Table 44. Structure of the common IBM MQ SMF header record QWHS</i>					
Offset: Dec	Off- set: Hex	Type	Length	Name	Description
0	0	Structure	128	QWHS	
0	0		6		Reserved
6	6	Character	1	QWHSNSDA	Number of self defining sections in the SMF records
7	7		5		Reserved
12	C	Character	4	QWHSSID	Subsystem name
16	10		24		Reserved
40	28	Character	8	QWHCAID	User ID associated with the z/OS job
48	30	Character	12	QWHCCV	Thread cross-reference
60	3C	Character	8	QWHCCN	Connection name

Table 44. Structure of the common IBM MQ SMF header record QWHS (continued)

Offset: Dec	Offset: Hex	Type	Length	Name	Description
68	44		8		Reserved
76	4C	Character	8	QWHCOPID	User ID associated with the transaction
84	54	Integer	4	QWHCATYP	Type of connecting system (1=CICS, 2=Batch or TSO, 3=IMS control region, 4=IMS MPP or BMP, 5=Command server, 6=Channel initiator, 7=RRS Batch)
88	58	Character	22	QWHCTOKN	Accounting token set to the z/OS accounting information for the user
110	6E	Character	16	QWHCNID	Network identifier
126	7E		2		Reserved

Combining CICS and IBM MQ performance data

Use this topic as a reference to the combination of IBM MQ and CICS performance data.

The common IBM MQ SMF header type accounting record section, QWHCTOKN, is used to correlate CICS type 110 SMF records with IBM MQ type 116 SMF records.

CICS generates an LU6.2 unit-of-work token, for each CICS task. The token is used to generate an accounting token that is written to QWHCTOKN in the correlation header of subtype zero records.

Details are also written to the WTIDACCT section in subtype 1 and 2 records. The accounting token enables correlation between CICS and IBM MQ performance data for a transaction.

Thread cross-reference data

Use this topic as a reference to the format of the thread cross-reference type accounting record.

The interpretation of the data in the thread cross-reference (QWHCCV) field varies. This depends on what the data relates to:

- CICS connections (QWHCATYP=1) - see [Table 45 on page 381](#)
- IMS connections (QWHCATYP=3 or 4) - see [Table 46 on page 382](#)
- Batch connections (QWHCATYP=2 or 7) - this field consists of binary zeros
- Others - no meaningful data

Table 45. Structure of the thread cross-reference for a CICS system

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	CICS thread number.
52	34	Character	4	CICS transaction name.
56	38	Integer	4	CICS task number.

Some entries contain blank characters. These apply to the task, rather than to a specific transaction.

Table 46. Structure of the thread cross-reference for an IMS system

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	IMS partition specification table (PST) region identifier.
52	34	Character	8	IMS program specification block (PSB) name.

Message manager data records

Use this topic as a reference to the format of the message manager accounting records.

The message manager is the component of IBM MQ that processes all API requests. The format of the message manager accounting records is described in assembler macro `thlqual.SCSQMACS(CSQDQMAC)`.

The QMAC data gives you information about the processor time spent processing IBM MQ calls, and counts of the number of MQPUT and MQGET requests for messages of different sizes.

Note: A single IMS application might write two SMF records. In this case, add the figures from both records to provide the correct totals for the IMS application.

Records containing zero processor time

Records are sometimes produced that contain zero processor time in the QMACCPUT field. These records occur when long running tasks identified to IBM MQ either terminate or are prompted to output accounting records by accounting trace being stopped. Such tasks exist in the CICS adapter and in the channel initiator (for distributed queuing). The number of these tasks with zero processor time depends upon how much activity there has been in the system:

- For the CICS adapter, this can result in up to nine records with zero processor time.
- For the channel initiator, the number of records with zero processor time can be up to the sum of `Adapters + Dispatchers + 6`, as defined in the queue manager attributes.

These records reflect the amount of work done under the task, and can be ignored.

Sample subtype zero accounting record

Use this topic as a reference to the format of the subtype zero accounting records.

Figure 23 on page 383 shows a type 116, subtype zero SMF record. In this figure, the SMF record header and the QMAC accounting data record are underlined. The self-defining sections are in bold.

```

000000 01A40000 5E740035 61240100 223FD4E5 *...;.../.....MV*
000010 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600....*
000020 00700001 00000054 00B00001 00000104 *.....*
000030 00300001 00000000 00000000 00000000 *.....*
000040 00000000 00000000 00000000 00000000 *.....*
000050 00000000 B478AB43 9C6C2280 B478AB47 *.....%.*
000060 9DB47E02 00000000 04C0F631 00000001 *..=.....}6....*
000070 9880E72D 00000000 014D9540 00000000 *..X.....(. ....*
000080 08480C80 00000010 40404040 40404040 *.....*
000090 00000000 00000000 00000051 00000000 *.....*
0000A0 00000000 00000000 00000000 00000000 *.....*
0000B0 00000000 00000000 00000000 00000000 *.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 00000000 00000000 00000000 *.....*
0000E0 00000000 00000000 00000000 00000000 *.....*
0000F0 00000000 00000000 00000000 00000000 *.....*
000100 00000000 D4140030 D8D4C1C3 00000000 *...M...QMAC...*
000110 689C738D 00000050 00000000 00000050 *.....&.....&*
000120 0000000A 00000000 00000000 00000000 *.....*
000130 00000000 0024011A 00030710 02DAACF0 *.....0*

```

Figure 23. Example SMF type 116, subtype zero record

Thread-level and queue-level data records

Use this topic as a reference to the format of the thread-level and queue-level accounting records.

Thread level accounting records are collected for each task using IBM MQ. In addition, queue-level accounting records are gathered about each queue that the task opens. A queue-level accounting record is written for each queue that the task has used since the thread-level accounting record was last written.

If the task uses a queue that is configured with a streaming queue, there is no queue-level accounting record for the streaming queue. Instead, the accounting record for the original queue accumulates data for the data points that would have been associated with the streaming queue.

The only exception to this is that the PUTN/PUT1N value shows the number of MQPUT/MQPUT1 requests made by the application, and excludes the extra MQPUT requests made to the streaming queue.

So, for example, if an application issues a single MQPUT request, the:

- PUTN value is 1
- Elapsed time (PUTET) and CPU time (PUTCT) for the MQPUT include the time taken to put to both the primary and streaming queue
- Number of page set requests (PUTPSN) includes those for both the primary and secondary queue, and so on

For each task, data is written to SMF when the task finishes.

From IBM MQ 9.3.0 onwards, for long running tasks, data is also written at the interval specified by either the ACCTIME, or STATIME, parameter of the CSQ6SYSP system parameter macro, or by the system SMF statistics broadcast, provided that the task was running the previous time data was gathered.

Thread-level and queue-level accounting records are produced if you specify class 3 when you start the accounting trace. For example, use the following command:

```
START TRACE(ACCTG) DEST(SMF) CLASS(3)
```

The thread level accounting information is written to an SMF type 116, subtype 1 record, and is followed by queue-level records. If the task opened many queues, further queue information is written to one or more SMF type 116 subtype 2 records. A thread identification control block is included in each subtype 1 and 2 record to enable you to relate each record to the correct task. Typically, the maximum number of queue-level records in each SMF record is about 45.

The format of the thread-level accounting record is described in assembler macro `thlqual.SCSQMACS(CSQDWTAS)`. The format of the queue-level accounting record is described in assembler macro `thlqual.SCSQMACS(CSQDWQ)`. The format of the thread identification record is described in assembler macro `thlqual.SCSQMACS(CSQDWTID)`. All these records are also described in C header file `thlqual.SCSQC370(CSQDSMFC)`. The field names in C are all in lowercase, for example `wtas`, `wtassex`.

z/OS *Meaning of the channel names*

Use this topic as a reference to the meaning of channel names.

The channel name in the WTID is constructed as shown in the following example. In this example a sender channel exists from queue manager QM1 to queue manager QM2.

The meaning of channel names are described in the following table.

Table 47. Meaning of channel names		
Field name	Meaning	Example
For queue manager QM1 the sender channel has the following fields set:		
WTIDCCN	The job name	QM1CHIN
WTIDCHL	The channel name	QM1.QM2
WTIDCHLC	This is defined in the CONNAME of the channel	WINMVS2B(2162)
For queue manager QM2 the receiver channel has the following fields set:		
WTIDCCN	The job name	QM2CHIN
WTIDCHL	The channel name	QM1.QM2
WTIDCHLC	Where the channel came from	9.20.101.14

z/OS *Sample subtype 1 and subtype 2 records*

Use this topic as a reference to the format of the subtype 1 and subtype 2 accounting records.

Figure 24 on page 384 and Figure 25 on page 385 show examples of SMF type 116, subtype 1 and subtype 2 records. These two accounting records were created for a batch job that opened 80 queues. Because many queues were opened, a subtype 2 record was required to contain all the information produced.

```

000000 703C0000 5E74002D 983B0100 229FD4E5 *...;.....MV*
000010 F4F1D4D8 F0F70001 F6F0F000 00006FCC *41MQ07..600...?.*
000020 00700001 0000003C 00D00001 0000010C *.....}*
000030 02C00001 000003CC 02400030 F70000D0 *.{.....}..7..}*
000040 E6E3C9C4 00000000 00000000 00000040 *WTID..... *
.
.
000100 00000000 00000000 7F4A4BB8 F70102C0 *....."...7..}*
000110 E6E3C1E2 B4802373 0BF07885 7F4AE718 *WTAS....0..".X.*

```

Figure 24. Example SMF type 116, subtype 1 record

The first self-defining section starts at X'24' and is **bold** in the example; X'0000003C' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic*; X'0000010C' is the offset to the WTAS data record, X'02C0' is the length of the WTAS record, and X'0001' is the number of WTAS records.

The third self-defining section starts at X'34' and is **bold** in the example; X'000003CC' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0030' is the number of WQST records.

Figure 25 on page 385 shows an example of an SMF type 116, subtype 2 record.

```

000000 49740000 5E74002D 983B0100 229FD4E5 *....;.....MV*
000010 F4F1D4D8 F0F70002 F6F0F000 00004904 *41MQ07..600....*
000020 00700001 00000034 00D00001 00000104 *.....}.....*
000030 02400020 F70000D0 E6E3C9C4 00000002 *. .7..}WTID....*
.
.
000100 7F4A4BB8 F7020240 E6D8E2E3 00000001 *"...7.. WQST....*

```

Figure 25. Example SMF type 116, subtype 2 record

The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic* ; X'00000104' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0020' is the number of WQST records.

Figure 26 on page 385 shows an example of an SMF type 116, subtype 1 record where no queues have been opened and there are consequently no self-defining sections for WQST records..

```

000000          5E740039 4E9B0104 344FD4E5 * .....|MV*
000010 F4F1D4D8 F0F70001 F6F0F000 000003DC *41MQ07..600....*
000020 00800001 00000034 00D00001 00000104 *.....}.....*
000030 02D80001 F70000D0 E6E3C9C4 00000002 *.Q..7..WTID....*
000040 C1F8C5C1 C4C5D740 C1F8C5C1 C4C54040 *A8EADEP A8EADE *
000050 40404040 40404040 00000000 00000000 * .....*
000060 40404040 40404040 4040          *          *

```

Figure 26. Example SMF type 116, subtype 1 record with no WQST data records

The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic* ; X'0000010C' is the offset to the WTAS data record, X'02D8' is the length of the WTAS record, and X'0001' is the number of WTAS records.

There is no self-defining section describing a WQST data record, equivalent to the third self-defining section in Figure 24 on page 384.

 **Layout of channel initiator SMF type 116 records**

The layout of channel accounting data (SMF type 116, subtype 10) records is described in this topic.

Self-defining section

The self-defining section for the channel accounting data follows the standard SMF header. It is structured in the standard triplet format. The format of the triplets is described in structure qws5 in the C programming language header file thlqual.SCSQC370 (CSQDSMFC), and in assembler macro thlqual.SCSQMACS (CSQDQWS5).

Table 48 on page 386 shows the format of the self-defining section.

Table 48. Structure of the channel accounting self-defining section

Offset: Dec	Off- set: Hex	Type	Length	Name	Description
0	0	Integer	4	QWS50PSO	Offset from the start of the SMF record to the first instrumentation standard header (QWHS)
4	4	Integer	2	QWS50PSL	Length of the QWHS
6	6	Integer	2	QWS50PSN	Number of instances of QWHS
8	8	Integer	4	QWS50R1O	Offset from the start of the SMF record to the first channel accounting data record (QCST)
12	C	Integer	2	QWS50R1L	Length of the QCST
14	E	Integer	2	QWS50R1N	Number of instances of QCST

Instrumentation standard header (QWHS)

The format of the QWHS is described in structure `qwhs` in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQWHS)`. It contains the following key fields that are relevant to channel initiator SMF 116 records:

Table 49. Key fields in the QWHS

Name	Length	Description
QWHSNDA	1 byte	Number of self-defining sections
QWHSSSID	4 bytes	Subsystem name
QWHS SMFC	1 bit	Indicates whether there are multiple SMF records containing information for this interval. If this bit is on, information for this interval is continued in further SMF records. If this bit is off, this is the last or only record.
QWHSTIME	8 bytes	Local time of the start of the interval in STCK format
QWHS DURN	8 bytes	Duration from the start of the interval to the end of the interval in STCK format
QWHSSTCK	8 bytes	End of the interval in UTC in STCK format

Channel accounting data records

Use this topic as a reference for channel accounting data records.

The format of the channel accounting data record is described in assembler macro `thlqual.SCSQMACS(CSQDQCST)`. The format is also described in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`. Note that the field names in C are all in lowercase, for example, `qcst`.

The channel accounting data gives you information about the status and statistics of each channel instance, including:

- Average network time (`qcstntav`)
- Average time on exit (`qcstetav`)

- Channel batch data limit (*qcstcbd*)
- Channel batch interval (*qcstcbit*)
- Channel batch size (*qcstcbsz*)
- Channel dispatcher number (*qcstdspn*)
- Channel disposition (*qcstchdp*)
- Channel name (*qcstchnm*)
- Channel state (*qcstchst*)
- Channel started time (*qcststrt*)
- Channel status collected time (*qcstcltm*)
- Channel stopped time (*qcstludt*)
- Channel type (*qcstchty*)
- Common name (CN) from SSLCERTI (*qcstslcn*)
- Compression rate (*qcstcpra*)
- Connection name (*qcstcnm*)
- Current shared conversations (*qcstcscv*)
- DNS resolution time (*qcstdnrt*)
- Effective value of STATCHL parameter (*qcststcl*)
- Last message time (*qcstlmst*)
- Maximum network time (*qcstntmx*)
- Maximum time on exit (*qcstetmx*)
- Minimum network time (*qcstntmn*)
- Minimum time on exit (*qcstetmn*)
- Name of the remote queue manager or application (*qcstrqmn*)
- Number of batches (*qcstbatc*)
- Number of bytes for message data (*qcstnbyt*)
- Number of bytes for persistent message data (*qcstnpby*)
- Number of bytes received for both message data and control information (*qcstbyrc*)
- Number of bytes sent for both message data and control information (*qcstbyst*)
- Number of full batches (*qcstfuba*)
- Number of messages, or number of MQI calls (*qcstnmsg*)
- Number of persistent messages (*qcstnmsg*)
- Number of put retries (*qcstptry*)
- Number of transmission queue becoming empty (*qcstqetc*)
- Number of transmission buffers received (**qcstbfrc**)
- Number of transmission buffers sent (*qcstbfst*)
- Serial number from SSLPEER (*qcstslsn*)
- SSL CipherSpec (zero means TLS not used) (*qcstslcs*)
- The date and time of maximum network time (*qcstntdt*)
- The date and time of maximum time on exit (*qcstetdt*)

Note, that for the channel accounting field *qcstetmn* (Minimum time on exit) and *qcstntmn* (Minimum network time) these two fields will be initialized to the hexadecimal value of 8FFFFFFF when unused.

You can use this information to see the throughput of a channel, if the actual batches are approaching the limit, the latency of the network, information about the remote end, performance of user exit, and so on.

Here is an example of the channel accounting data which has been formatted with IBM MQ SupportPac MP1B.

The fields available are based on the display channel status command (DIS CHS) and channel statistics by IBM MQ on platforms except z/OS, with some additional fields.

The data and time of the start and end of the record in local time, and the duration

```
SMF interval start      2014/03/26,02:30:00
SMF interval end       2014/03/26,02:45:00
SMF interval duration   899.997759 seconds
```

Information about the channel

```
Connection name      9.20.4.159
Channel disp        PRIVATE
Channel type        RECEIVER
Channel status      CLOSING
Channel STATCHL     HIGH
```

```
Start date & time      2014/03/26,02:44:58
Channel status collect time 2014/03/26,02:45:00
Last status changed   1900/01/01,00:00:00
Last msg time        2014/03/26,02:44:59
```

```
Batch size            50
Messages/batch        3.3
Number of messages    1,102
Number of persistent messages 1,102
Number of batches     335
Number of full batches 0
Number of partial batches 335
Buffers sent          337
Buffers received      1,272
Message data          5,038,344  4 MB
Persistent message data 5,038,344  4 MB
Non persistent message data 0 0 B
Total bytes sent      9,852  9 KB
Total bytes received  5,043,520  4 MB
Bytes received/Batch  15,055  14 KB
Bytes sent/Batch      29 29 B
Batches/Second        1
Bytes received/message 4,576  4 KB
Bytes sent/message    8 8 B
Bytes received/second 28,019 27 KB/sec
Bytes sent/second     54 54 B/sec
Compression rate      0
```

The name of the queue manager at the remote end of the connection

```
Remote qmgr/app      MQPH
Put retry count      0
```

IBM MQ-Netz optimieren

Verwenden Sie die Optimierungstipps in diesem Abschnitt, um die Leistung Ihres Warteschlangenmanagements zu verbessern.

Tuning Client-und Serververbindungskanäle

Die Standardeinstellung für **SHARECNV** ist 10. Dies ermöglicht bis zu 10 Clientdialoge für jede Kanalinstanz. Die Verwendung einer anderen Anzahl gemeinsam genutzter Dialoge kann jedoch die Leistung verbessern. Wenn Sie keine gemeinsamen Dialoge benötigen oder einen verteilten Server verwenden, setzen Sie **SHARECNV** auf 1. Wenn Sie vorhandene Clientanwendungen haben, die nicht ordnungsgemäß ausgeführt werden, wenn Sie **SHARECNV** auf 1 oder höher setzen, setzen Sie **SHARECNV** auf 0.

Informationen zu diesem Vorgang

Bei einigen Konfigurationen bringt die Verwendung von gemeinsamen Dialogen erhebliche Vorteile mit sich. Bei verteilten Servern ist die Verarbeitung von Nachrichten auf Kanälen, die die Standardkonfiguration von 10 gemeinsamen Datenaustauschvorgängen verwenden, jedoch im Durchschnitt 15% langsamer

als auf Kanälen, die keine gemeinsamen Dialoge verwenden. In einer MQI-Kanalinstanz, die Dialoge gemeinsam verwendet, werden alle Dialoge in einem Socket vom selben Thread empfangen. Wenn die Dialoge, die einen Socket gemeinsam nutzen, alle ausgelastet sind, sind die Dialoge miteinander verbunden, um den empfangenden Thread zu verwenden. Der Konflikt verursacht Verzögerungen, und in dieser Situation ist die Verwendung einer kleineren Anzahl von gemeinsamen Gesprächen besser.

Sie verwenden den Parameter **SHARECNV**, um die maximale Anzahl an Dialogen anzugeben, die über eine bestimmte TCP/IP-Clientkanalinstanz gemeinsam genutzt werden. Ausführliche Informationen zu allen möglichen Werten finden Sie unter [Supported IBM MQ client: Default behavior of client-connection and server-connection channels](#).

Wenn Sie **SHARECNV** auf 1 oder höher setzen, aktivieren Sie die folgenden Leistungsverbesserungen:

- Bidirektionale Überwachungssignale
- Administratorstop-quiet
- Vorauslesen
- Asynchrone-konsumieren durch Clientanwendungen

Wenn Sie keine gemeinsam genutzten Dialoge benötigen, bieten diese beiden Einstellungen die beste Leistung:

- **SHARECNV**(1).
- **SHARECNV**(0).

Anmerkungen:

- Wenn der Clientverbindungswert **SHARECNV** nicht dem Serververbindungswert **SHARECNV** entspricht, wird der niedrigere Wert verwendet.
- Wenn Anwendungen mit einer nicht wiedereintrittsfähigen Bibliothek verbunden oder kompiliert werden, wird der Wert **CURSHCNV** (0) ausgehandelt, auch wenn ein höherer Wert in **CLNTCONN** und **SVRCONN** festgelegt ist.

Führen Sie einen der folgenden Schritte aus, um die Leistung für eine bestimmte Kanalinstanz zu optimieren:

Prozedur

- Kanäle überwachen, die den Standardwert 10 für **SHARECNV** verwenden.

Die Standardeinstellung von **SHARECNV** (10) funktioniert in vielen Szenarios gut, ist jedoch möglicherweise nicht die optimale Einstellung für eine bestimmte Kanalinstanz. Beispiel: Bei verteilten Servern ist die Verarbeitung von Nachrichten auf Kanälen, die diese Einstellung verwenden, durchschnittlich 15% langsamer als auf Kanälen, die keine gemeinsamen Dialoge verwenden.

Um sicherzustellen, dass die Standardeinstellung für eine bestimmte Kanalinstanz geeignet ist, überwachen Sie die Leistung des Kanals mit dieser Einstellung.

- Legen Sie für **SHARECNV** den Wert 2 oder mehr fest.

Sie können **SHARECNV** (2) auf **SHARECNV** (999999999) setzen. Um sicherzustellen, dass die ausgewählte Einstellung für eine bestimmte Kanalinstanz geeignet ist, überwachen Sie die Leistung des Kanals mit der neuen Einstellung.

- Legen Sie für **SHARECNV** den Wert 1 fest.

Wenn Sie keine gemeinsam genutzten Dialoge benötigen, verwenden Sie diese Einstellung wann immer möglich. Es beseitigt Konflikte bei der Verwendung des empfangenden Threads, und Ihre Clientanwendungen können die Leistungsverbesserungen nutzen, die im Abschnitt "Über diese Task" beschrieben sind.

Mit dieser Einstellung wird die Leistung des verteilten Servers deutlich verbessert. Die Leistungsverbesserungen gelten für Clientanwendungen, die synchrone GET-Warteaufrufe ohne Vorauslesen ausgeben, z. B. C-Client-MQGET-Warteaufrufe. Wenn diese Clientanwendungen verbunden sind, verwendet der verteilte Server weniger Threads und weniger Speicher, und der Durchsatz wird erhöht.

Wenn mit einem Server Clients verbunden sind, die Dialoge über einen Socket gemeinsam nutzen, und Sie die Einstellung für gemeinsame Dialoge von SHARECNV (10) auf SHARECNV (1) verringern, hat dies folgende Auswirkungen:

- Höhere Socketverwendung auf dem Server.
- Höhere Kanalinstanzen auf dem Server.

In diesem Fall können Sie auch die Einstellungen für **MaxChannels** und **MaxActiveChannels** erhöhen.

Anmerkung: Sie können auch die Option MQCONNX festlegen MQCNO_NO_CONV_SHARING und die Anwendung mit einem Kanal verbinden, bei dem **SHARECNV** auf einen Wert größer als 1 gesetzt ist. Das Ergebnis ist dasselbe wie das Verbinden der Anwendung mit einem Kanal, bei dem **SHARECNV** auf 1 gesetzt ist.

- Legen Sie für SHARECNV den Wert 0 fest.

Die Kanalinstanz verhält sich genau so, als wäre es ein IBM WebSphere MQ 6.0 -Server- oder Clientverbindungskanal. Sie erhalten keine gemeinsamen Dialoge oder die Leistungsverbesserungen, die verfügbar sind, wenn Sie **SHARECNV** auf 1 oder höher setzen. Verwenden Sie den Wert 0 nur, wenn Clientanwendungen vorhanden sind, die nicht ordnungsgemäß ausgeführt werden, wenn Sie **SHARECNV** auf 1 oder höher setzen.

Zugehörige Konzepte

Unterstützter IBM MQ -Client: Standardverhalten von Clientverbindungs- und Serververbindungskanälen

Verteilte Publish/Subscribe-Netze optimieren

Verwenden Sie die Optimierungstipps in diesem Abschnitt, um die Leistung der verteilten Publish/Subscribe-Cluster und -Hierarchien von IBM MQ zu verbessern.

Zugehörige Konzepte

„Cluster überwachen“ auf Seite 325

In einem Cluster können Sie Anwendungsnachrichten, Steuernachrichten und Protokolle überwachen. Es gibt spezielle Überwachungsaspekte, wenn die Clusterauslastung zwischen zwei oder mehr Instanzen einer Warteschlange verteilt wird.

Direkte Publish/Subscribe-Clusterleistung

In direkt weitergeleiteten Publish/Subscribe-Clustern werden Informationen wie z. B. Clusterthemen und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dieser Prozess kann eine erhebliche zusätzliche Belastung für das System verursachen. Um die Auswirkungen des Cluster-Managements auf die Leistung zu reduzieren, können Sie Aktualisierungen zu Zeiten hoher Systemauslastung ausführen, eine viel kleinere Untergruppe von Warteschlangenmanagern definieren, die an Publish/Subscribe beteiligt sind, und dass ein "überlappender" Cluster oder ein Wechsel zur Verwendung des Topic-Host-Routing erfolgt.

In einem Publish/Subscribe-Cluster sind zwei Workloadquellen für einen Warteschlangenmanager vorhanden:

- Nachrichten für Anwendungsprogramme direkt verarbeiten.
- Handhabung von Nachrichten und Kanälen, die für die Verwaltung des Clusters benötigt werden.

In einem typischen Punkt-zu-Punkt-Cluster ist die Auslastung des Clustersystems weitgehend auf die Informationen beschränkt, die von den Mitgliedern des Clusters nach Bedarf explizit angefordert werden. Daher können Sie in einem anderen Punkt als einem sehr großen Punkt-zu-Punkt-Cluster, z. B. einer, der Tausende von Warteschlangenmanagern enthält, den Leistungseffekt der Verwaltung des Clusters weitgehend absehen. In einem direkt weitergeleiteten Publish/Subscribe-Cluster werden jedoch Informationen wie z. B. Clusterthemen, Warteschlangenmanager-Mitgliedschaften und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dies kann zu einer erheblichen zusätzlichen Belastung des

Systems führen. Daher müssen Sie die Auswirkungen der Clusterverwaltung auf die Leistung des Warteschlangenmanagers in Betracht ziehen, und zwar sowohl in der Ablaufsteuerung als auch in der Größe.

Leistungsmerkmale von Direkt-Routing-Clustern

Vergleichen Sie einen Punkt-zu-Punkt-Cluster mit einem direkt weitergeleiteten Publish/Subscribe-Cluster in Bezug auf die Kernverwaltungstasks.

Zunächst ein Punkt-zu-Punkt-Cluster:

1. Wenn eine neue Clusterwarteschlange definiert ist, werden die Zielinformationen an die vollständigen WS-Manager-Repositorys übertragen und nur an andere Cluster-Member gesendet, wenn sie zum ersten auf eine Clusterwarteschlange verweisen (z. B., wenn eine Anwendung versucht, sie zu öffnen). Diese Informationen werden dann lokal vom WS-Manager zwischengespeichert, um die Notwendigkeit zu entfernen, die Informationen bei jedem Zugriff auf die Warteschlange fern abzurufen.
2. Das Hinzufügen eines WS-Managers zu einem Cluster wirkt sich nicht direkt auf die Auslastung anderer WS-Manager aus. Die Informationen zum neuen Warteschlangenmanager werden in die vollständigen Repositorys verschoben, aber die Kanäle für den neuen Warteschlangenmanager von anderen WS-Managern im Cluster werden nur erstellt und gestartet, wenn der Datenverkehr zum oder vom neuen WS-Manager fließt.

Zusammengefasst bezieht sich die Last auf einen Warteschlangenmanager in einem Punkt-zu-Punkt-Cluster auf den Nachrichtendatenverkehr, der für Anwendungsprogramme verarbeitet wird, und hängt nicht direkt mit der Größe des Clusters zusammen.

Zweitens, ein Direct-Routing-Publish/Subscribe-Cluster:

1. Wenn ein neues Clusterthema definiert wird, werden die Informationen an die vollständigen WS-Manager-Repositorys und von dort aus direkt an alle Member des Clusters übertragen, wodurch die Kanäle für jedes Member des Clusters aus den vollständigen Repositorys gestartet werden, wenn dies noch nicht begonnen hat. Wenn es sich um das erste direkte Clusterthema handelt, wird jedes WS-Manager-Member Informationen zu allen anderen WS-Manager-Membern im Cluster gesendet.
2. Wenn eine Subskription für ein Clusterthema in einer neuen Themenzeichenfolge erstellt wird, werden die Informationen direkt von diesem Warteschlangenmanager an alle anderen Member des Clusters übertragen, wodurch Kanäle für jedes Member des Clusters von diesem Warteschlangenmanager aus gestartet werden, falls dies noch nicht begonnen hat.
3. Wenn ein neuer Warteschlangenmanager einem vorhandenen Cluster beiträgt, werden Informationen zu allen Clusterthemen (und allen WS-Manager-Membern, wenn ein direkter Clusterthema definiert ist) von den vollständigen WS-Managern des Repositorys an den neuen WS-Manager übertragen. Der neue Warteschlangenmanager synchronisiert dann die Kenntnisse aller Subskriptionen für Clusterthemen im Cluster mit allen Membern des Clusters.

Zusammengefasst wächst die Clustermanagementbelastung an jedem WS-Manager in einem direkt weitergeleiteten Publish/Subscribe-Cluster mit der Anzahl an Warteschlangenmanagern, Clusterthemen und Änderungen an Subskriptionen für verschiedene Themenzeichenfolgen im Cluster, unabhängig von der lokalen Verwendung dieser Clusterthemen in jedem Warteschlangenmanager.

In einem großen Cluster oder einer, in der die Geschwindigkeit der Subskriptionen hoch ist, kann diese Ebene der Clusterverwaltung für alle Warteschlangenmanager einen erheblichen Systemaufwand darstellen.

Reduzierung der Auswirkungen von Direct-Routing-Publish/Subscribe auf die Leistung

Wenn Sie die Auswirkungen der Clusterverwaltung auf die Leistung eines Direct-routed-Publish/Subscribe-Clusters verringern möchten, sollten Sie die folgenden Optionen berücksichtigen:

- Führen Sie Cluster-, Topic- und Subskriptionsaktualisierungen zu Off-Peak-Zeiten des Tages aus.
- Definieren Sie eine viel kleinere Untergruppe von Warteschlangenmanagern, die an Publish/Subscribe beteiligt sind, und stellen Sie sicher, dass ein "überlappendes" Cluster vorhanden ist. Dieser Cluster

ist dann der Cluster, in dem die Clusterthemen definiert sind. Obwohl einige WS-Manager jetzt in zwei Clustern vorhanden sind, wird die Gesamtwirkung von Publish/Subscribe reduziert:

- Die Größe des Publish/Subscribe-Clusters ist kleiner.
- Warteschlangenmanager, die nicht im Publish/Subscribe-Cluster enthalten sind, sind durch den Datenverkehr im Cluster-Management wesentlich weniger betroffen.

Wenn die vorherigen Optionen Ihre Leistungsprobleme nicht ausreichend lösen, sollten Sie stattdessen einen Publish/Subscribe-Cluster mit dem *Topic-Host* verwenden. Ein detaillierter Vergleich der direkten Routing- und Themenhostweiterleitung in Publish/Subscribe-Clustern finden Sie unter [Publish/Subscribe-Cluster entwerfen](#).

Zugehörige Konzepte

Thema Host für Publish/Subscribe-Clusterleistung bereitstellen

Ein Publish/Subscribe-Cluster im Themenhost gibt Ihnen präzise Kontrolle darüber, welche WS-Manager die einzelnen Themen hosten. Diese Themenhosts werden zu den *Routing*-Warteschlangenmanagern für diese Verzweigung der Themenstruktur. Darüber hinaus müssen Warteschlangenmanager ohne Subskriptionen oder Publisher keine Verbindung zu den Themenhosts herstellen. Diese Konfiguration kann die Anzahl der Verbindungen zwischen WS-Managern im Cluster und die Menge der Informationen, die zwischen den Warteschlangenmanagern übergeben werden, erheblich reduzieren.

Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

Subskriptionsleistung in Publish/Subscribe-Netzen

Verteiltes Publish/Subscribe in IBM MQ funktioniert durch die Weitergabe von Kenntnissen zur Position, an der Subskriptionen für verschiedene Themenzeichenfolgen im Warteschlangenmanagernetz erstellt wurden. Auf diese Weise kann der Warteschlangenmanager, auf dem eine Nachricht veröffentlicht wird, angeben, welche anderen WS-Manager eine Kopie der veröffentlichten Nachricht benötigen, damit sie mit ihren Subskriptionen übereinstimmen.

Thema Host für Publish/Subscribe-Clusterleistung bereitstellen

Ein Publish/Subscribe-Cluster im Themenhost gibt Ihnen präzise Kontrolle darüber, welche WS-Manager die einzelnen Themen hosten. Diese Themenhosts werden zu den *Routing*-Warteschlangenmanagern für diese Verzweigung der Themenstruktur. Darüber hinaus müssen Warteschlangenmanager ohne Subskriptionen oder Publisher keine Verbindung zu den Themenhosts herstellen. Diese Konfiguration kann die Anzahl der Verbindungen zwischen WS-Managern im Cluster und die Menge der Informationen, die zwischen den Warteschlangenmanagern übergeben werden, erheblich reduzieren.

Der Publish/Subscribe-Cluster eines Topic-Hosts verhält sich wie folgt:

- Die Themen werden manuell auf einzelnen WS-Managern des *Topic-Hosts* im Cluster definiert.
- Wenn eine Subskription für einen Cluster-WS-Manager ausgeführt wird, werden Proxy-Subskriptionen nur auf den Themenhosts erstellt.
- Wenn eine Anwendung Informationen zu einem Thema veröffentlicht, leitet der empfangende WS-Manager die Veröffentlichung an einen Warteschlangenmanager weiter, der das Thema hostet. Der Topic-Host sendet die Veröffentlichung dann an alle Warteschlangenmanager im Cluster, die gültige Subskriptionen für das Thema besitzen.

Eine detailliertere Einführung in das Thema Host-Routing finden Sie unter [Topic host routing in clusters](#).

Bei vielen Konfigurationen ist das Thema-Host-Routing eine besser geeignete Topologie als *direktes Routing*, da es die folgenden Vorteile bietet:

- Verbesserte Skalierbarkeit größerer Cluster. Nur die TOPICHOST-Warteschlangenmanager müssen mit allen anderen Warteschlangenmanagern im Cluster eine Verbindung herstellen können. Deshalb gibt es weniger Kanäle zwischen den Warteschlangenmanagern und zwischen den Warteschlangenmanagern weniger Datenverkehr in Verbindung mit der Publish/Subscribe-Verwaltung, als dies beim direkten

Routing der Fall ist. Wenn sich Subskriptionen auf einem Warteschlangenmanager ändern, müssen nur die Topic-Host-Warteschlangenmanager informiert werden.

- Größere Kontrollmöglichkeiten bei der physischen Konfiguration. Beim direkten Routing setzen alle Warteschlangenmanager sämtliche Rollen voraus und müssen daher gleichermaßen befähigt sein. Beim TOPICHOST-Routing wählen Sie explizit die TOPICHOST-Warteschlangenmanager aus. Damit können Sie sicherstellen, dass diese Warteschlangenmanager auf entsprechend leistungsfähigen Systemen aktiv sind, während die anderen Warteschlangenmanager auf weniger leistungsfähigen Systemen installiert werden können.

Beim TOPICHOST-Routing gelten für Ihr System jedoch bestimmte Einschränkungen:

- Die Systemkonfiguration und -wartung muss sorgfältiger geplant werden als dies beim DIRECT-Routing erforderlich ist. Sie müssen in der Themenstruktur die Punkte festlegen, die zu Clustern zusammengeslossen werden sollen; ebenso müssen Sie festlegen, wo im Cluster sich die Themendefinitionen befinden.
- Wenn ein neues Thema mit TOPICHOST-Routing definiert wird, werden die Informationen wie bei Themen, für die DIRECT-Routing definiert ist, an die Warteschlangenmanager mit vollständigem Repository und von dort direkt an alle Clustermitglieder übertragen. Dadurch werden von den vollständigen Repositories aus Kanäle zu jedem Clustermitglied gestartet (sofern dies noch nicht geschehen ist).
- Veröffentlichungen werden immer von einem Warteschlangenmanager, bei dem es sich nicht um einen Themenhost handelt, an einen Warteschlangenmanager, der als Themenhost dient, gesendet; dies ist auch dann der Fall, wenn im Cluster keine Subskriptionen vorhanden sind. Wenn daher davon ausgegangen werden kann, dass Subskriptionen vorliegen, oder wenn der Aufwand für die globale Konnektivität und für globale Informationen höher ist als der eventuell zusätzliche Datenverkehr beim Übertragen von Veröffentlichungen, sollten Sie Themen-Routing verwenden.
- Nachrichten, die auf Warteschlangenmanagern veröffentlicht werden, die keine Hosts sind, gehen nicht direkt an den Warteschlangen, der die Subskription hostet, sondern werden immer über einen TOPICHOST-Warteschlangenmanager weitergeleitet. Dadurch erhöhen sich der Gesamtaufwand für den Cluster und die Latenzzeit bei der Nachrichtenübertragung, wodurch sich die Leistung verschlechtert.

Anmerkung: Für bestimmte Konfigurationen können Sie diese Einschränkung sinnvoll entfernen, wie im Abschnitt [Topic-Host-Routing mit zentralisierten Publishern oder Subskribenten](#) beschrieben.

- Die Verwendung nur eines Warteschlangenmanagers als Themenhost stellt einen SPoF (Single Point of Failure) für alle Nachrichten dar, die zu einem Thema veröffentlicht werden. Durch eine Definition mehrerer Themenhosts wird ein solcher SPoF ausgeschlossen. Bei Verwendung mehrerer Hosts ändert sich allerdings die Reihenfolge, in der veröffentlichte Nachrichten für Subskriptionen empfangen werden.
- TOPICHOST-Warteschlangenmanager erzeugen ein zusätzliches Nachrichtenvolumen, weil sie Veröffentlichungsdatenverkehr von mehreren Warteschlangenmanagern verarbeiten müssen. Dieses Volumen kann verringert werden, indem mehrere TOPICHOSTs für ein einzelnes Thema verwendet werden (wobei die Reihenfolge der Nachrichten nicht beibehalten wird) oder indem verschiedene Warteschlangenmanager als Hosts für weitergeleitete Themen für verschiedene Zweige der Themenstruktur verwendet werden.

Thema Host-Routing mit zentralisierten Publishern oder Subskribenten

Wenn Sie den zusätzlichen "Hop" entfernen möchten, wenn Veröffentlichungen immer über einen Topic-Host-Warteschlangenmanager an Subskriptionen weitergeleitet werden, konfigurieren Sie die Publisher oder die Subskriptionen auf demselben Warteschlangenmanager, auf dem sich das Thema befindet. Dieser Ansatz bringt maximale Leistungsvorteile in den folgenden beiden Fällen:

- Topics mit vielen Publishern und wenigen Abonnements. In diesem Fall hosten die Subskriptionen auf dem Topic-Host-Warteschlangenmanager.
- Topics mit wenigen Publishern und vielen Abonnements. In diesem Fall hosten die Publisher auf dem Topic-Host-Warteschlangenmanager.

Die folgende Abbildung zeigt einen Topic-Host-Warteschlangenmanager, der auch die Subskriptionen enthält. Dieser Ansatz entfernt den zusätzlichen "Hop" zwischen dem Bereitsteller und dem Subskribenten

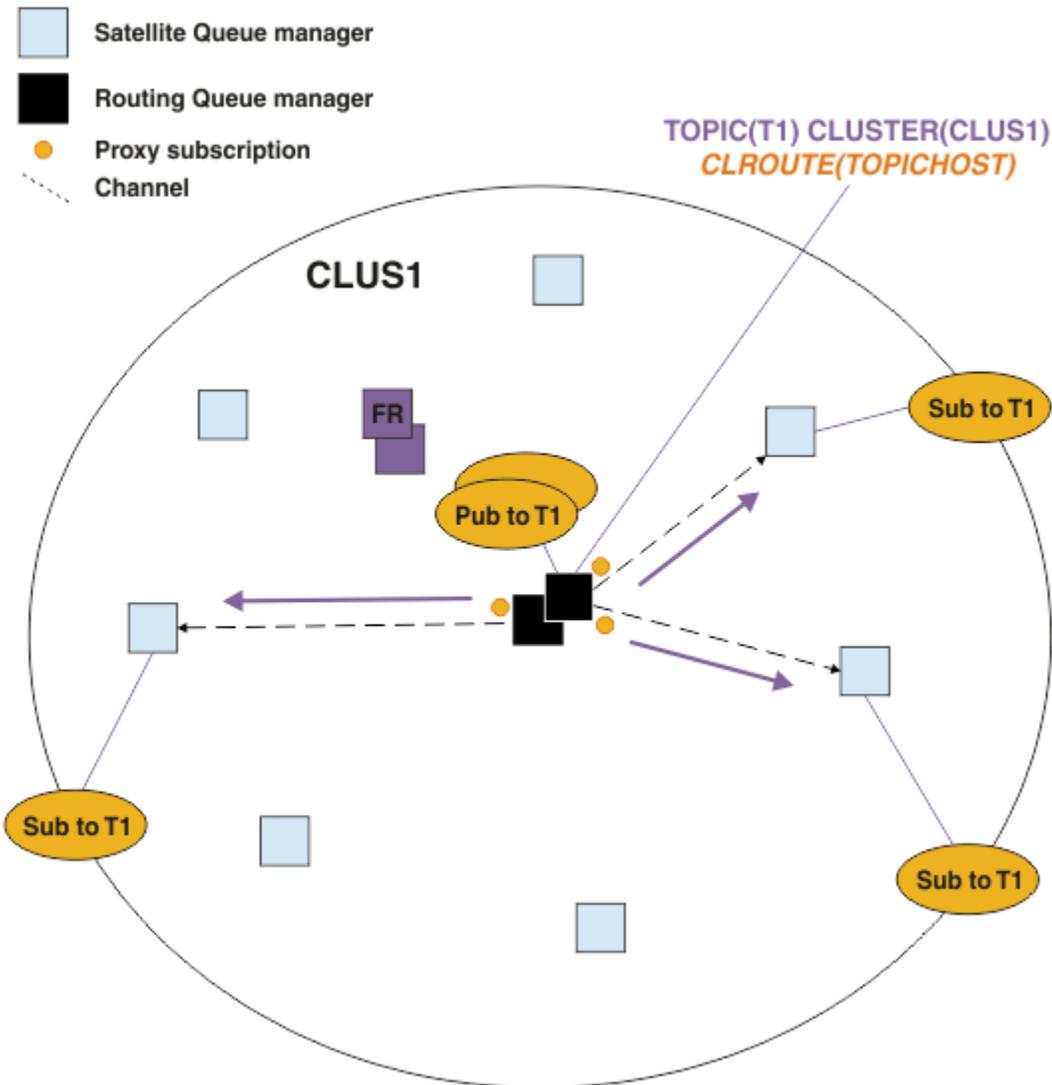


Abbildung 28. Hosting-Veröffentlichungen zu einem Topic-Host-Warteschlangenmanager

Zugehörige Konzepte

Direkte Publish/Subscribe-Clusterleistung

In direkt weitergeleiteten Publish/Subscribe-Clustern werden Informationen wie z. B. Clusterthemen und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dieser Prozess kann eine erhebliche zusätzliche Belastung für das System verursachen. Um die Auswirkungen des Cluster-Managements auf die Leistung zu reduzieren, können Sie Aktualisierungen zu Zeiten hoher Systemauslastung ausführen, eine viel kleinere Untergruppe von Warteschlangenmanagern definieren, die an Publish/Subscribe beteiligt sind, und dass ein "überlappendes" Cluster oder ein Wechsel zur Verwendung des Topic-Host-Routing erfolgt.

Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

Subskriptionsleistung in Publish/Subscribe-Netzen

Verteiltes Publish/Subscribe in IBM MQ funktioniert durch die Weitergabe von Kenntnissen zur Position, an der Subskriptionen für verschiedene Themenzeihenfolgen im Warteschlangenmanagernetz erstellt wurden. Auf diese Weise kann der Warteschlangenmanager, auf dem eine Nachricht veröffentlicht wird,

angeben, welche anderen WS-Manager eine Kopie der veröffentlichten Nachricht benötigen, damit sie mit ihren Subskriptionen übereinstimmen.

Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

In einer Punkt-zu-Punkt-Messaging-Topologie ist die Beziehung zwischen Nachrichtenkonsumenten und Nachrichtenproduzenten leicht verständlich. Sie können Schätzungen von Nachrichtenproduktion und -verbrauch, Warteschlange nach Warteschlange, Kanal nach Kanal abrufen. Wenn ein Mangel an Ausgewogenheit besteht, werden die Engpässe schnell erkannt und dann behoben.

Es ist schwieriger zu prüfen, ob Publisher und Subskribenten in einer Publish/Subscribe-Topologie ausgeglichen sind. Beginnen Sie bei jeder Subskription, und arbeiten Sie mit den WS-Managern, die Publisher für das Thema verwenden, zurück. Berechnen Sie die Anzahl der Veröffentlichungen, die von jedem WS-Manager zu jedem Subskribenten fließen.

Jede Veröffentlichung, die mit einer Subskription in einem fernen Warteschlangenmanager (basierend auf Proxy-Subskriptionen) übereinstimmt, wird in eine Übertragungswarteschlange gestellt. Wenn mehrere ferne WS-Manager über Proxy-Subskriptionen für diese Veröffentlichung verfügen, werden mehrere Kopien der Nachricht in eine Übertragungswarteschlange gestellt, die jeweils für einen anderen Senderkanal bestimmt sind.

In einem Publish/Subscribe-Cluster werden diese Veröffentlichungen an die SYSTEM.INTER.QMGR.PUBS-Warteschlange auf den fernen Warteschlangenmanagern gerichtet, die die Subskriptionen hosten. In einer Hierarchie richtet sich jede Veröffentlichung an die SYSTEM.BROKER.DEFAULT.STREAM-Warteschlange oder an alle anderen Datenstromwarteschlangen, die in SYSTEM.QPUB.SUB.QUEUE.NAMELIST auf den fernen Warteschlangenmanagern aufgelistet sind. Jeder WS-Manager verarbeitet Nachrichten, die in dieser Warteschlange ankommen, und stellt sie den richtigen Subskriptionen auf diesem Warteschlangenmanager zu.

Überwachen Sie daher die Auslastung an den folgenden Stellen, an denen Engpässe auftreten können:

- Überwachen Sie die Auslastung in den einzelnen Subskriptionswarteschlangen.
 - Dieser Engpass impliziert, dass die subskribierende Anwendung die Veröffentlichungen nicht so schnell konsumiert, wie sie veröffentlicht werden.
- Überwachen Sie die Auslastung in der SYSTEM.INTER.QMGR.PUBS-Warteschlange oder in den Datenstromwarteschlangen.
 - Dieser Engpass impliziert, dass der Warteschlangenmanager Veröffentlichungen von einem oder mehreren fernen Warteschlangenmanagern schneller empfängt, als er sie an die lokalen Subskriptionen verteilen kann.
 - Wenn Sie in einem Topic-Host-Warteschlangenmanager bei der Verwendung von Topic-Host-Routing in einem Cluster sehen, sollten Sie zusätzliche Warteschlangenmanager-Topic-Hosts verwenden, die es ermöglichen, die Workload der Veröffentlichung auf diese zu balancieren. Dies wirkt sich jedoch auf die Nachrichtenreihenfolge in allen Veröffentlichungen aus. Weitere Informationen finden Sie unter [Topic-Host-Routing unter Verwendung mehrerer Topic-Hosts für ein einzelnes Thema](#).
- Überwachen Sie die Auslastung auf den Kanälen zwischen dem Veröffentlichungswarteschlangenmanager und den subskribierenden WS-Managern, die von den Übertragungswarteschlangen des Veröffentlichungswarteschlangenmanagers gespeist werden.
 - Dieser Engpass impliziert, dass entweder ein oder mehrere Kanäle nicht aktiv sind, oder Nachrichten werden schneller in den lokalen WS-Manager veröffentlicht, als die Kanäle sie an den fernen Warteschlangenmanager übergeben können.
 - Wenn Sie einen Publish/Subscribe-Cluster verwenden, können Sie zusätzliche Clusterempfängerkanäle auf dem Zielwarteschlangenmanager definieren. Auf diese Weise kann die Auslastung der Veröffentlichungen auf diese Weise ausgeglichen werden. Dies wirkt sich jedoch auf die Nachrichten-

reihenfolge in Veröffentlichungen aus. Stellen Sie außerdem in Betracht, eine Konfiguration mit mehreren Clusterübertragungswarteschlangen zu verschieben, da dadurch die Leistung unter bestimmten Umständen verbessert werden kann.

- Wenn die Veröffentlichungsanwendung eine Publish/Subscribe-Schnittstelle in der Warteschlange verwendet, überwachen Sie die Last an (a) der SYSTEM.BROKER.DEFAULT.STREAM-Warteschlange und an allen anderen Datenstromwarteschlangen, die in SYSTEM.QPUBSUB.QUEUE.NAMELIST aufgelistet sind. und (b) die SYSTEM.BROKER.DEFAULT.SUBPOINT-Warteschlange und alle anderen Unterpunktwarteschlangen, die in der SYSTEM.QPUBSUB.SUBPOINT.NAMELIST aufgelistet sind.
 - Dieser Engpass impliziert, dass Nachrichten von lokalen Veröffentlichungsanwendungen schneller gestellt werden, als der lokale WS-Manager die Nachrichten verarbeiten kann.

Zugehörige Konzepte

Direkte Publish/Subscribe-Clusterleistung

In direkt weitergeleiteten Publish/Subscribe-Clustern werden Informationen wie z. B. Clusterthemen und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dieser Prozess kann eine erhebliche zusätzliche Belastung für das System verursachen. Um die Auswirkungen des Cluster-Managements auf die Leistung zu reduzieren, können Sie Aktualisierungen zu Zeiten hoher Systemauslastung ausführen, eine viel kleinere Untergruppe von Warteschlangenmanagern definieren, die an Publish/Subscribe beteiligt sind, und dass ein "überlappender" Cluster oder ein Wechsel zur Verwendung des Topic-Host-Routing erfolgt.

Thema Host für Publish/Subscribe-Clusterleistung bereitstellen

Ein Publish/Subscribe-Cluster im Themenhost gibt Ihnen präzise Kontrolle darüber, welche WS-Manager die einzelnen Themen hosten. Diese Themenhosts werden zu den *Routing*-Warteschlangenmanagern für diese Verzweigung der Themenstruktur. Darüber hinaus müssen Warteschlangenmanager ohne Subskriptionen oder Publisher keine Verbindung zu den Themenhosts herstellen. Diese Konfiguration kann die Anzahl der Verbindungen zwischen WS-Managern im Cluster und die Menge der Informationen, die zwischen den Warteschlangenmanagern übergeben werden, erheblich reduzieren.

Subskriptionsleistung in Publish/Subscribe-Netzen

Verteiltes Publish/Subscribe in IBM MQ funktioniert durch die Weitergabe von Kenntnissen zur Position, an der Subskriptionen für verschiedene Themenzeichenfolgen im Warteschlangenmanagernetz erstellt wurden. Auf diese Weise kann der Warteschlangenmanager, auf dem eine Nachricht veröffentlicht wird, angeben, welche anderen WS-Manager eine Kopie der veröffentlichten Nachricht benötigen, damit sie mit ihren Subskriptionen übereinstimmen.

„Cluster überwachen“ auf Seite 325

In einem Cluster können Sie Anwendungsnachrichten, Steuernachrichten und Protokolle überwachen. Es gibt spezielle Überwachungsaspekte, wenn die Clusterauslastung zwischen zwei oder mehr Instanzen einer Warteschlange verteilt wird.

Subskriptionsleistung in Publish/Subscribe-Netzen

Verteiltes Publish/Subscribe in IBM MQ funktioniert durch die Weitergabe von Kenntnissen zur Position, an der Subskriptionen für verschiedene Themenzeichenfolgen im Warteschlangenmanagernetz erstellt wurden. Auf diese Weise kann der Warteschlangenmanager, auf dem eine Nachricht veröffentlicht wird, angeben, welche anderen WS-Manager eine Kopie der veröffentlichten Nachricht benötigen, damit sie mit ihren Subskriptionen übereinstimmen.

Dieser Ansatz minimiert das Senden von veröffentlichten Nachrichten an Warteschlangenmanager, auf denen keine übereinstimmenden Subskriptionen vorhanden sind. Die Weitergabe des Subskriptionswissens kann jedoch zu einem erheblichen Systemaufwand werden, wenn die Anzahl der Themenzeichenfolgen, die subskribiert werden, hoch ist und sich durch häufige Subskriptionserstellung und -löschung ständig ändert.

Sie können die Leistung beeinflussen, indem Sie anpassen, wie Veröffentlichungen und Subskriptionen in Ihrem Publish/Subscribe-Netz ausgeführt werden. Wenn Ihr Netzdatenverkehr nur wenige Veröffentlichungen enthält und die Erstellung, Löschung oder Änderung der Schnellschreibung beendet ist, können Sie die Subskription von Subskriptionsinformationen an alle Warteschlangenmanager stoppen

und alle Veröffentlichungen an alle WS-Manager im Netz weiterleiten. Sie können auch den Fluss von Proxy-Subskriptionen und Veröffentlichungen für ein bestimmtes Thema zwischen verbundenen Warteschlangenmanagern einschränken, den Fluss von Proxy-Subskriptionen, die Platzhalterzeichen enthalten, beschränken und die Anzahl und die transiente Art von Themenzeichenfolgen reduzieren.

Weitergabe von Einzelsubskriptionen und *überall veröffentlichen*

Überall veröffentlichen ist eine Alternative zur Weitergabe einzelner Abonnements. Bei der individuellen Weitergabe werden nur Veröffentlichungen, die über eine übereinstimmende Subskription in einem Warteschlangenmanager verfügen, an diesen Warteschlangenmanager weitergeleitet. Mit *überall veröffentlichen* werden alle Veröffentlichungen an alle WS-Manager im Netz weitergeleitet. Die empfangenden WS-Manager stellen dann die Veröffentlichungen zur Verfügung, die mit lokalen Subskriptionen übereinstimmen.

Weitergabe von Einzelabonn

Dieser Mechanismus führt dazu, dass der Datenverkehr zwischen den Veröffentlichungsdaten des Warteschlangenmanagers am wenigsten liegt, da nur die Veröffentlichungen gesendet werden, die mit Subskriptionen für einen Warteschlangenmanager übereinstimmen.

Es gilt jedoch auch Folgendes:

- Für jede einzelne Themenzeichenfolge, die subskribiert ist, wird eine Proxy-Subskription an andere WS-Manager in der Publish/Subscribe-Topologie gesendet. Die Gruppe der Warteschlangenmanager richtet sich nach dem verwendeten Routing-Modell, wie im Abschnitt Verteiltes Publish/Subscribe-Netz planen beschrieben.
 - Dieser Messaging-Systemaufwand kann von Bedeutung sein, wenn viele Tausende von Subskriptionen erstellt oder gelöscht werden (z. B. wenn alle nicht permanenten Subskriptionen nach einem Neustart eines Warteschlangenmanagers erneut erstellt werden) oder wenn sich die Subskriptionsgruppe schnell ändert und jeweils eine andere Themenzeichenfolge verwendet wird.
 - Die Anzahl der WS-Manager, an die die Proxy-Subskription weitergegeben wird, wirkt sich auch auf den Umfang des Systemaufwands aus.
- Proxy-Subskriptionen werden mit asynchronem Messaging an andere WS-Manager geleitet. Dies hat die folgende Wirkung:
 - Es gibt eine Verzögerung zwischen der Erstellung einer Subskription und der Erstellung, Bereitstellung und Verarbeitung der Proxy-Subskription durch die anderen Warteschlangenmanager.
 - Nachrichten, die in diesen Warteschlangenmanagern in diesem Intervall veröffentlicht werden, werden nicht an die ferne Subskription zugestellt.

Überall veröffentlichen

Bei diesem Mechanismus gibt es keinen Systemaufwand für die Proxy-Subskription pro Themenzeichenfolge auf dem System. Dies bedeutet, dass die Erstellung, Löschung oder Änderung der Schnellschubskription nicht zu einer erhöhten Netzauslastung und -verarbeitung führt.

Es besteht auch keine Verzögerung zwischen der Erstellung einer Subskription und den Veröffentlichungen, die an einen Warteschlangenmanager fließen, da alle Veröffentlichungen an alle Warteschlangenmanager flossen. Daher gibt es kein Fenster, in dem die Veröffentlichungen nicht an neu erstellte ferne Subskriptionen zugestellt werden.

Es gilt jedoch auch Folgendes:

- Wenn Sie alle Veröffentlichungen an alle Warteschlangenmanager in der Publish/Subscribe-Topologie senden, kann es zu einem übermäßigen Datenaustausch im Netz kommen, in dem Veröffentlichungen nicht über übereinstimmende Subskriptionen auf den einzelnen Warteschlangenmanagern verfügen.
 - Je größer die Anzahl der WS-Manager in der Topologie ist, um so größer ist der Systemaufwand.

Sie sollten die Verwendung des Mechanismus *publish überall* in Betracht ziehen, wenn Sie erwarten, dass eine Veröffentlichung von einem erheblichen Teil Ihrer Warteschlangenmanager subskribiert wird, oder wenn die Proxy-Subskriptionsüberschrift wegen der Häufigkeit von Subskriptionsänderungen zu hoch sind. Sie sollten die individuelle Proxy-Abonnementweiterleitung verwenden, wenn Sie einen erhöhten

Messaging-Datenverkehr erleben, wenn die Veröffentlichungen an alle Warteschlangenmanager gesendet werden, und nicht an die Warteschlangenmanager mit übereinstimmenden Subskriptionen.

Sie können das Verhalten *publish überall* auf jeder Ebene in der Themenstruktur festlegen. Um *überall veröffentlichen* zu aktivieren, setzen Sie den Parameter **PROXYSUB** für ein übergeordnetes Themenobjekt auf ERZWINGEN. Dies führt zu einer einzelnen Proxy-Subskription mit Platzhalterzeichen, die alle Topics unter diesem Themenobjekt in der Themenstruktur abgleicht. Wenn es für ein Clusterthemenobjekt festgelegt wird, wird das Attribut **PROXYSUB (FORCE)** an jeden Warteschlangenmanager im Netz weitergegeben, nicht nur an den Warteschlangenmanager, auf dem das Thema definiert wurde.

Anmerkung: Bei Verwendung in einer Hierarchie legen Sie **PROXYSUB (FORCE)** individuell auf jedem Warteschlangenmanager fest, sodass der Topologiemechanismus natürlich die Anzahl der Kanäle begrenzt. Wenn jedoch in einem Cluster verwendet wird, können viele zusätzliche Kanäle gestartet werden:

- In einem Topic-Host-Routing-Cluster werden die Kanäle von jedem WS-Manager zu jedem Topic-Host-Warteschlangenmanager gestartet.
- In einem direkt weitergeleiteten Cluster werden Kanäle von jedem WS-Manager zu jedem anderen Warteschlangenmanager gestartet.

Der Systemaufwand für das Starten vieler Kanäle ist in einem direkt weitergeleiteten Cluster am stärksten ausgeprägt und kann zu Leistungsproblemen führen. Siehe [„Direkte Publish/Subscribe-Clusterleistung“](#) auf Seite 390.

Andere Möglichkeiten, den Fluss von Proxy-Subskriptionen und Veröffentlichungen zwischen verbundenen Warteschlangenmanagern zu beschränken

Themenzeichenfolgen konsolidieren

Die Verwendung vieler unterschiedlicher, transienter Themenzeichenfolgen führt zu einem gewissen Verwaltungsaufwand für jeden Warteschlangenmanager in dem System, an dem Publisher oder Subskriptionen angehängt sind. Sie sollten die Verwendung von Themenzeichenfolgen regelmäßig bewerten, um festzustellen, ob sie konsolidiert werden können. Durch die Verringerung der Anzahl und der transienten Merkmale von Themenzeichenfolgen und damit der Publisher und Subskriptionen für diese werden die Auswirkungen auf das System reduziert.

Schränken Sie die Veröffentlichungs- und Subskription

Für ein bestimmtes Thema können Sie die Einstellungen von [Veröffentlichungsbereich](#) und [Subskriptionsbereich](#) verwenden, damit Veröffentlichungen und Subskriptionen für den Warteschlangenmanager, für den sie definiert sind, lokal bleiben.

Blocksubskriptionen für Themen mit Platzhalterzeichen

Sie können den Ablauf von Proxy-Subskriptionen, die Platzhalterzeichen enthalten, beschränken, indem Sie das Attribut **Topic PLATZHALTER** auf BLOCK setzen. Weitere Informationen finden Sie unter [Platzhalterzeichen in Proxy-Subskriptionen](#).

Siehe auch [„Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen“](#) auf Seite 396

Proxy-Subskriptionsdatenverkehr in Clustern überwachen

Wenn Sie die Auslastung des Systems durch den Proxy-Subskriptionsdatenverkehr berücksichtigen, überwachen Sie zusätzlich zu den in [„Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen“](#) auf Seite 396 aufgeführten Warteschlangen auch die folgenden Clusterwarteschlangen:

- Die Warteschlange SYSTEM.INTER.QMGR.FANREQ auf dem Subskribentenwarteschlangenmanager.
- Die Warteschlange SYSTEM.INTER.QMGR.CONTROL auf allen anderen Warteschlangenmanagern im Cluster.

Jeder signifikante Nachrichtenrückstand in diesen Warteschlangen impliziert, dass entweder die Rate der Subskriptionsänderung für das System zu hoch ist oder dass ein Warteschlangenmanager im Cluster nicht ordnungsgemäß funktioniert. Wenn Sie vermuten, dass das Problem bei einem bestimmten WS-Manager liegt, überprüfen Sie, ob die Publish/Subscribe-Unterstützung für diesen Warteschlangenmanager nicht inaktiviert ist. Siehe **PSMODE** in [ALTER QMGR \(Warteschlangenmanager\)](#).

Zugehörige Konzepte

Direkte Publish/Subscribe-Clusterleistung

In direkt weitergeleiteten Publish/Subscribe-Clustern werden Informationen wie z. B. Clusterthemen und Proxy-Subskriptionen an alle Mitglieder des Clusters übertragen, unabhängig davon, ob alle Cluster-WS-Manager aktiv an Publish/Subscribe-Messaging beteiligt sind. Dieser Prozess kann eine erhebliche zusätzliche Belastung für das System verursachen. Um die Auswirkungen des Cluster-Managements auf die Leistung zu reduzieren, können Sie Aktualisierungen zu Zeiten hoher Systemauslastung ausführen, eine viel kleinere Untergruppe von Warteschlangenmanagern definieren, die an Publish/Subscribe beteiligt sind, und dass ein "überlappender" Cluster oder ein Wechsel zur Verwendung des Topic-Host-Routing erfolgt.

Thema Host für Publish/Subscribe-Clusterleistung bereitstellen

Ein Publish/Subscribe-Cluster im Themenhost gibt Ihnen präzise Kontrolle darüber, welche WS-Manager die einzelnen Themen hosten. Diese Themenhosts werden zu den *Routing*-Warteschlangenmanagern für diese Verzweigung der Themenstruktur. Darüber hinaus müssen Warteschlangenmanager ohne Subskriptionen oder Publisher keine Verbindung zu den Themenhosts herstellen. Diese Konfiguration kann die Anzahl der Verbindungen zwischen WS-Managern im Cluster und die Menge der Informationen, die zwischen den Warteschlangenmanagern übergeben werden, erheblich reduzieren.

Ausgleich von Erzeugern und Verbrauchern in Publish/Subscribe-Netzen

Ein wichtiges Konzept für die asynchrone Messaging-Leistung ist *balance*. Wenn die Nachrichtenkonsumenten nicht mit den Nachrichtenerzeugern in Einklang stehen, besteht die Gefahr, dass sich ein Rückstand nicht konsumierter Nachrichten aufbauen kann und die Leistung mehrerer Anwendungen ernsthaft beeinträchtigt wird.

Proxy-Subskriptionen in einem Publish/Subscribe-Netz

Reduzieren der Anzahl unerwünschter Themen in der Themenstruktur

Die Leistung eines Publish/Subscribe-Systems wird verbessert, indem die Anzahl unerwünschter Themen in der Themenstruktur reduziert wird. Was ist ein unerwünschtes Thema und wie können Sie sie entfernen?

Sie können eine große Anzahl von Themen erstellen, ohne die Leistung zu beeinträchtigen. Einige Methoden zum Verwenden von Publish/Subscribe führen jedoch dazu, dass Themenstrukturen kontinuierlich erweitert werden. Eine außergewöhnlich große Anzahl von Themen wird einmal erstellt und nie wieder verwendet. Die wachsende Anzahl von Themen kann zu einem Leistungsproblem werden.

Wie können Sie Designs vermeiden, die zu einer großen und wachsenden Zahl unerwünschter Themen führen? Was können Sie tun, um dem WS-Manager zu helfen, unerwünschte Themen aus der Themenstruktur zu entfernen?

Der WS-Manager erkennt ein unerwünschtes Thema, da es seit 30 Minuten ungenutzt ist. Der WS-Manager entfernt nicht verwendete Themen aus der Themenstruktur für Sie. Die Dauer von 30 Minuten kann geändert werden, indem das Warteschlangenmanagerattribut **TREELIFE** geändert wird. Sie können dem Warteschlangenmanager helfen, unerwünschte Themen zu entfernen, indem Sie sicherstellen, dass der WS-Manager nicht mehr verwendet wird. Der Abschnitt „Was ist ein nicht verwendetes Thema?“ auf Seite [400](#) erläutert, was ein nicht verwendetes Thema ist.

Ein Programmierer, der jede Anwendung entwirft und insbesondere eine lange laufende Anwendung entwirft, berücksichtigt seine Ressourcennutzung: Wie viel Ressourcen benötigt das Programm, gibt es irgendwelche unbegrenzten Anforderungen, und irgendwelche Ressourcenlecks? Topics sind eine Ressource, die Publish/Subscribe-Programme verwenden. Scrüönisieren Sie die Verwendung von Themen wie alle anderen Ressourcen, die ein Programm verwendet.

Was ist ein nicht verwendetes Thema?

Bevor Sie definieren, was ein nicht verwendetes Thema ist, was zählt eigentlich zu einem Thema?

Wenn eine Themenzeichenfolge, wie z. B. USA/Alabama/Auburn, in ein Thema konvertiert wird, wird das Thema zur Themenbaumstruktur hinzugefügt. Weitere Themenknoten und die entsprechenden The-

men werden in der Baumstruktur erstellt, falls erforderlich. Die Themenzeichenfolge USA/Alabama/Auburn wird in eine Baumstruktur mit drei Themen konvertiert.

- USA
- USA/Alabama
- USA/Alabama/Auburn

Um alle Themen in der Themenstruktur anzuzeigen, verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('#') TYPE(TOPIC)`.

Ein nicht verwendetes Thema in der Themenstruktur weist die folgenden Eigenschaften auf.

Sie ist einem Themenobjekt nicht zugeordnet.

Ein Verwaltungsthemenobjekt verfügt über eine Themenzeichenfolge, die sie einem Thema zuordnet. Falls beim Definieren des Themenobjekts Alabama das Thema USA/Alabama, dem es zugeordnet werden soll, nicht vorhanden ist, wird das Thema aus der Themenzeichenfolge erstellt. Wenn das Thema vorhanden ist, werden das Themenobjekt und das Thema unter Verwendung der Themenzeichenfolge verknüpft.

Es hat keine ständige Veröffentlichung

Ein Thema mit einer ständigen Veröffentlichung entsteht, wenn ein Publisher eine Nachricht mit der Option `MQPMO_RETAIN` in ein Thema einreicht.

Verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama') RETAINED`, um zu überprüfen, ob USA/Alabama über eine ständige Veröffentlichung verfügt. Die Antwort lautet YES oder NO.

Mit dem **runmqsc** -Befehl `CLEAR TOPICSTR('USA/Alabama') CLTRTYPE(RETAINED)` können Sie eine ständige Veröffentlichung aus USA/Alabama entfernen.

Es hat keine untergeordneten Themen

USA/Alabama/Auburn ist ein Thema ohne untergeordnete Themen. USA/Alabama/Auburn ist das direkt untergeordnete Thema von USA/Alabama.

Zeigen Sie die direkten untergeordneten Elemente von USA/Alabama mit dem **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama+')` an.

Es sind keine aktiven Publisher für den Knoten vorhanden.

Ein aktiver Publisher für einen Knoten ist eine Anwendung, die das Thema für die Ausgabe geöffnet hat.

Eine Anwendung öffnet beispielsweise das Themenobjekt mit dem Namen **Alabama** mit den Optionen zum Öffnen `MQOO_OUTPUT`.

Verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACT-CONN`, um aktive Publisher für USA/Alabama und alle untergeordneten Elemente anzuzeigen.

Es sind keine aktiven Subskribenten für den Knoten vorhanden.

Ein aktiver Subskribent kann entweder eine permanente Subskription oder eine Anwendung sein, die eine Subskription für ein Thema bei `MQSUB` registriert und nicht geschlossen hat.

Verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACT-CONN`, um aktive Subskriptionen für USA/Alabama anzuzeigen.

Verwenden Sie den **runmqsc** -Befehl `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACT-CONN`, um aktive Subskriptionen für USA/Alabama und alle untergeordneten Elemente anzuzeigen.

Anzahl der Themen in einer Themenstruktur verwalten

Zusammenfassend gibt es eine Reihe von Möglichkeiten, die Anzahl der Themen in einer Themenstruktur zu verwalten.

TPCOUNT anzeigen

Verwenden Sie den **runmqsc** -Befehl `DISPLAY PUBSUB ALL` regelmäßig, um die Eigenschaft **TPCOUNT** anzuzeigen. Dies ist die Anzahl der Themenknoten in der Themenstruktur. Wenn die Zahl größer wird, kann dies darauf hinweisen, dass ein kürzerer **TREELIFE** erforderlich ist, oder dass eine Neugestaltung der Themen selbst erforderlich ist.

TREELIFE ändern

Ein nicht verwendetes Thema hat standardmäßig eine Lebensdauer von 30 Minuten. Sie können die Lebensdauer eines nicht verwendeten Themas kleiner machen.

Der Befehl **runmqsc**, `ALTER QMGR TREELIFE(900)`, reduziert beispielsweise die Lebensdauer eines nicht verwendeten Themas von 30 Minuten auf 15 Minuten.

Ausnahmsweise Neustart des WS-Managers

Wenn der Warteschlangenmanager erneut gestartet wird, wird die Themenstruktur von Themenobjekten, Knoten mit ständigen Veröffentlichungen und permanenten Subskriptionen erneut initialisiert. Themen, die durch den Betrieb von Veröffentlichungs- und Subskribentenprogrammen erstellt wurden, werden eliminiert.

Als letztes Mittel, wenn das Wachstum in unerwünschten Themen die Ursache von Leistungsproblemen in der Vergangenheit ist, starten Sie den Warteschlangenmanager erneut.

Zugehörige Konzepte

[Themenstrukturen](#)

Aspera gateway kann die Leistung in Netzen mit hoher Latenz verbessern

IBM Aspera faspio Gateway stellt einen schnellen TCP/IP-Tunnel bereit, der den Netzdurchsatz für IBM MQ erheblich erhöhen kann.

Mit dem Aspera gateway kann die Leistung von Warteschlangenmanagerkanälen verbessert werden. Es ist besonders effektiv, wenn das Netz eine längere Latenzzeit hat oder dazu neigt, Pakete zu verlieren, und wird normalerweise dazu verwendet, die Verbindung zwischen Warteschlangenmanagern in verschiedenen Rechenzentren zu beschleunigen.

Bei ohnehin schnellen Netzen, in denen keine Paketverluste auftreten, wird die Netzleistung durch das Aspera gateway jedoch eher beeinträchtigt. Sie sollten die Netzleistung daher unbedingt vor und nach der Definition einer Aspera gatewayverbindung überprüfen.

Ein Warteschlangenmanager, der auf einer beliebigen berechtigten Plattform ausgeführt wird, kann über einen Aspera gateway eine Verbindung herstellen. Das Gateway selbst wird in Red Hat® oder Ubuntu Linux oder Windows bereitgestellt.

Weitere Informationen finden Sie unter [Aspera gateway -Verbindung unter Linux oder Windows definieren](#).

Bemerkungen

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

Möglicherweise bietet IBM die in diesem Dokument beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf Produkte, Programme oder Services von IBM bedeuten nicht, dass nur Produkte, Programme oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Défense
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East & Africa
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmieretechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos ohne Zahlung an IBM in jeder Form kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben sind. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

Informationen zu Programmierschnittstellen

Die bereitgestellten Informationen zur Programmierschnittstelle sollen Sie bei der Erstellung von Anwendungssoftware für dieses Programm unterstützen.

Dieses Handbuch enthält Informationen zu geplanten Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zum Abrufen der Services von IBM MQ zu schreiben.

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

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

Marken

IBM, das IBM Logo, ibm.com, sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml. Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein.

Microsoft und Windows sind eingetragene Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

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

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Dieses Produkt enthält Software, die von Eclipse Project (<https://www.eclipse.org/>) entwickelt wurde.

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



Teilenummer:

(1P) P/N: