

9.4

IBM MQ のモニターとパフォーマンス

IBM

注記

本書および本書で紹介する製品をご使用になる前に、[391 ページの『特記事項』](#)に記載されている情報をお読みください。

本書は、IBM® MQ バージョン 9 リリース 4、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様が IBM に情報を送信する場合、お客様は IBM に対し、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で情報を使用または配布する非独占的な権利を付与します。

© Copyright International Business Machines Corporation 2007 年, 2024.

目次

モニターおよびパフォーマンス	5
IBM MQ ネットワークのモニター.....	5
OpenTelemetry の統合.....	5
キュー・マネージャーのヘルス・チェックの動作.....	6
イベント・モニター.....	9
メッセージ・モニター.....	59
アカウンティング・メッセージと統計メッセージ.....	138
アプリケーション・アクティビティ・トレース.....	204
モニターとアクティビティ・トレースに関するシステム・トピック.....	291
リアルタイム・モニター.....	302
クラスターのモニター.....	314
アプリケーション・バランシングのモニター.....	317
Monitoring performance and resource usage on z/OS.....	319
IBM MQ ネットワークのチューニング.....	375
クライアントとサーバーの接続チャンネルのチューニング.....	375
分散パブリッシュ/サブスクライブ・ネットワークのチューニング.....	377
トピック・ツリー内の不要なトピック数の削減.....	387
Aspera gateway を使用して、待ち時間の長いネットワークでのパフォーマンスの向上を図る....	389
特記事項	391
プログラミング・インターフェース情報.....	392
商標.....	392

IBM MQ モニターおよびパフォーマンス

このセクションにあるモニター情報やガイダンス、および特定のチューニングのヒントを使用して、キュー・マネージャー・ネットワークのパフォーマンスを向上させます。

このタスクについて

ネットワークをモニターすることで、キュー・マネージャー・ネットワークのサイズおよび複雑さに応じたさまざまな情報を取得することができます。そのような情報と特定のチューニングのヒントで得られる情報を使用して、ネットワークのパフォーマンスを調整できます。

IBM MQ ネットワークのモニター

キュー・マネージャー・ネットワークがどのように稼働しているかについての統計および他の特定の情報を取得するために、さまざまなモニター技法を IBM MQ で使用できます。このセクションにあるモニター情報およびガイダンスを使用して、キュー・マネージャー・ネットワークのパフォーマンスを向上させます。

キュー・マネージャー・ネットワークをモニターする理由の例を以下にリストします。

- キュー・マネージャー・ネットワークにある問題を検出する。
- キュー・マネージャー・ネットワークにある問題の原因の判別を支援する。
- キュー・マネージャー・ネットワークの効率を向上させる。
- キュー・マネージャー・ネットワークの実行方法を習得できるようにする。
- キュー・マネージャー・ネットワークが正しく実行されていることを確認する。
- 特定のイベントが発生したときにメッセージを生成する。
- メッセージ・アクティビティを記録する。
- メッセージが最後に確認された場所を特定する。
- キュー・マネージャー・ネットワークの各種統計をリアルタイムで確認する。
- 監査証跡を生成する。
- アプリケーション・リソースの使用を報告する。
- キャパシティー・プランニングを行う。

▼ 9.4.0 ▼ ALW OpenTelemetry の統合

IBM MQ を OpenTelemetry トレース・システムと統合することができます。

OpenTelemetry トレース

OpenTelemetry トレースを使用すると、アプリケーションがデータ・フロー内でどのように動作しているかを監視できます。データ・フローは、多くの場合、いくつかの異なるアプリケーションを取り込むことができます。トレースにより、ジャーニー全体を表示し、個々のアプリケーションの動作に関する洞察を得ることができます。IBM MQ には、OpenTelemetry トレース・システムとの統合を可能にするトレース・サービスが用意されています。

IBM MQ OpenTelemetry トレース・サービスは、IBM MQ API 出口として実装されます。これは、IBM サポート・プログラムとして提供されます。これは、IBM MQ ライセンスの一部としてトレース・サービスを使用し、サポートを受けるためのライセンスを持っていることを意味します。IBM MQ ライセンスでは、IBM Instana サポート・プログラムの IBM MQ トレース出口コンポーネントの使用のみが許可されていることに注意してください。

出口の使用に起因する問題は、IBM Instana モニター・システムおよび IBM Instana ライセンスでトレース出口を使用している場合を除き、IBM MQ サポートに報告する必要があります。

この出口は、<https://ibm.biz/mqinstanaexit> からダウンロードできます。

IBM MQ Open Telemetry トレース・サービスのインストールおよび構成

IBM MQ トレースをインストールして構成する方法については、[IBM MQ Tracing](#) を参照してください。

IBM Instana IBM MQ 出口でサポートされるプラットフォームについては、[オンプレミスでサポートされるプラットフォーム IBM MQ](#) を参照してください。

Multi キュー・マネージャーのヘルス・チェックの動作

キュー・マネージャーは、安定した信頼性の高いパフォーマンスを確保するために、定期的なヘルス・チェックを実行します。このトピックでは、キュー・マネージャーが行うヘルス・チェックの一部について説明し、環境要件に基づいてそれらを構成する方法について説明します。

ほとんどの環境では、デフォルト構成が適しているため、これらの検査の頻度を変更する必要はありません。デフォルト設定を使用する場合でも、問題が検出されたときにキュー・マネージャーがどのように動作するか、およびチェックの失敗の原因となる可能性がある環境上の問題を理解することが役立ちます。このトピックは、これらの動作のいくつかを説明することを目的としています。



注意: IBM サポートからの指示がない限り、これらの検査の頻度を変更しないでください。

キュー・マネージャーのさまざまなコンポーネントは、不整合を検出して解決するためにさまざまな手段を使用します。このトピックは、そのようなメカニズムをすべて説明することを目的としたものではありません。例えば、IBM MQ プロセスはさまざまなメカニズムを使用して、依存先の他のプロセスが引き続き実行されるようにします。説明されている動作は、環境またはその他の予期しない状況を発見するために実行コントローラーによって定期的に行われる動作です。(実行コントローラーは、他のほとんどのキュー・マネージャー・プロセスを開始および管理する 1 次 IBM MQ プロセスです。) これらは定期的な検査であるため、適切なチューニング・パラメーターを設定することによってある程度変更できる特定の間隔で行われます。

記述されている検査の一部は、専用のヘルス・チェック・スレッドによって行われます。ヘルス・チェック・スレッド自体に問題が検出されると、警告メッセージ [AMQ5066](#) がキュー・マネージャーのエラー・ログに書き込まれます。

このトピックで説明されている動作は、将来のリリースで変更される可能性があります。例えば、特定のプラットフォームまたは構成で別のデフォルト値がより安定していることが確認された場合などです。

一般的なヘルス・チェック

キュー・マネージャーは、一定の間隔でさまざまな検査を実行します。デフォルトでは、これらの検査は 10 秒ごとに実行されます(場合によっては、検査でエラーを報告する前に 2 サイクルが許可されることがあります。その場合、検査の間隔は 20 秒になります)。検査中に、キュー・マネージャーは、キュー・マネージャーの一部として実行されるさまざまなプロセスが引き続き実行されていることを確認します。ネイティブ HA キュー・マネージャーの場合、キュー・マネージャーがスタンバイ・インスタンスに正常にデータを複製しているかどうかを検査します。

この時点でクリティカル検査が失敗した場合(例えば、`amqzmuc0` プロセスが実行されなくなった場合)、キュー・マネージャーは実行を継続できなくなります。ただし、ほとんどの検査は、不要になったシステム・リソースをタイディアップするために行われ、単にメッセージがキュー・マネージャーのエラー・ログに書き込まれる可能性があります。

ほとんどの場合、これらの一般的なヘルス・チェックの頻度を変更する必要はありません。キュー・マネージャーまたは稼働環境の大部分のイベントは、一般的なヘルス・チェック・プロセスによる検出を必要とせず、即時に検出されます。このプロセスは、キュー・マネージャー内の他の場所で検出されないものに対する定期的な検査として機能します。必要に応じて、`ECHearBeatLen` チューニング・パラメーターを使用して頻度を構成できます。最小値は 10000 ミリ秒 (10 秒) です。最大値は 60000 ミリ秒 (60 秒) です。最大値 60000 に設定すると、特定のチェックで 2 分の遅延が発生する可能性があります。

ログの進行状況が行われていることを確認します。

キュー・マネージャーは、ログへの書き込みが妥当な速度で行われているかどうかを検査します。これは、ロガーのパフォーマンスが最適であるかどうかの検査ではなく、さらに注意が必要な可能性がある条件を検出するように設計されています。例えば、ログ・ファイルが保管されているディスクの速度が特に遅い場合や、コンテナ化された環境でキュー・マネージャーがすべての作業を実行するのに十分な CPU 時間を受け取っていない場合などです。

この検査が失敗した場合、キュー・マネージャーが取るアクションは、使用されているキュー・マネージャーのタイプによって異なります。

• 非 HA キュー・マネージャーの場合:

- `xecL_W_PERFORMANCE_BOTTLENECK FDC` が書き込まれます。これは、システムの一部がさらに注意を必要とする可能性があることを示すために使用できます。キュー・マネージャーは実行中のままです。エラー・ディレクトリーに `xecL_W_PERFORMANCE_BOTTLENECK FDC` がある場合は、ストレージ・チームまたはプラットフォーム・チームと協力して、IBM MQ を実行するために基礎となるシステム・リソースが十分であるかどうかを理解する必要があります。IBM MQ がオーバーコミットされたノード上のコンテナで実行されている場合、IBM MQ は、すべてのメッセージング・ワークロードを実行するのに十分なスケジュールされた CPU 時間を受信しない可能性があります。
- IBM MQ 9.3.0 以降、警告メッセージ `AMQ5068W` がキュー・マネージャーのエラー・ログに書き込まれ、`xecL_W_PERFORMANCE_BOTTLENECK FDC` は書き込まれません。`AMQ5068W` メッセージがログに表示される場合は、ストレージ・チームまたはプラットフォーム・チームと協力して、基礎となるシステム・リソースが IBM MQ の実行に十分であるかどうかを理解する必要がある場合があります。IBM MQ がオーバーコミットされたノード上のコンテナで実行されている場合、IBM MQ は、すべてのメッセージング・ワークロードを実行するのに十分なスケジュールされた CPU 時間を受信しない可能性があります。5 つの `AMQ5068W` 警告メッセージが連続して書き込まれる場合、`xecL_W_PERFORMANCE_BOTTLENECK FDC` が書き込まれます。

• 複数インスタンス・キュー・マネージャーの場合:

- ログ進行ヘルス・チェックが失敗すると、1 次インスタンスは終了します。スタンバイ・インスタンスが使用可能な場合、スタンバイ・インスタンスが開始され、1 次インスタンスになります。
- IBM MQ 9.3.0 以降、1 次インスタンスは、終了前にスタンバイ・インスタンスが使用可能かどうかを検査します。スタンバイ・キュー・マネージャーが使用可能な場合、1 次インスタンスへのフェイルオーバーは終了します。さらに、警告メッセージ `AMQ5068W` がキュー・マネージャーのエラー・ログに書き込まれます。

• ネイティブ HA キュー・マネージャーでは、このチェックは非 HA キュー・マネージャーと同じように動作します。

• RDQM (複製データ・キュー・マネージャー) では、このチェックは非 HA キュー・マネージャーと同じように動作します。

IBM MQ ログの進行に関する問題は、キュー・マネージャー自体のパフォーマンスの問題が原因である可能性があります。

デフォルトでは、この検査は 60 秒ごとに行われますが、キュー・マネージャーは検査の 2 サイクル待機してからアクションを実行します。これは、デフォルト設定では、キュー・マネージャーがエラー・メッセージを書き込む (HA キュー・マネージャーの場合はフェイルオーバーする) 前に 2 分が経過する必要があることを意味します。

ほとんどの場合、ファイル・システムが低速であるか、キュー・マネージャーに少量の CPU 時間が割り振られている場合でも、デフォルトの動作が適しています。これは、[ファイル・ロック \(8 ページの『ファイル・ロックがまだ保持されていることを検査します。』を参照\)](#) や基本ファイル・システム操作などの他の検査によって、この検査が行われる前に 1 次インスタンスがフェイルオーバーするためです。必要に応じて、**LivenessHeartBeatLen** チューニング・パラメーターを使用して、このチェックの頻度を構成することができます。これに構成できる最大値は 600 秒 (10 分) です。最小値 0 は、チェックを完全に無効にする効果があります。非 HA キュー・マネージャーの場合、検査の唯一の影響は、キュー・マネージャーのエラー・ログに追加の警告メッセージがあることです。複数インスタンス・キュー・マネージャーの場合、**LivenessHeartBeatLen** を構成して、キュー・マネージャーの 1 次インスタンスがより迅速にフェイルオーバーするように (値を小さくすることによって)、または徐々にフェイルオーバーするように (値

を大きくすることによって) することができます。ログ進行状況検査の頻度を減らすために値を大きくすると、ご使用の環境でファイル・システムの入出力が非常に遅くなることがありますが、キュー・マネージャーの1次インスタンスを実行したままにしておきたい場合に役立ちます。これは、スタンバイ・インスタンスに自動的に再接続するように設計されていないアプリケーションがあり、それらを再始動するために手操作による介入が必要な場合に役立ちます。

注: **ECHearBeatLen**が増加した場合、これは **LivenessHeartBeatLen** チェックのタイミングに影響します。一般的なヘルス・チェックが実行されると、ログの進行状況チェックが行われます。そのため、一般的なヘルス・チェック (**ECHearBeatLen**) の頻度を減らすと、構成された **LivenessHeartBeatLen** から最大 30 秒後にログの進行状況チェックが行われる可能性があります。

汎用ロギング・ファイル・システム・パフォーマンス

V 9.4.0

IBM MQ 9.4.0 以降、キュー・マネージャーのエラー・ログに警告メッセージ **AMQ6729W** が出されるのは、このストレージに対する一般的な読み取り/書き込み操作に予想よりも長い時間がかかっていると思われる場合です。**AMQ_IODELAY** 環境変数を使用すると、ストレージ・パフォーマンスの問題の診断に役立つ診断とタイミングを微調整したり、そのような遅延に対する許容度を高めることができます。詳しくは、[AMQ_IODELAY](#)、[AMQ_IODELAY_INMS](#)、および [AMQ_IODELAY_FFST](#) を参照してください。

ファイル・ロックがまだ保持されていることを検査します。

複数インスタンス・キュー・マネージャーの場合、実行コントローラーは定期的に、1次複数インスタンス・ファイルの排他ロックを保持していることを確認します。多くの場合、NFS サーバーの問題が原因でロックが失われると、1次インスタンスは(この検査が行われる前に) ほぼ即時にフェイルオーバーします。ファイル・システムの異常問題が発生した場合に1次キュー・マネージャーがフェイルオーバーすることを確認するために、追加の定期的なファイル・ロック検査が行われます。

デフォルトでは、これらのファイル・ロック検査は 20 秒ごとに行われます。必要に応じて、**FileLockHeartBeatLen** チューニング・パラメーターを設定することで、この値を変更できます。チューニング・パラメーターのデフォルト値は 10 秒です(キュー・マネージャーは、20 秒ごとにチェックするデフォルトの動作になるアクションを実行する前に、2 サイクルのチェックを許可します)。チューニング・パラメーターの最小値は 10 秒、最大値は 600 秒 (10 分) です。

注: **ECHearBeatLen**が増加している場合は、**FileLockHeartBeatLen** チェックのタイミングに影響します。ファイル・ロック検査は、一般的なヘルス・チェックが実行される時に行われるため、一般的なヘルス・チェック (**ECHearBeatLen**) の頻度を減らすと、ファイル・ロック検査が構成された **FileLockHeartBeatLen** の最大 30 秒後に行われる可能性があります。

ユーザー・アプリケーションの正常性の検査

キュー・マネージャーは、実行されなくなったローカルにバインドされたアプリケーションが終了前に MQDISC MQI 呼び出しを実行したかどうかを定期的に検査します。これらの検査は、6 ページの『一般的なヘルス・チェック』で説明されている一般的なヘルス・チェックと同時に実行されます。したがって、このような検査のデフォルト間隔は 10000 ミリ秒 (10 秒) であり、**ECHearBeatLen** チューニング・パラメーターの値を変更すると、それらの検査が行われる頻度に変更されます。この検査は主に、接続されているアプリケーションに関連付けられているリソースが解放されていることを確認するためのものであり、HA キュー・マネージャーまたは非 HA キュー・マネージャーが終了したり、代替インスタンスにフェイルオーバーしたりすることはありません。

MQDISC MQI 呼び出しを発行せずに終了した IBM MQ クライアント・アプリケーションは、エージェント・プロセスによって個別に検出され、接続に関連するすべてのリソースが解放されます。

関連概念

[高可用性の構成](#)

イベント・モニター

イベント・モニターは、キュー・マネージャー・ネットワーク内で発生する観測イベントを検出する処理です。観測イベントとは、キュー・マネージャーまたはチャンネル・インスタンスが検出するイベントが論理的に組み合わせられたものです。このようなイベントが発生すると、イベント・メッセージという特殊なメッセージがキュー・マネージャーまたはチャンネル・インスタンスによってイベント・キューに書き込まれます。

IBM MQ 観測イベントからは、キュー・マネージャーに関するエラー、警告、その他の重要なイベントについての情報が得られます。これらのイベントを使用してキュー・マネージャー・ネットワーク内のキュー・マネージャーの操作をモニターすることで、以下の目的が達成されます。

- キュー・マネージャー・ネットワークにある問題を検出する。
- キュー・マネージャー・ネットワークにある問題の原因の判別を支援する。
- 監査証跡を生成する。
- キュー・マネージャーの状態変更に対応する。

関連資料

[12 ページの『イベント・タイプ』](#)

このページでは、キュー・マネージャーまたはチャンネル・インスタンスが報告できる観測イベントのタイプを示します。

[イベント・メッセージ参照](#)

[イベント・メッセージの形式](#)

IBM MQ イベント・メッセージのパブリッシュ

イベント・メッセージをパブリッシュするように IBM MQ を準備する方法。

このタスクについて

イベント・メッセージは、SYSTEM.ADMIN.<feature name>.EVENT という特殊な名前のキューに書き込まれます。

これらのイベント・キューに関して注意する必要がある要点は、大切なのはその名前であるということです。デフォルトでは、キュー・マネージャー上で、すべてのイベント・キューがローカル・キューとして定義されています。ただし、すべてのイベントが専用のイベント処理キュー・マネージャーに集まるように、これらのキューを削除して、リモート・キューなどとして再定義することが可能です。またはその代わりに、トピック・オブジェクトを示している別名キューを使用することもできます。

どちらの場合でも、リダイレクト技法を使用するためには、イベント・キューを読み取るアプリケーションで読み取り元のキューの名前がハードコーディングされていないことが必要です。そのため、アプリケーションが読み取るキューの構成が可能でなければなりません。

以下のコマンドは、イベント・メッセージがパブリッシュされるようにイベント・キューを再定義する方法を、次の前提を使用して示しています。前提は次のとおりです。

- イベントの使用を開始していないこと。または、
- 既存のイベント・キューからすべてのメッセージを削除して、以下の手順を実行する前に、ローカル・キューを削除していること。

以下の手順では、QMGR および CHANNEL イベント・キューが再定義される方法だけを示していますが、この方法はすべてのイベントに対しても使用できます。

注: トピック・ストリングは、必要に応じてアプリケーションを、ワイルドカードを使用してすべてのイベントにサブスクライブすることも、特定のイベントにサブスクライブすることもできるように設計されています。

手順

次のコマンドを発行します。

```

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)

```

イベントを読み取るアプリケーションがすべてのキューからイベント・メッセージを読み取ることができると想定すると、上で定義されたいずれかのキューから読み取るように、必要に応じてそのアプリケーションを再構成することができます。

DEFINE SUB コマンドの PSPROP(NONE) 構成は、パブリッシュ/サブスクライブ・エンジンによって追加されるメッセージ・プロパティ (MQTopicString など) が イベント・メッセージに追加されないようにします。これにより、既存のアプリケーションは変更なしで動作を続けることができます。

さらに、管理 DEFINE SUB コマンドを使用する代わりに代替方法として、アプリケーションは [MQSUB](#) 呼び出しを使用して情報を受け取ることにより、直接サブスクライブすることもできます。

これで、複数のアプリケーションが、キュー・マネージャーによりイベント内に出された情報を取り込むことができるようになりました。

観測イベント

観測イベントは、キュー・マネージャーまたはチャンネル・インスタンスが検出する条件と、イベント・メッセージと呼ばれる特別メッセージをイベント・キューに入れる条件を論理的に組み合わせたものです。

IBM MQ 観測イベントからは、キュー・マネージャーに関するエラー、警告、その他の重要なイベントについての情報が得られます。これらのイベントを使用して、キュー・マネージャーの操作をモニターできます (Tivoli® NetView® for z/OS® などの他の方式を使用して)。

[11 ページの図 1](#) に、観測イベントの概念を図示します。

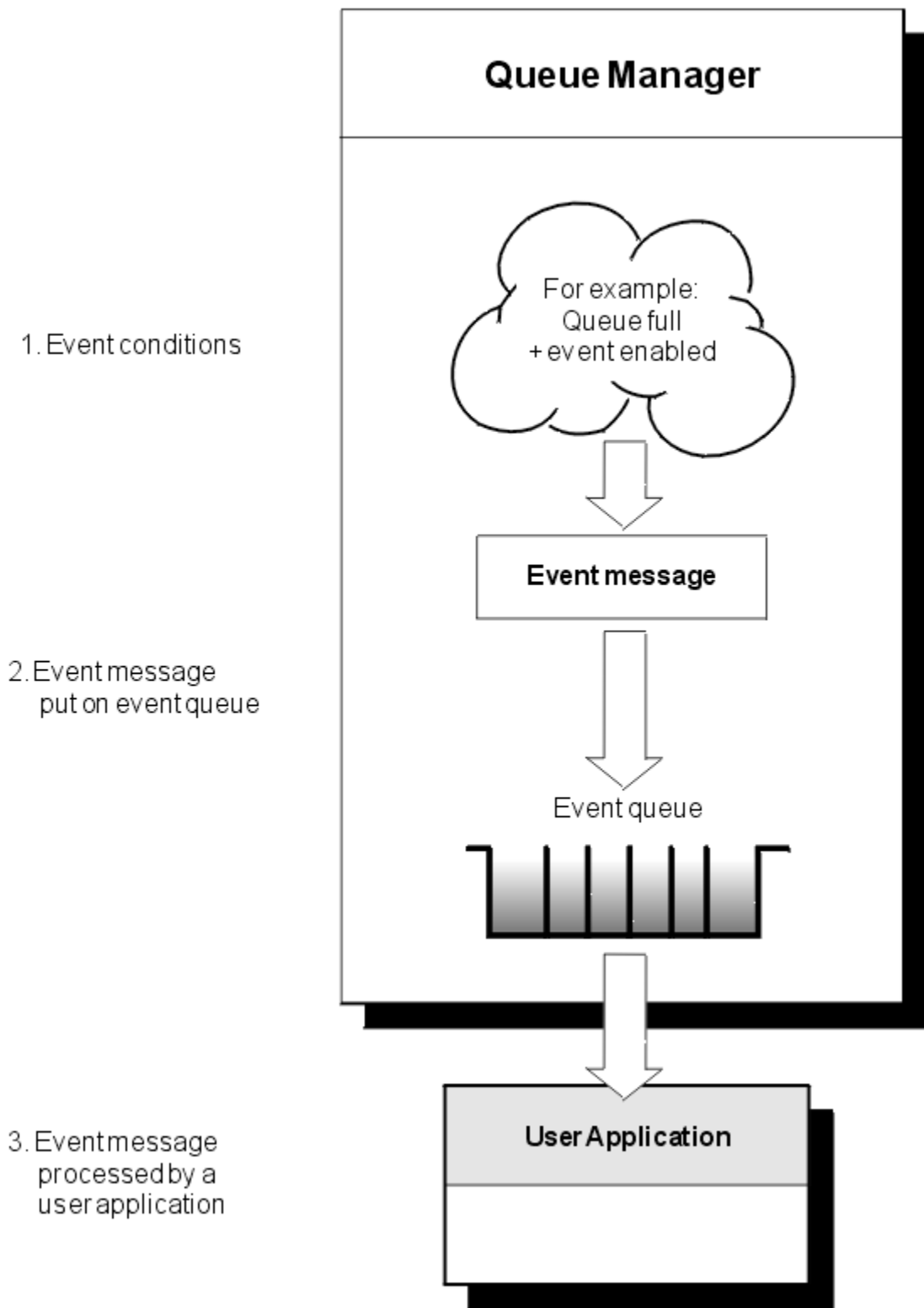


図 1. 観測イベントについて

イベント・モニター・アプリケーション

イベントを使用してキュー・マネージャーをモニターするアプリケーションには、以下のプロビジョンが含まれている必要があります。

1. ネットワーク内のキュー・マネージャー間のチャンネルのセットアップ。

- 必要なデータ変換の実行。データ変換に関する通常の規則が適用されます。例えば、UNIX システムのキュー・マネージャーにおけるイベントを z/OS キュー・マネージャーからモニターする場合は、必ず EBCDIC から ASCII へ変換します。

イベント・キューによるイベント通知

イベントが発生すると、キュー・マネージャーはイベント・メッセージを、該当するイベント・キュー (定義されている場合) に入れます。イベント・メッセージにはイベントに関する情報が含まれています。それらの情報は、以下のステップを実行する適切な MQI アプリケーション・プログラムを作成することによって取り出すことができます。

- キューからメッセージを取得します。
- メッセージを処理してイベント・データを取り出します。

関連情報にイベント・メッセージの形式に関する説明があります。

イベントを引き起こす条件

観測イベントが実行される可能性のある条件の例を以下にリストします。

- キューのメッセージ数のしきい値に達した。
- チャンネル・インスタンスが開始または停止した。
- キュー・マネージャーがアクティブになったか、または停止要求された。
- IBM MQ for IBM i, AIX®, Linux®, and Windows システムで許可されていないユーザー ID を指定して、アプリケーションがキューをオープンしようとした。
- オブジェクトが作成、削除、変更、またはリフレッシュされた。
- MQSC コマンドまたは PCF コマンドが正常に実行された。
- キュー・マネージャーが新しいログ・エクステンションへの書き込みを開始した。
- イベント条件が満たされた場合に、送達不能キューにメッセージが書き込まれた。

関連概念

[24 ページの『パフォーマンス・イベント』](#)

パフォーマンス・イベントは、指定したキューを使用するアプリケーションのパフォーマンスに影響のある条件に関連しています。パフォーマンス・イベントの有効範囲はキューです。1つのキューに対する **MQPUT** 呼び出しおよび **MQGET** 呼び出しは、別のキュー上のパフォーマンス・イベントの生成に影響を与えません。

[56 ページの『観測イベントをモニターするためのサンプル・プログラム \(Multiplatforms\)』](#)

amqsevt は IBM MQ for Multiplatforms に付属しており、キュー・マネージャーが作成できる観測イベントをフォーマットします。このプログラムは、イベント・キューからメッセージを読み取り、人が読めるストリングにフォーマットします。

イベント・タイプ

このページでは、キュー・マネージャーまたはチャンネル・インスタンスが報告できる観測イベントのタイプを示します。

IBM MQ の観測イベントには、以下のタイプがあります。

- キュー・マネージャー・イベント
- チャンネル・イベントとブリッジ・イベント
- パフォーマンス・イベント
- 構成イベント
- コマンド・イベント
- ロガー・イベント
- ローカル・イベント

イベントの各カテゴリーには、キュー・マネージャーごとに固有のイベント・キューがあります。そのカテゴリーのイベントが発生すると、すべて同じキューにイベント・メッセージが書き込まれることになります。

イベント・キュー

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

キューに入るメッセージの発生元

キュー・マネージャー・イベント

チャンネル・イベント

パフォーマンス・イベント

構成イベント

コマンド・イベント

ロガー・イベント

パブリッシュ/サブスクライブに関連した取得イベント。マルチキャストでのみ使用されます。詳しくは、[マルチキャスト・アプリケーション・モニター](#)を参照してください。

ご使用のシステム管理アプリケーションに観測イベントを組み込むことにより、複数のキュー・マネージャーや多くの異なるノードで、また複数の IBM MQ アプリケーションに対して、アクティビティをモニターできます。特に、[13 ページの図 2](#) に示すように、単一ノード (IBM MQ イベントをサポートするノード) からシステム内のすべてのノードをモニターできます。

観測イベントは、ユーザー作成のレポート・メカニズムを介して、管理アプリケーションにレポートできます。オペレーターは、管理アプリケーションでそれらのイベントを見ることができます。

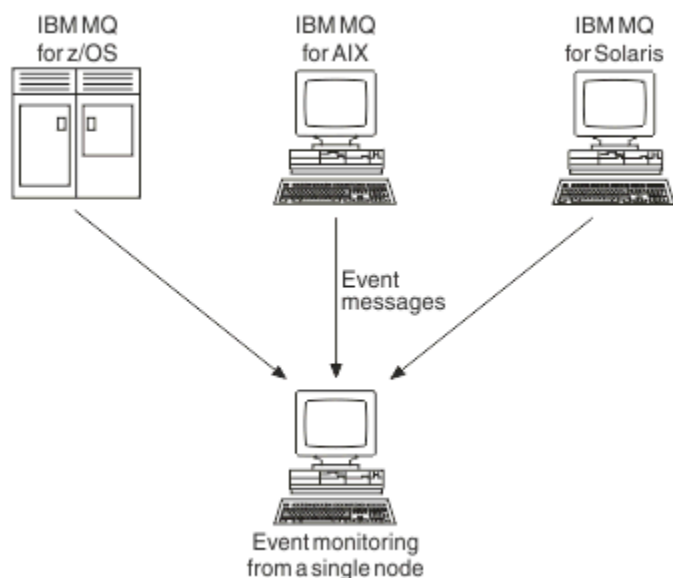


図 2. 単一のノードから異なるプラットフォームにあるキュー・マネージャーをモニターする

観測イベントを使用すると、他の管理ネットワークのエージェントとして動作するアプリケーション (Tivoli NetView for z/OS など) でレポートをモニターし、適切なアラートを生成することもできます。

キュー・マネージャー・イベント

キュー・マネージャー・イベントは、キュー・マネージャー内のリソースの使用に関係しています。例えば、アプリケーションが、存在しないキューにメッセージを書き込もうとすると、キュー・マネージャー・イベントが生成されます。

キュー・マネージャー・イベントが実行される可能性のある条件の例を以下に示します。

- アプリケーションが MQI 呼び出しを発行し、それが失敗した場合。その呼び出しの理由コードは、イベント・メッセージの理由コードと同じです。

キュー・マネージャーの内部動作中、例えばレポート・メッセージの生成中に、同様の条件が発生することがあります。イベント・メッセージの理由コードは、どのアプリケーションとも関連していない場合でも、MQI 理由コードと一致することがあります。イベント・メッセージの理由コードが MQI 理由コードに類似していても、そのイベントが、アプリケーションからの MQI 呼び出しの失敗によって発生したとは限りません。

- コマンドがキュー・マネージャーに対して発行され、このコマンドの処理が原因でイベントが発生した場合。以下に例を示します。

- キュー・マネージャーが停止または開始した。
- コマンドを入力したが、関連ユーザー ID では、そのコマンドに関して許可が得られない。

IBM MQ は、キュー・マネージャー・イベントのメッセージを SYSTEM.ADMIN.QMGR.EVENT キュー。以下のキュー・マネージャー・イベント・タイプをサポートします。

ALW 権限 (AIX, Linux, and Windows のみ)

権限イベントは、権限についての報告を行います。例えば、必要とされる権限を持っていないアプリケーションがキューをオープンしようとしたり、必要な権限のないユーザー ID でコマンドを入力したような場合です。権限イベント・メッセージには、次のイベント・データを含めることができます。

- [非許可 \(タイプ 1\)](#)
- [非許可 \(タイプ 2\)](#)
- [非許可 \(タイプ 3\)](#)
- [非許可 \(タイプ 4\)](#)
- [不許可 \(タイプ 5\)](#)
- [不許可 \(タイプ 6\)](#)

すべての権限イベントは、AIX, Linux, and Windows でのみ有効です。

禁止

禁止イベントは、書き込みまたは読み取りが禁止されているキューに対して、またはパブリッシュが禁止されているトピックに対して、MQPUT 操作または MQGET 操作が試行されたことを示します。禁止イベント・メッセージには、次のイベント・データを含めることができます。

- [読み取り禁止](#)
- [書き込み禁止](#)

ローカル

例えばオブジェクトが未定義であるなどの理由で、アプリケーションまたはキュー・マネージャーがローカル・キューまたはその他のローカル・オブジェクトにアクセスできない場合、キュー・マネージャーはローカル・イベント・メッセージを生成することができます。ローカル・イベント・メッセージには、次のイベント・データを含めることができます。

- [別名基本キュー・タイプ・エラー](#)
- [不明別名基本キュー](#)
- [不明オブジェクト名](#)

リモート

例えば伝送キューが正しく定義されていないなどの理由で、アプリケーションまたはキュー・マネージャーが別のキュー・マネージャー上のリモート・キューにアクセスできない場合、キュー・マネージャーはリモート・イベント・メッセージを生成することができます。リモート・イベント・メッセージには、次のイベント・データを含めることができます。

- [デフォルト伝送キュー・タイプ・エラー](#)
- [デフォルト伝送キュー使用エラー](#)
- [キュー・タイプ・エラー](#)
- [リモート・キュー名エラー](#)

- [伝送キュー・タイプ・エラー](#)
- [伝送キュー使用エラー](#)
- [不明デフォルト伝送キュー](#)
- [不明リモート・キュー・マネージャー](#)
- [不明伝送キュー](#)

開始および停止

開始イベントおよび停止イベントは、キュー・マネージャーが開始されたこと、あるいはキュー・マネージャーが停止または静止するように要求されたことを示します。

 z/OS は開始イベントのみをサポートしています。

SYSTEM.ADMIN.QMGR.EVENT キューのデフォルトのメッセージ持続性が持続的と定義されていない限り、停止イベントは記録されません。開始イベント・メッセージおよび停止イベント・メッセージには、次のイベント・データを含めることができます。

- [キュー・マネージャーがアクティブ](#)
- [キュー・マネージャーが非アクティブ](#)

このリストのイベント・タイプごとに、そのイベント・タイプを有効または無効にするキュー・マネージャー属性を設定できます。

チャンネル・イベントとブリッジ・イベント

これらのイベントは、チャンネル操作時に検出された条件の結果として、チャンネルによって報告されます。例えば、チャンネル・インスタンスが停止した場合です。

チャンネル・イベントは、以下の状況で生成されます。

- コマンドによってチャンネルが開始または停止される時。
- チャンネル・インスタンスが開始または停止するとき。
- メッセージの取得時に、チャンネルが変換エラー警告を受信したとき。
- チャンネルを自動的に作成しようとしたとき (この試みが成功しても失敗しても、イベントが生成されません)。

注: クライアント接続によってチャンネル開始イベントまたはチャンネル停止イベントが生成されることはありません。

コマンドを使用してチャンネルを開始すると、イベントが生成されます。チャンネル・インスタンスが開始すると、別のイベントが生成されます。しかし、リスナー **runmqchl** コマンド、またはキュー・マネージャーのトリガー・メッセージによってチャンネルを開始すると、イベントは生成されません。この場合、イベントが生成されるのは、チャンネル・インスタンスが開始するときのみです。

チャンネル開始コマンドまたはチャンネル停止コマンドが正常に実行されると、イベントが少なくとも2つは生成されます。これらのイベントは、チャンネルで接続されている両方のキュー・マネージャーについて生成されます (キュー・マネージャーがイベントをサポートしている場合)。

チャンネル・イベントがイベント・キューに書き込まれると、エラー条件によってキュー・マネージャーはイベントを生成します。

チャンネル・イベントおよびブリッジ・イベントに関するイベント・メッセージは、SYSTEM.ADMIN.CHANNEL.EVENT キューに書き込まれます。

チャンネル・イベント・メッセージには、次のイベント・データを含めることができます。

- [チャンネルアクティブ化](#)
- [チャンネル自動定義エラー](#)
- [チャンネル自動定義 OK](#)
- [チャンネル変換エラー](#)
- [チャンネル非アクティブ化](#)
- [チャンネル開始](#)

- [チャンネル停止](#)
- [ユーザーによるチャンネル停止](#)
- [チャンネルのブロック](#)

IMSブリッジ・イベント (z/OSのみ)

▶ z/OS

このイベントは、IMSブリッジが始動または停止するときにレポートされます。

IMSブリッジ・イベント・メッセージには、次のイベント・データを含めることができます。

- [ブリッジ開始](#)
- [ブリッジ停止](#)

SSL イベント

TLS イベントだけがチャンネル SSL エラー・イベントです。このイベントは、TLS を使用するチャンネルが TLS 接続を確立できなかった場合にレポートされます。

SSL イベント・メッセージには、次のイベント・データを含めることができます：

- [チャンネル SSL エラー](#)
- [チャンネル SSL 警告](#)

パフォーマンス・イベント

パフォーマンス・イベントは、リソースがしきい値条件に達したことの通知です。例えば、キューのサイズが限界に達した場合などです。

パフォーマンス・イベントは、指定したキューを使用するアプリケーションのパフォーマンスに影響のある条件に関連しています。イベント・キュー自身については、それらは生成されません。

イベント・タイプは、メッセージ・データのコマンド ID フィールドに戻ります。

キュー・マネージャーが、キュー・マネージャー・イベント・メッセージまたはパフォーマンス・イベント・メッセージをイベント・キューに書き込もうとしたときに、イベントを通常生成するエラーが検出された場合、別のイベントが生成されることはなく、処置は実行されません。

作業単位がコミットされるか、バックアウトされるかに関係なく、1つの作業単位の MQGET 呼び出しと MQPUT 呼び出しによってパフォーマンス・イベントが生成されることがあります。

パフォーマンス・イベントに関するイベント・メッセージは、SYSTEM.ADMIN.PERFM.EVENT キューに書き込まれます。

パフォーマンス・イベントには、次の2つのタイプがあります。

キュー・サイズ・イベント

キュー・サイズ・イベントは、キュー上のメッセージの数、つまり、キューがどの程度埋まっているか、またはどの程度空になっているかに関連したイベントです。このタイプのイベントは、共有キューに対してサポートされています。キュー・サイズ・イベント・メッセージには、次のイベント・データを含めることができます。

- [キュー・サイズ上限](#)
- [キュー・サイズ下限](#)
- [キュー満杯](#)

キュー・サービス間隔イベント

キュー・サービス間隔イベントは、ユーザーが指定した時間間隔内にメッセージが処理されるかどうかに関連したイベントです。このタイプのイベントは、共有キュー用にサポートされていません。

▶ z/OS

IBM MQ for z/OS は、QSGDISP (SHARED) キューのキュー・サイズ・イベントをサポートしていますが、サービス間隔イベントはサポートしていません。キュー・マネージャーとチャンネル・

イベントには、共有キューの影響は生じません。キュー・サービス・イベント・メッセージには、次のイベント・データを含めることができます。

- キュー・サービス間隔上限
- キュー・サービス間隔OK

構成イベント

構成イベントは、構成イベントが明示的に要求されたときに生成されます。また、オブジェクトが作成、変更、または削除されたときには自動的に生成されます。

構成イベント・メッセージには、オブジェクトの属性に関する情報が含まれます。例えば、名前リスト・オブジェクトが作成されたときに構成イベント・メッセージが生成され、名前リスト・オブジェクトの属性に関する情報がそのメッセージに入れられます。

構成イベントに関するイベント・メッセージは、SYSTEM.ADMIN.CONFIG.EVENT キューに書き込まれます。

構成イベントは、以下のように機能します。

- キュー・マネージャー構成に対する変更の監査証跡は、SYSTEM.ADMIN.CONFIG.EVENT キューに書き込まれるイベント・メッセージによって保持されます。これらのイベントは、ALTER QMGR コマンドの **CONFIGEV** パラメーターを使用して書き込むことができます。
- これらのイベントは、DEFINE、ALTER または DELETE コマンドがオブジェクトに対して実行されたとき、または MQSET 呼び出しが使用されたときに生成されます。
- REFRESH QMGR TYPE(CONFIGEV) コマンドを使用して、現在のキュー・マネージャー構成のベースライン情報を作成でき、このとき、キュー・マネージャーの各オブジェクトに対してイベント・メッセージが1つずつ作成されます。これは時間のかかる操作になる可能性があるため、多数のオブジェクトがある場合は、コマンドで NAME および OBJECT 修飾子を使用してより小さいオブジェクトにタスクを分割できるように注意してください。
- イベント・メッセージでは、次の4つの理由のうち1つを記録します。

- MQRC_CONFIG_CHANGE_OBJECT
- MQRC_CONFIG_CREATE_OBJECT
- MQRC_CONFIG_DELETE_OBJECT
- MQRC_CONFIG_REFRESH_OBJECT

MQRC_CONFIG_CHANGE_OBJECT、MQRC_CONFIG_CREATE_OBJECT、または MQRC_CONFIG_DELETE_OBJECT は、オブジェクトに対して発行できる、該当する MQSC または PCF コマンドで発生します。

MQRC_CONFIG_REFRESH_OBJECT は、ベースライン情報を作成するとき、作成されるイベント・メッセージに対して発生します。

構成イベントには、以下の4つのタイプがあります。

オブジェクトの作成イベント

オブジェクトの作成イベントは、オブジェクトが作成されたときに生成されます。このイベント・メッセージには次のイベント・データが含まれています：オブジェクトの作成。

オブジェクトの変更イベント

オブジェクトの変更イベントは、オブジェクトが変更されたときに生成されます。このイベント・メッセージには次のイベント・データが含まれています：オブジェクトの変更。

オブジェクトの削除イベント

オブジェクトの削除イベントは、オブジェクトが削除されたときに生成されます。このイベント・メッセージには次のイベント・データが含まれています：オブジェクトの削除。

オブジェクトのリフレッシュ・イベント

オブジェクトのリフレッシュ・イベントは、明示的なリフレッシュ要求によって生成されます。このイベント・メッセージには次のイベント・データが含まれています：オブジェクトのリフレッシュ。

コマンド・イベント

MQSC または PCF コマンドが正常に実行されると、コマンド・イベントが報告されます。

コマンド・イベント・メッセージには、コマンドの発行元、コンテキスト、および内容に関する情報が含まれます。例えば、MQSC コマンド ALTER QLOCAL が正常に実行された場合に、それらの情報を持つコマンド・イベント・メッセージが生成されます。

コマンド・イベントに関するイベント・メッセージは、SYSTEM.ADMIN.COMMAND.EVENT キューに書き込まれます。

コマンド・イベントには次のイベント・データが含まれています: [コマンド](#)。

Multi

ローガー・イベント

ローガー・イベントは、リニア・ログを使用するキュー・マネージャーが、新しいログ・エクステン

IBM i

(または IBM i の場合は、新しいジャーナル・レシーバー) に対してログ・レコードの書き込み

を開始したときに報告されます。 [z/OS](#) ローガー・イベントは IBM MQ for z/OS では使用できません。

ローガー・イベント・メッセージには、ログ・エクステン

トを指定する情報が含まれています。ログ・エクステン

トはキュー・マネージャーがキュー・マネージャー再始動またはメディア・リカバリーを行うために必要です。

ローガー・イベントに関するイベント・メッセージは、SYSTEM.ADMIN.LOGGER.EVENT キューに書き込まれます。

このローガー・イベント・メッセージには次のイベント・データが含まれています: [ローガー](#)。

イベント・メッセージ・データの要約

各タイプのイベント・メッセージに含めることができるイベント・データについての情報が必要な場合には、この要約を使用してください。

イベント・タイプ	トピックの参照
権限イベント	非許可 (タイプ 1)
	非許可 (タイプ 2)
	非許可 (タイプ 3)
	非許可 (タイプ 4)
	不許可 (タイプ 5)
	不許可 (タイプ 6)
チャンネル・イベント	チャンネルアクティブ化
	チャンネル自動定義エラー
	チャンネル自動定義 OK
	チャンネルのブロック
	チャンネル変換エラー
	チャンネル非アクティブ化
	チャンネル開始
	チャンネル停止
	ユーザーによるチャンネル停止
コマンド・イベント	コマンド

イベント・タイプ	トピックの参照
構成イベント	オブジェクトの作成
	オブジェクトの変更
	オブジェクトの削除
	オブジェクトのリフレッシュ
IMS ブリッジ・イベント	ブリッジ開始
	ブリッジ停止
禁止イベント	読み取り禁止
	書き込み禁止
ローカル・イベント	別名基本キュー・タイプ・エラー
	不明別名基本キュー
	不明オブジェクト名
ロガー・イベント	ロガー
パフォーマンス・イベント	キュー・サイズ上限
	キュー・サイズ下限
	キュー満杯
	キュー・サービス間隔上限
	キュー・サービス間隔 OK
リモート・イベント	デフォルト伝送キュー・タイプ・エラー
	デフォルト伝送キュー使用エラー
	キュー・タイプ・エラー
	リモート・キュー名エラー
	伝送キュー・タイプ・エラー
	伝送キュー使用エラー
	不明デフォルト伝送キュー
	不明リモート・キュー・マネージャー
	不明伝送キュー
SSL イベント	チャンネル SSL エラー
開始および停止イベント	キュー・マネージャーがアクティブ
	キュー・マネージャーが非アクティブ


イベントの制御

イベントのタイプに応じて、キュー・マネージャー属性またはキュー属性、あるいはその両方に適切な値を指定することにより、イベントを有効にしたり無効にしたりします。

生成したい各観測イベントを有効にする必要があります。例えば、キュー満杯イベントの実行条件は以下のとおりです。

- キュー満杯イベントが、指定されたキューについて有効である。かつ
- アプリケーションが、そのキューにメッセージを書き込むために MQPUT 要求を発行するが、キューが満杯であるため、その要求が失敗する。

以下のいずれかの方法で、イベントを有効または無効にします。

- IBM MQ スクリプト・コマンド (MQSC)。
- 対応する IBM MQ PCF コマンド。
-  z/OS 上のキュー・マネージャーの操作パネルおよび制御パネル。
- IBM MQ Explorer.

注: キューとキュー・マネージャーの両方のイベントに関連する属性は、コマンドによってのみ設定できません。MQI 呼び出し MQSET では、イベント関連の属性をサポートしていません。

関連概念

[10 ページの『観測イベント』](#)

観測イベントは、キュー・マネージャーまたはチャンネル・インスタンスが検出する条件と、イベント・メッセージと呼ばれる特別メッセージをイベント・キューに入れる条件を論理的に組み合わせたものです。

[z/OS での操作および制御パネルの使用](#)

関連タスク

[管理タスクの自動化](#)

[プログラマブル・コマンド・フォーマットの使用](#)

関連資料

[12 ページの『イベント・タイプ』](#)

このページでは、キュー・マネージャーまたはチャンネル・インスタンスが報告できる観測イベントのタイプを示します。

MQSC コマンド

キュー・マネージャー・イベントの制御

キュー・マネージャー属性を使用して、キュー・マネージャー・イベントを制御します。キュー・マネージャー・イベントを有効にするには、該当するキュー・マネージャー属性を **ENABLED** に設定します。キュー・マネージャー・イベントを無効にするには、該当するキュー・マネージャー属性を **DISABLED** に設定します。

キュー・マネージャー・イベントを有効または無効にするには、MQSC コマンド **ALTER QMGR** を使用して、該当するキュー・マネージャー属性を指定します。[20 ページの表 1](#) は、キュー・マネージャー・イベントを有効にする方法を要約したものです。キュー・マネージャー・イベントを無効にするには、該当するパラメーターを **DISABLED** に設定してください。

イベント	ALTER QMGR パラメーター
権限 禁止 ローカル リモート 開始および停止	AUTHOREV (ENABLED) INHIBTEV (ENABLED) LOCALEV (ENABLED) REMOTEEV (ENABLED) STRSTPEV (ENABLED)

チャンネル・イベントおよびブリッジ・イベントの制御

キュー・マネージャー属性を使用して、チャンネル・イベントを制御します。チャンネル・イベントを有効にするには、該当するキュー・マネージャー属性を **ENABLED** に設定します。チャンネル・イベントを無効にするには、該当するキュー・マネージャー属性を **DISABLED** に設定します。

チャンネル・イベントを有効または無効にするには、MQSC コマンド **ALTER QMGR** を使用して、該当するキュー・マネージャー属性を指定します。[21 ページの表 2](#) は、チャンネル・イベントおよびブリッジ・イベントを有効にする方法を要約したものです。キュー・マネージャー・イベントを無効にするには、該当するパラメーターを **DISABLED** に設定してください。

制約事項:  チャンネル自動定義イベントは、IBM MQ for z/OS では使用できません。

表 2. MQSC コマンドによるチャンネル・イベントおよびブリッジ・イベントの有効化

イベント	ALTER QMGR パラメーター
チャンネル チャンネル・エラーのみに関連 IMS ブリッジ SSL チャンネル自動定義	CHLEV (ENABLED) CHLEV (EXCEPTION) BRIDGEEV (ENABLED) SSLEV (ENABLED) CHADEV(ENABLED)

CHLEV が例外に設定された状態では、以下の戻りコードおよび対応する理由修飾子が生成されます。

- MQRC_CHANNEL_ACTIVATED
- MQRC_CHANNEL_CONV_ERROR
- MQRC_CHANNEL_NOT_ACTIVATED
- MQRC_CHANNEL_STOPPED
 - 以下の ReasonQualifier (理由修飾子) が付きます
 - MQRQ_CHANNEL_STOPPED_ERROR
 - MQRQ_CHANNEL_STOPPED_RETRY
 - MQRQ_CHANNEL_STOPPED_DISABLED
- MQRC_CHANNEL_STOPPED_BY_USER
- MQRC_CHANNEL_BLOCKED
 - 以下の ReasonQualifier (理由修飾子) が付きます
 - MQRQ_CHANNEL_BLOCKED_NOACCESS
 - MQRQ_CHANNEL_BLOCKED_USERID
 - MQRQ_CHANNEL_BLOCKED_ADDRESS

パフォーマンス・イベントの制御

PERFMEV キュー・マネージャー属性を使用して、パフォーマンス・イベントを制御します。パフォーマンス・イベントを有効にするには、PERFMEV を ENABLED に設定します。パフォーマンス・イベントを無効にするには、PERFMEV キュー・マネージャー属性を DISABLED に設定します。

PERFMEV キュー・マネージャー属性を ENABLED に設定するには、次の MQSC コマンドを使用します。

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

特定のパフォーマンス・イベントを有効にするには、該当するキュー属性を設定します。そのイベントの実行条件も指定します。

キュー・サイズ・イベント

デフォルトでは、すべてのキュー・サイズ・イベントが無効になります。キュー・サイズ・イベントのいずれかについてキューを構成するには、次の手順を実行する必要があります。

1. キュー・マネージャーのパフォーマンス・イベントを有効にします。
2. 必要なキューのイベントを有効にします。
3. 必要に応じて、最大のキューのサイズに対するパーセンテージで表される該当するレベルに、限度を設定します。

キュー・サービス間隔イベント

キュー・サービス間隔イベントのキューを構成するには、次の手順を実行する必要があります。

1. キュー・マネージャーのパフォーマンス・イベントを有効にします。
2. 必要に応じてキューのキュー・サービス間隔上限または OK イベントについて、制御属性を設定します。

3. キューの QSVICINT 属性を適切な時間に設定して、サービス間隔の時間を指定します。

注: キュー・サービス間隔イベントは、有効になっている場合、適宜生成することができます。そのため、キューの MQI 呼び出しが発行されるのを待つ必要はありません。ただし、メッセージの書き込みまたは削除のためにキューで MQI 呼び出しが使用された場合、該当するパフォーマンス・イベントはその時点で生成されません。経過時間がサービス間隔の時間に等しくなると、そのイベントは生成されません。

構成イベント、コマンド・イベント、およびロガー・イベントの制御

キュー・マネージャー属性 CONFIGEV、CMDEV、および LOGGEREV を使用して、構成イベント、コマンド・イベント、およびロガー・イベントを制御します。これらのイベントを有効にするには、該当するキュー・マネージャー属性を ENABLED に設定します。これらのイベントを無効にするには、該当するキュー・マネージャー属性を DISABLED に設定します。

構成イベント

構成イベントを有効にするには、CONFIGEV を ENABLED に設定します。構成イベントを無効にするには、CONFIGEV を DISABLED に設定します。構成イベントを有効にするには、次のような MQSC コマンドを使用します。

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

コマンド・イベント

コマンド・イベントを有効にするには、CMDEV を ENABLED に設定します。DISPLAY MQSC コマンドおよび PCF コマンドの照会以外のコマンドに対してコマンド・イベントを使用可能にするには、CMDEV を NODISPLAY に設定します。コマンド・イベントを無効にするには、CMDEV を DISABLED に設定します。コマンド・イベントを有効にするには、次のような MQSC コマンドを使用します。

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

ロガー・イベント

ロガー・イベントを有効にするには、LOGGEREV を ENABLED に設定します。ロガー・イベントを無効にするには、LOGGEREV を DISABLED に設定します。ロガー・イベントを有効にするには、次のような MQSC コマンドを使用します。

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

イベント・キュー

イベントが発生すると、キュー・マネージャーはイベント・メッセージを該当するイベント・キューを定義します。イベント・メッセージには、イベントに関する情報が含まれています。

イベント・キューを、以下のいずれかとして定義できます。

- ローカル・キュー
- 別名キュー
- リモート・キューのローカル定義、または
- リモート・クラスター・キュー

イベント・キューすべてを同じリモート・キューのローカル定義として、1つのキュー・マネージャーに定義すると、モニター・アクティビティーを単一地点で統括的に行うことができます。

イベント・メッセージの形式は、伝送キューに必要なメッセージの形式と互換性がないため、イベント・キューは伝送キューとして定義しないでください。

共用イベント・キューとは、QSGDISP(SHARED) 値を使用して定義されたローカル・キューです。

z/OS 上での共有キューの定義について詳しくは、[共有キューを使用したアプリケーション・プログラミング](#)を参照してください。

イベント・キューが使用不可の場合

イベント・キューを使用できないときにイベントが発生すると、イベント・メッセージは失われてしまいます。例えば、あるカテゴリのイベント用にイベント・キューを定義しないと、そのカテゴリのイベント・メッセージがすべて失われます。イベント・メッセージは、送達不能 (未配布メッセージ) キューに保存されるようなことはありません。

ただし、イベント・キューをリモート・キューとして保管することはできます。その場合、解決されたキューにメッセージを書き込むリモート・システムに問題があると、イベント・メッセージはリモート・システムの送達不能キューに到達します。

次のようなさまざまな理由から、イベント・キューを使用できない場合があります。

- キューが定義されていない。
- キューが既に削除されている。
- キューが満杯である。
- キューが書き込み禁止になっている。

イベント・キューがない場合でもイベントは発生します。例えば、パフォーマンス・イベントが発生した後、キュー・マネージャーはキュー属性を変更し、キュー統計をリセットします。この変更は、イベント・メッセージがパフォーマンス・イベント・キューに書き込まれるかどうかにかかわらず発生します。構成イベントおよびコマンド・イベントの場合も同様です。

トリガー・イベント・キューの使用

イベントが生成されると、イベント・メッセージがイベント・キューに書き込まれ、それによってユーザー作成モニター・アプリケーションが開始されるように、トリガー付きのイベント・キューをセットアップすることができます。このアプリケーションはイベント・メッセージを処理し、該当する処置を実行することができます。例えば、オペレーターに通知する必要があるイベントもあれば、一部の管理タスクを自動的に実行するアプリケーションを開始するイベントもあります。

イベント・キューにトリガー処置が関連付けられている場合があります。イベント・キューがトリガー・メッセージを生成する場合があります。ただし、次に、そのトリガー・メッセージがイベントを通常生成するような条件を引き起こす場合でも、イベントは生成されません。このインスタンスでイベントが生成されないため、ループは決して発生しません。

関連概念

[19 ページの『イベントの制御』](#)

イベントのタイプに応じて、キュー・マネージャー属性またはキュー属性、あるいはその両方に適切な値を指定することにより、イベントを有効にしたり無効にしたりします。

[23 ページの『イベント・メッセージの形式』](#)

イベント・メッセージには、イベントおよびその原因についての情報が含まれています。他の IBM MQ メッセージと同様に、イベント・メッセージには、メッセージ記述子とメッセージ・データの 2 つの部分があります。

[共用キューを使用したアプリケーション・プログラミング](#)

[トリガー・イベントの条件](#)

関連資料

[QSGDisp \(MQLONG\)](#)

イベント・メッセージの形式

イベント・メッセージには、イベントおよびその原因についての情報が含まれています。他の IBM MQ メッセージと同様に、イベント・メッセージには、メッセージ記述子とメッセージ・データの 2 つの部分があります。

- メッセージ記述子は MQMD 構造体に基づいています。
- メッセージ・データは、イベント・ヘッダーとイベント・データから構成されます。イベント・ヘッダーには、イベント・タイプを識別する理由コードが設定されています。イベント・メッセージの書き込

みおよびその後行われる処置は、イベントを引き起こした MQI 呼び出しによって戻される理由コードに影響しません。イベント・データは、イベントに関する詳しい情報を提供します。

通常、イベント・メッセージは、実行環境となる企業の要件に合わせて調整されたシステム管理アプリケーションを使用して処理します。

キュー共有グループ内の複数のキュー・マネージャーがイベント・メッセージを生成する条件を検出した場合は、複数のキュー・マネージャーが共有キューに対するイベント・メッセージを生成することになり、その結果、イベント・メッセージが複数になる可能性があります。システムで数種類のキュー・マネージャーからの複数のイベント・メッセージを確実に相関できるようにするために、メッセージ記述子 (MQMD) 中にこれらのイベント・メッセージの固有な相関 ID (*CorrelId*) の集合があります。

関連資料

[102 ページの『アクティビティ・レポート MQMD \(メッセージ記述子\)』](#)

このページを使用して、アクティビティ・レポートの MQMD 構造に含まれた値を確認します。

[106 ページの『アクティビティ・レポート MQEPH \(組み込み PCF ヘッダー\)』](#)

このページを使用して、アクティビティ・レポートの MQEPH 構造に含まれた値を確認します。

[107 ページの『アクティビティ・レポート MQCFH \(PCF ヘッダー\)』](#)

このページを使用して、アクティビティ・レポートの MQCFH 構造に含まれた PCF 値を確認します。

[イベント・メッセージ参照](#)

[イベント・メッセージの形式](#)

[イベント・メッセージ MQMD \(メッセージ記述子\)](#)

[イベント・メッセージ MQCFH \(PCF ヘッダー\)](#)

[イベント・メッセージ記述](#)

パフォーマンス・イベント

パフォーマンス・イベントは、指定したキューを使用するアプリケーションのパフォーマンスに影響のある条件に関連しています。パフォーマンス・イベントの有効範囲はキューです。1つのキューに対する **MQPUT** 呼び出しおよび **MQGET** 呼び出しは、別のキュー上のパフォーマンス・イベントの生成に影響を与えません。

パフォーマンス・イベント・メッセージは適宜生成することができます。そのため、キューの MQI 呼び出しが発行されるのを待つ必要はありません。ただし、メッセージの書き込みまたは削除のためにキューで MQI 呼び出しが使用された場合、該当するパフォーマンス・イベントはその時点で生成されます。

生成される各パフォーマンス・イベント・メッセージは、キュー SYSTEM.ADMIN.PERFM.EVENT に入れます。

イベント・データには、イベントの原因を識別する理由コード、一連のパフォーマンス・イベント統計、およびその他のデータが含まれています。パフォーマンス・イベント・メッセージで返すことのできるイベント・データのタイプを次のリストに示します。

- [キュー・サイズ上限](#)
- [キュー・サイズ下限](#)
- [キュー満杯](#)
- [キュー・サービス間隔上限](#)
- [キュー・サービス間隔 OK](#)

パフォーマンス・イベントの使用法を示す例では、適切な IBM MQ コマンド (MQSC) を使用してキュー属性を設定することを前提としています。z/OS では、キュー属性はキュー・マネージャーの操作パネルと制御パネルを使用して設定することもできます。

関連資料

[12 ページの『イベント・タイプ』](#)

このページでは、キュー・マネージャーまたはチャンネル・インスタンスが報告できる観測イベントのタイプを示します。


パフォーマンス・イベント統計

イベント・メッセージのパフォーマンス・イベント・データには、イベントについての統計が含まれています。統計を使用して、指定したキューの動作を分析できます。

イベント・メッセージのイベント・データには、システム管理プログラムで発生するイベントについての情報が含まれています。いずれのパフォーマンス・イベントの場合も、イベント・データには、イベントに関連したキュー・マネージャーとキューの名前が含まれています。また、イベント・データには、イベントに関連した統計も含まれています。25 ページの表 3 に、キューの動作を分析する場合に使用できるイベント統計をまとめます。すべての統計は、前回統計をリセットしてから発生したことを示します。

パラメーター	説明
TimeSinceReset	統計を最後にリセットしてからの経過時間
HighQDepth	統計を最後にリセットしてからのキュー上のメッセージの最大数
MsgEnqCount	統計を最後にリセットしてからキューに入ったメッセージの数 (キューに対する MQPUT 呼び出しの回数)
MsgDeqCount	統計を最後にリセットしてからキューから取り出されたメッセージの数 (キューに対する MQGET 呼び出しの回数)

パフォーマンス・イベント統計は、次のいずれかが起きた場合にリセットされます。

- パフォーマンス・イベントが発生した場合 (すべてのアクティブ・キュー・マネージャーで、統計がリセットされます)。
- キュー・マネージャーが停止して再始動した場合。
- PCF コマンド `Reset Queue Statistics` がアプリケーション・プログラムから発行された場合。
-  `z/OS` `RESET QSTATS` コマンドがコンソールで発行された場合 (z/OS のみ)。

関連概念

[24 ページの『パフォーマンス・イベント』](#)

パフォーマンス・イベントは、指定したキューを使用するアプリケーションのパフォーマンスに影響のある条件に関連しています。パフォーマンス・イベントの有効範囲はキューです。1つのキューに対する MQPUT 呼び出しおよび MQGET 呼び出しは、別のキュー上のパフォーマンス・イベントの生成に影響を与えません。

[27 ページの『サービス・タイマー』](#)

キュー・サービス間隔イベントは、サービス・タイマーと呼ばれる内部タイマーを使用します。このタイマーは、キュー・マネージャーが制御します。サービス・タイマーが使用されるのは、キュー・サービス間隔イベントが有効になっている場合のみです。

[27 ページの『キュー・サービス間隔イベントの規則』](#)

正式な規則によって、サービス・タイマーを設定してキュー・サービス間隔イベントを生成する時点を制御します。

関連タスク

[28 ページの『キュー・サービス間隔イベントの有効化』](#)

キュー・サービス間隔イベントのキューを構成するには、適切なキュー・マネージャーとキュー属性を設定する必要があります。

関連資料

[キュー・サイズ上限](#)

[Reset Queue Statistics](#)

[RESET QSTATS](#)

キュー・サービス間隔イベント

キュー・サービス間隔イベントは、サービス間隔と呼ばれるユーザー定義の時間間隔内でキューに対して操作が実行されたかどうかを示します。インストール・システムにより、キュー・サービス間隔イベントを使用して、メッセージがキューから取り出される速さをモニターできます。

共有キューの場合は、キュー・サービス間隔イベントはサポートされません。

次のタイプのキュー・サービス間隔イベントが発生する可能性があります。ここで、取得操作という用語は、**MQGET** 呼び出し、または **CLEAR QLOCAL** コマンドの使用など、キューからメッセージを除去するアクティビティを指します。

キュー・サービス間隔 OK

これは、

- MQPUT 呼び出し
- キューを空のままにすることのない取得操作

のいずれかの後に、サービス間隔と呼ばれるユーザー定義の時間間隔内で、取得操作が実行されたことを示します。

取得操作の場合のみ、キュー・サービス間隔 OK イベント・メッセージが生成される可能性があります。キュー・サービス間隔 OK イベントを OK イベントと記述しています。

キュー・サービス間隔上限

これは、

- MQPUT 呼び出し
- キューを空のままにすることのない取得操作

のいずれかの後に、ユーザー定義のサービス間隔内で取得操作が実行されなかったことを示します。

取得操作または MQPUT 呼び出しのいずれかの場合に、キュー・サービス間隔上限イベント・メッセージが生成される可能性があります。キュー・サービス間隔上限イベントを上限イベントと記述しています。

キュー・サービス間隔 OK イベントとキュー・サービス間隔上限イベントの両方を有効にするには、QServiceIntervalEvent 制御属性を High に設定してください。キュー・サービス間隔 OK イベントは、キュー・サービス間隔上限イベントが生成されると自動的に有効になります。キュー・サービス間隔 OK イベントを別個に有効にする必要はありません。

OK および上限イベントは相互に排他的です。つまり、一方が有効になると、他方は無効になります。ただし、両方のイベントを同時に無効にすることはできません。

26 ページの図 3 は、キュー・サイズと時間との関係のグラフです。P1 の時点では、あるアプリケーションが MQPUT を発行して、キューにメッセージを書き込みます。G1 の時点では、別のアプリケーションが MQGET を発行して、キューからメッセージを取り除きます。

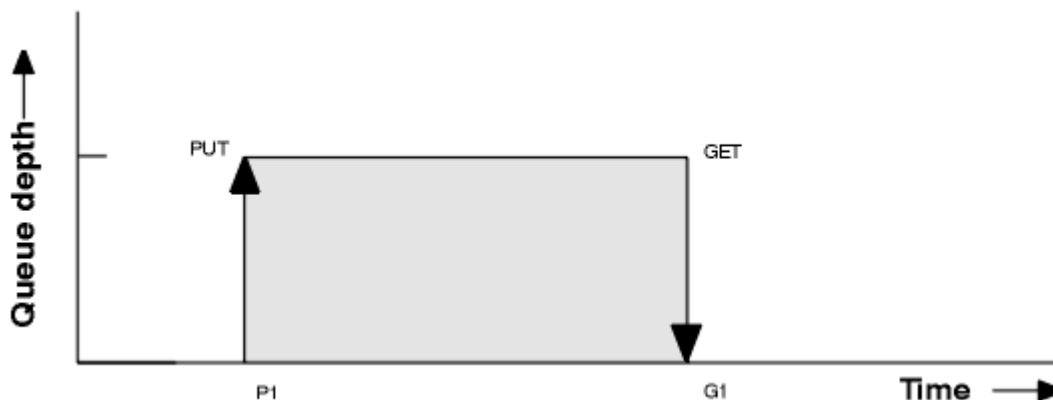


図 3. キュー・サービス間隔イベントについて

キュー・サービス間隔イベントが発生する可能性がある状況は、次のとおりです。

- 書き込みと読み取りの間の経過時間がサービス間隔以下の場合
 - OK イベントが有効になっているときは、キュー・サービス間隔 OK イベントが G1 の時点で生成されます。
- 書き込みと読み取りの間の経過時間がサービス間隔を超える場合
 - キュー・サービス間隔イベントが有効になっているときは、キュー・サービス間隔上限 イベントが G1 の時点で生成されます。

サービス・タイマーを始動し、イベントを生成するためのアルゴリズムについては、[27 ページの『キュー・サービス間隔イベントの規則』](#)で説明します。

関連資料

[キュー・サービス間隔 OK](#)

[キュー・サービス間隔上限](#)

[QServiceIntervalEvent \(MQLONG\)](#)

[QServiceIntervalEvent \(10 桁の符号付き整数\)](#)

サービス・タイマー

キュー・サービス間隔イベントは、サービス・タイマーと呼ばれる内部タイマーを使用します。このタイマーは、キュー・マネージャーが制御します。サービス・タイマーが使用されるのは、キュー・サービス間隔イベントが有効になっている場合のみです。

サービス・タイマーは何を正確に測定するか

サービス・タイマーは、空のキューに対する MQPUT 呼び出しまたは取得操作と、次の書き込みまたは読み取りの間の経過時間を、キューのサイズがこれらの 2 つの操作の間で 0 にならない場合に測定します。

サービス・タイマーはいつアクティブか

キューにメッセージがあり (サイズは 0 ではない)、キュー・サービス間隔イベントが有効になっている場合、サービス・タイマーは常にアクティブ (実行状態) です。キューが空 (キューのサイズが 0) になると、タイマーは OFF の状態になり、次の書き込みで再始動されます。

サービス・タイマーはいつリセットされるか

サービス・タイマーは、取得操作の後で常にリセットされます。空のキューに対する MQPUT 呼び出しによってもリセットされます。しかし、キュー・サービス間隔イベントで必ずリセットされるとは限りません。

サービス・タイマーはどのように使われるか

取得操作または MQPUT 呼び出しの後、キュー・マネージャーは、サービス・タイマーで測定された経過時間をユーザー定義のサービス間隔と比較します。この比較の結果、次のようになります。

- 操作の種類が取得操作であり、経過時間がサービス間隔以下であり、OK イベントが有効になっている場合は、OK イベントが生成されます。
- 経過時間がサービス間隔を超えており、しかも上限イベントが有効になっている場合は、上限イベントが生成されます。

アプリケーションはサービス・タイマーを読み取れるか

読み取れません。サービス・タイマーは、アプリケーションにとっては使用不可の内部タイマーです。

TimeSinceReset パラメーターとは

TimeSinceReset パラメーターは、イベント・データ内のイベント統計の一部として戻ります。このパラメーターは、イベント統計がリセットされない限り、連続する 2 つのキュー・サービス間隔イベント間の時間を指定します。

キュー・サービス間隔イベントの規則

正式な規則によって、サービス・タイマーを設定してキュー・サービス間隔イベントを生成する時点を制御します。

サービス・タイマーの規則

サービス・タイマーは、次の時点で 0 にリセットされて再始動されます。

- 空のキューに対する MQPUT 呼び出しの後
 - MQGET 呼び出しの後でキューが空にならない場合は、MQGET 呼び出しの後
- タイマーのリセットは、イベントが生成されたかどうかに影響されません。
- キュー・サイズが 0 より大きい値なら、キュー・マネージャーの始動時にサービス・タイマーは始動時間に設定されます。
- 取得操作の後でキューが空になると、タイマーは OFF の状態になります。

キュー・サービス間隔上限イベント

キュー・サービス間隔イベントを有効にする (HIGH に設定する) 必要があります。

キュー・サービス間隔上限イベントは、キュー・サービス間隔 OK イベントが生成されると自動的に有効になります。

サービス時間がサービス間隔を超えると、次の MQPUT または取得操作のときに、またはその前にイベントが生成されます。

キュー・サービス間隔 OK イベント

キュー・サービス間隔 OK イベントは、キュー・サービス間隔上限イベントが生成されると自動的に有効になります。

サービス時間 (経過時間) がサービス間隔以下であれば、次の取得操作で、または次の取得操作の前にイベントが生成されます。

関連タスク

28 ページの『[キュー・サービス間隔イベントの有効化](#)』

キュー・サービス間隔イベントのキューを構成するには、適切なキュー・マネージャーとキュー属性を設定する必要があります。

キュー・サービス間隔イベントの有効化

キュー・サービス間隔イベントのキューを構成するには、適切なキュー・マネージャーとキュー属性を設定する必要があります。

このタスクについて

上限イベントと OK イベントは、相互に排他的です。つまり、一方が有効になると、他方は自動的に無効になります。

- 上限イベントがあるキューに関して生成されると、キュー・マネージャーは、自動的にそのキューに関して上限イベントを無効にし、OK イベントを有効にします。
- OK イベントがあるキューに関して生成されると、キュー・マネージャーは、自動的にそのキューに関して OK イベントを無効にし、OK イベントを有効にします。

キュー・サービス間隔イベント	キューの属性
キュー・サービス間隔上限 キュー・サービス間隔 OK キュー・サービス間隔イベントなし	QSVCI EV (HIGH) QSVCI EV (OK) QSVCI EV (NONE)
サービス・インターバル	QSVCI NT (<i>tt</i>) (<i>tt</i> はサービス) ミリ秒単位のインターバル時間。

キュー・サービス間隔イベントを有効にするには、以下のステップを実行します。

手順

1. キュー・マネージャー属性 **PERFMEV** を **ENABLED** に設定します。
キュー・マネージャーでパフォーマンス・イベントが有効になります。
2. 必要に応じてキューのキュー・サービス間隔上限イベントまたは **OK** イベントについて、制御属性 **QSVCI EV** を設定します。
3. キューの **QSVCI NT** 属性を設定して、適切なサービス間隔時間を指定します。

例

サービス間隔の時間を 10 秒 (10 000 ミリ秒) に設定してキュー・サービス間隔上限イベントを有効にするには、次の MQSC コマンドを使用します。

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QSVCI NT(10000) QSVCI EV(HIGH)
```

キュー・サービス間隔イベントの例

このセクションの例を使用すると、キュー・サービス間隔イベントから得られる情報について理解できます。

次の 3 つのサブピックの例では、徐々に複雑な図を示して、キュー・サービス間隔イベントの使用法を説明します。

各サブピックの例に示されている図は、次のような同じ構造になっています。

- 図 1 はキューのサイズと時間の関係を示したグラフであり、個々の MQGET 呼び出しと MQPUT 呼び出しが示されています。
- 説明のセクションには、時間の制約に関する比較が示されています。考慮する時間枠として、次の 3 種類があります。
 - ユーザー定義のサービス間隔
 - サービス・タイマーで測定される時間
 - イベント統計の前のリセット以降の時間 (イベント・データの TimeSinceReset)
- イベント統計の要約セクションには、どのイベントがどの時点で有効になるか、またどのようなイベントが生成されるかが示されています。

これらの例では、キュー・サービス間隔イベントの次の面について説明します。

- 時間が経過するにつれてキュー・サイズはどのように変化するか。
- サービス・タイマーで測定された経過時間は、サービス間隔と比較してどのように異なるか。
- どのイベントが有効か。
- どのイベントが生成されるか。

要確認: 例 1 には、メッセージが断続的に発生し、各メッセージは、次のメッセージが着信する前にキューから取り出される単純な例を示します。キュー上のメッセージの最大数は 1 であったことが、イベント・データからわかります。したがって、各メッセージがキュー上に存在していた時間を算出できます。

しかし、一般的な場合では、複数のメッセージがキューにあり、MQGET 呼び出しと MQPUT 呼び出しの順序は予測可能ではありません。したがって、キュー・サービス間隔イベントを使用して、個々のメッセージがキュー上に存在する時間を算出することはできません。イベント・データで返される TimeSinceReset パラメーターの時間には、メッセージがまったくキューにない時間が含まれることがあります。したがって、この統計値から導いた時間を平均すると、必ずこのようなメッセージなしの時間が含まれています。

関連概念

26 ページの『キュー・サービス間隔イベント』

キュー・サービス間隔イベントは、サービス間隔と呼ばれるユーザー定義の時間間隔内でキューに対して操作が実行されたかどうかを示します。インストール・システムにより、キュー・サービス間隔イベントを使用して、メッセージがキューから取り出される速さをモニターできます。

27 ページの『サービス・タイマー』

キュー・サービス間隔イベントは、サービス・タイマーと呼ばれる内部タイマーを使用します。このタイマーは、キュー・マネージャーが制御します。サービス・タイマーが使用されるのは、キュー・サービス間隔イベントが有効になっている場合のみです。

キュー・サービス間隔イベント - 例 1

キュー・サイズが常に 1 または 0 になる、MQGET 呼び出しと MQPUT 呼び出しの基本的な順序。

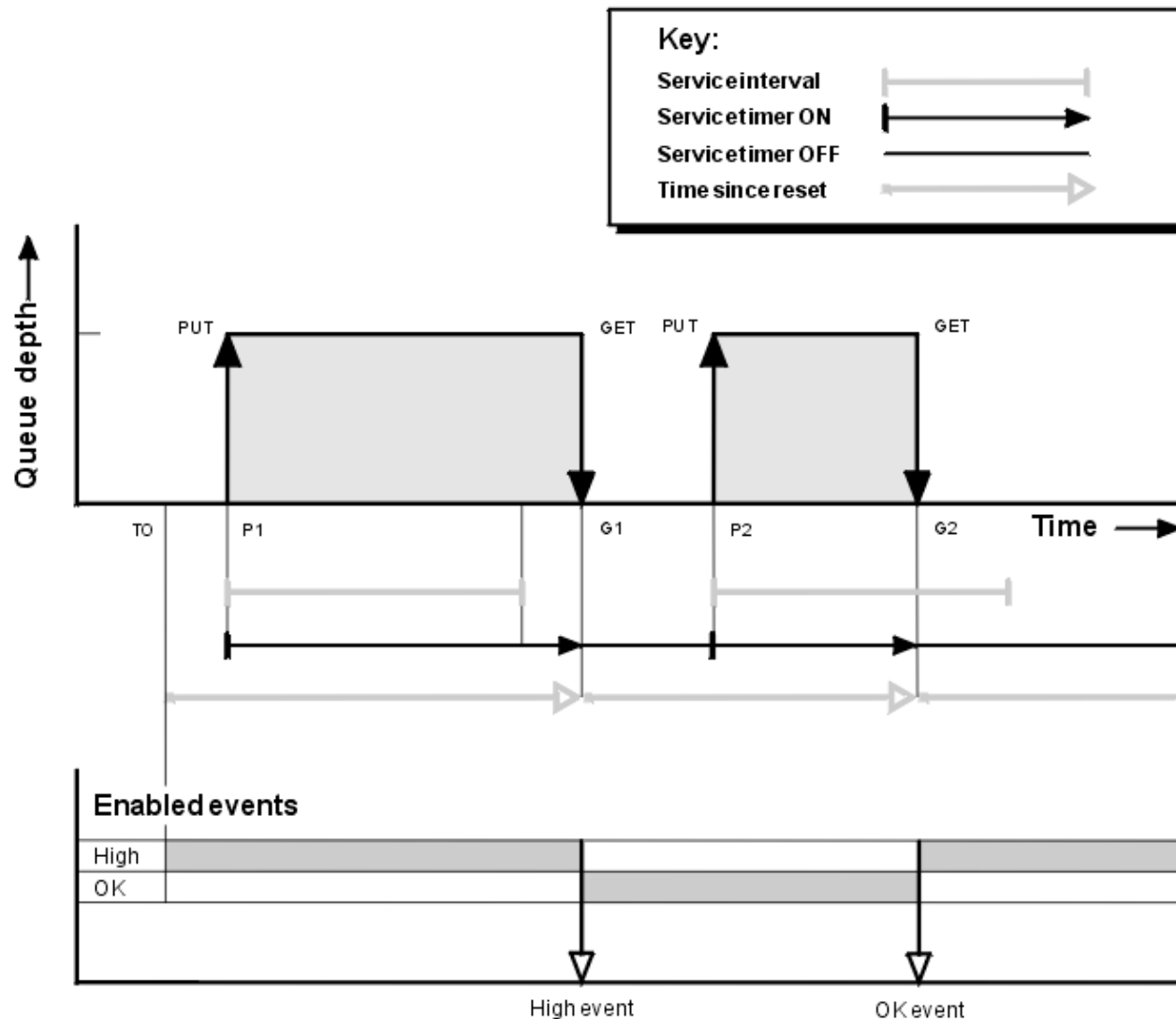


図 4. キュー・サービス間隔イベント - 例 1

解説

1. P1 の時点でアプリケーションは、メッセージを空のキューに書き込みます。これによってサービス・タイマーが始動します。

場合によっては T0 がキュー・マネージャーの始動時点であることに注目してください。

2. G1 で、別のアプリケーションがメッセージをキューから取り出します。P1 から G1 までの経過時間がサービス間隔を超えているので、G1 での MQGET 呼び出しでキュー・サービス間隔上限イベントが生成

されます。上限イベントが生成されると、キュー・マネージャーはイベント制御属性をリセットし、その結果、次のようになります。

- a. OK イベントが自動的に有効になります。
- b. 上限イベントが無効になります。

この時点でキューが空になったため、サービス・タイマーは OFF の状態に切り替わります。

- 3. P2 で 2 番目のメッセージがキューに書き込まれます。これによってサービス・タイマーが再始動されます。
- 4. G2 の時点で、メッセージがキューから取り出されます。ただし、P2 から G2 までの経過時間はサービス間隔より短いので、G2 での MQGET 呼び出しではキュー・サービス間隔 OK イベントが生成されません。OK イベントが生成されると、キュー・マネージャーは制御属性をリセットし、その結果、次のようになります。
 - a. 上限イベントが自動的に有効になります。
 - b. OK イベントが無効になります。

キューが空になったため、サービス・タイマーは、再び OFF の状態に切り替わります。

イベント統計の要約

31 ページの表 5 は、この例のイベント統計の要約です。

属性	イベント 1	イベント 2
イベントの時点	T(G1)	T(G2)
イベントのタイプ	高	OK
TimeSinceReset	T(G1) - T(0)	T(G2) - T(G1)
HighQDepth	1	1
MsgEnqCount	1	1
MsgDeqCount	1	1

30 ページの図 4 の中央部分には、サービス・タイマーで測定された経過時間がそのキューのサービス間隔と対比して示されています。キュー・サービス間隔イベントが発生するかどうかは、サービス・タイマーを表す横方向の線 (矢印が付いている) の長さ和服务間隔を表す線の長さを比較するとわかります。サービス・タイマーの線の方が長く、キュー・サービス間隔上限イベントが有効の場合、次の読み取りでキュー・サービス間隔上限イベントが発生します。サービス・タイマーの線のほうが短く、キュー・サービス間隔 OK イベントが有効の場合は、次の読み取りでキュー・サービス間隔 OK イベントが発生します。

キュー・サービス間隔イベント - 例 2

キューのサイズが必ずしも 1 または 0 になるとは限らない一連の MQPUT 呼び出しと MQGET 呼び出し。

この例は、P2 の時点でイベントが生成されずにタイマーがリセットされる例も示しています。

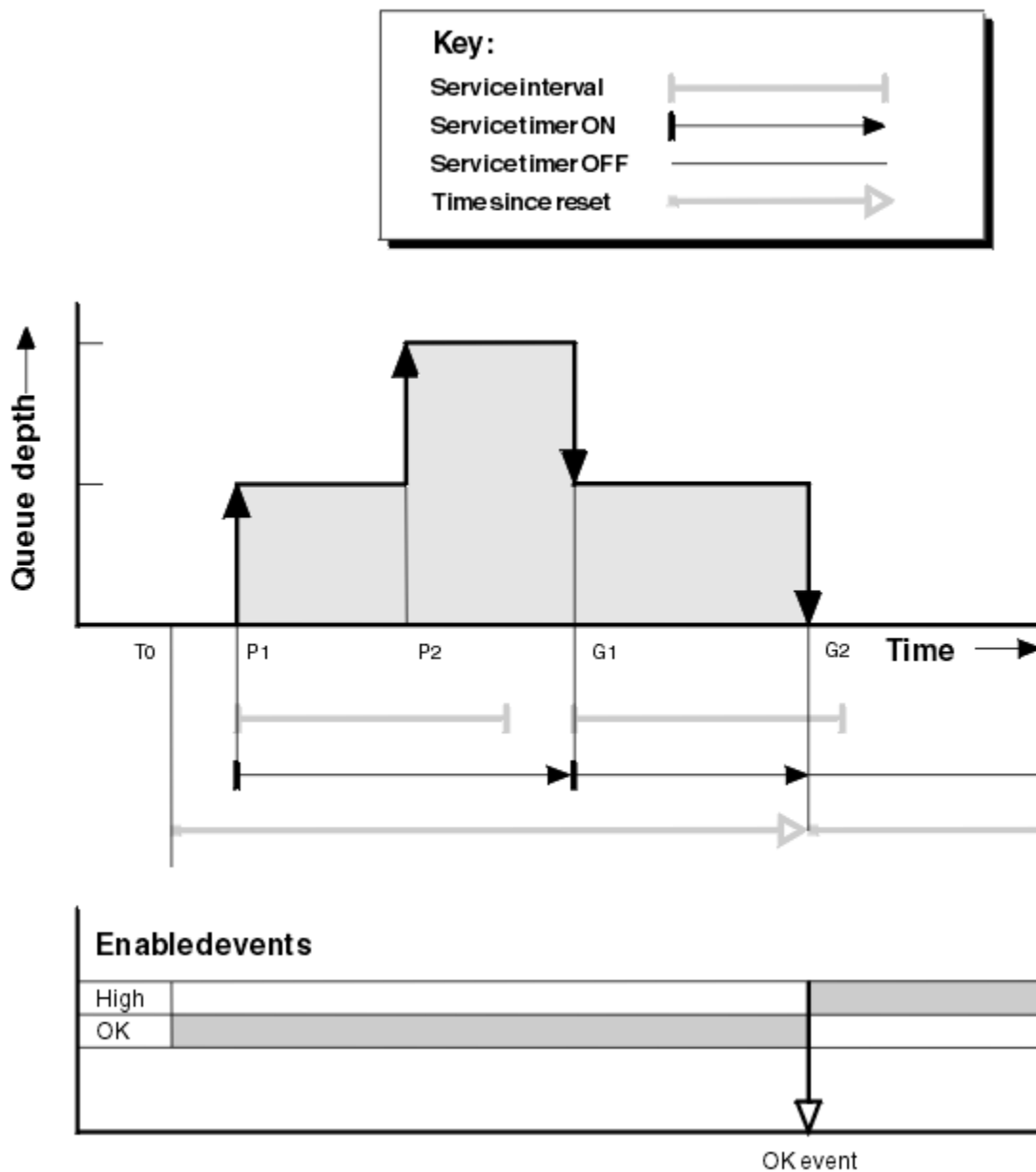


図 5. キュー・サービス間隔イベント - 例 2

解説

この例では、最初に OK イベントが有効であり、キュー統計が T0 の時点でリセットされています。

1. P1 では、最初の書き込みによってサービス・タイマーが始動します。
2. P2 では、2 番目の書き込みがイベントを生成することはありません。書き込みは OK イベントを引き起こすことができないためです。
3. G1 では、サービス間隔を既に超えているので、OK イベントは生成されません。ただし、MQGET 呼び出しによって、サービス・タイマーがリセットされます。
4. G2 では、2 番目の読み取りがサービス間隔内に発生し、今回は OK イベントが生成されます。キュー・マネージャーはイベント制御属性をリセットし、その結果、次のようになります。
 - a. 上限イベントが自動的に有効になります。
 - b. OK イベントが無効になります。

この時点でキューが空になったため、サービス・タイマーは OFF の状態に切り替わります。

イベント統計の要約

33 ページの表 6 は、この例のイベント統計の要約です。

表 6. 例 2 のイベント統計の要約	
属性	イベント 2
イベントの時点	T(G2)
イベントのタイプ	OK
TimeSinceReset	T(G2) - T(0)
HighQDepth	2
MsgEnqCount	2
MsgDeqCount	2

キュー・サービス間隔イベント - 例 3

一連の MQGET 呼び出しと MQPUT 呼び出しが、直前の例に比べて散発的に発生する例。

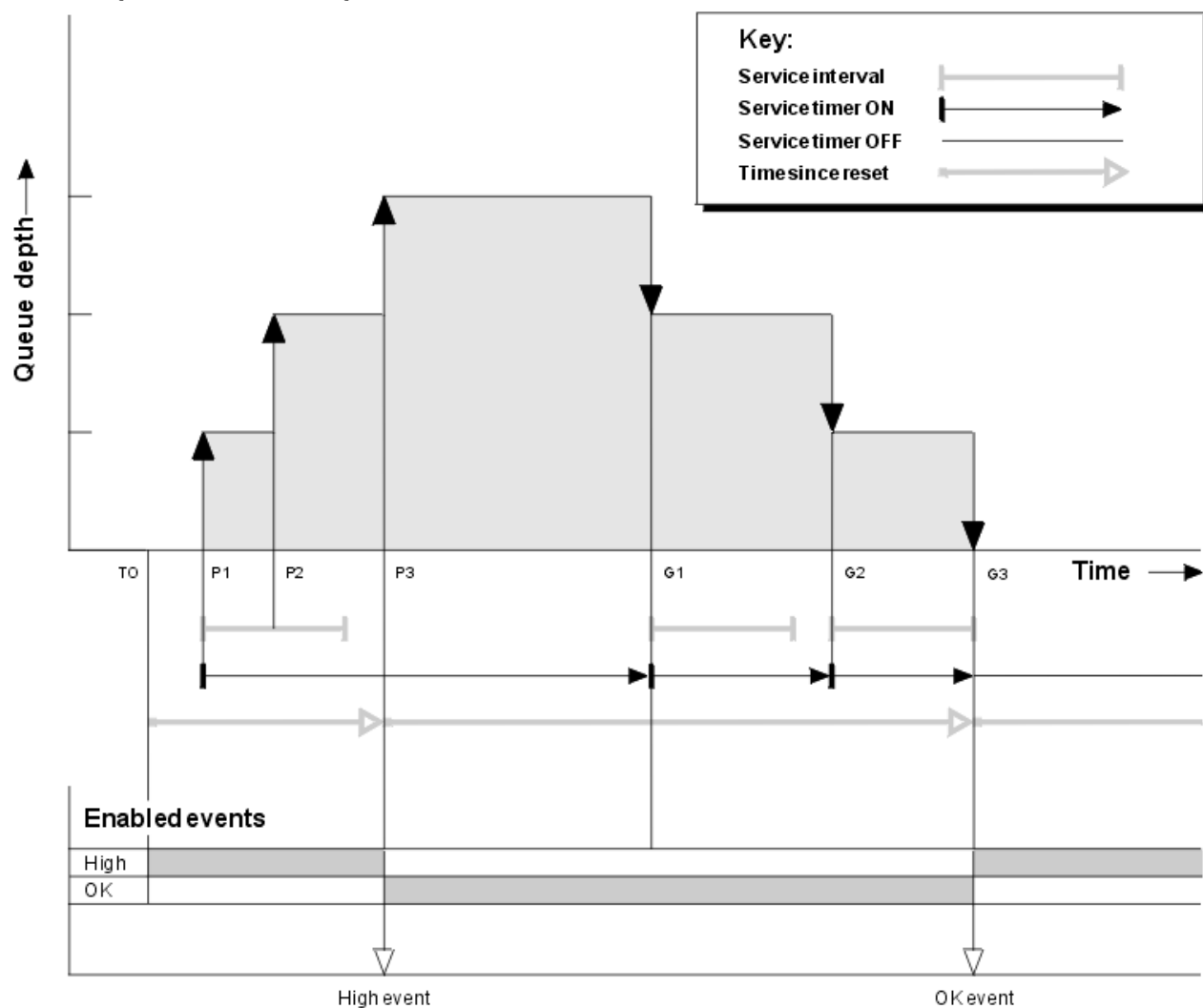


図 6. キュー・サービス間隔イベント - 例 3

解説

1. T(0) の時点で、キュー統計がリセットされ、キュー・サービス間隔上限イベントが有効になります。
2. P1 では、最初の書き込みによってサービス・タイマーが始動します。
3. P2 では、2 番目の書き込みによってキュー・サイズが 2 になります。この時点ではサービス間隔の時間をまだ超えていないので、上限イベントは生成されません。
4. P3 では、3 番目の書き込みによって上限イベントが生成されます (タイマーの時間は、既にサービス間隔を超えています)。キュー・サイズがこの書き込みの前に 0 になっていなかったため、タイマーはリセットされません。しかし、OK イベントが有効になります。
5. G1 では、サービス間隔を既に超えており、OK イベントが有効になっているため、MQGET 呼び出しはイベントを生成しません。ただし、MQGET 呼び出しは、サービス・タイマーをリセットします。
6. G2 では、サービス間隔を既に超えており、OK イベントが有効になっているため、MQGET 呼び出しはイベントを生成しません。MQGET 呼び出しは、再度サービス・タイマーをリセットします。
7. G3 では、3 番目の読み取りによってキューは空になり、サービス・タイマーの時間はサービス間隔に等しくなります。したがって、OK イベントが生成されます。サービス・タイマーはリセットされ、上限イベントが有効になります。MQGET 呼び出しにより、キューが空になり、その結果タイマーは OFF の状態になります。

イベント統計の要約

34 ページの表 7 は、この例のイベント統計の要約です。

属性	イベント 1	イベント 2
イベントの時点	T(P3)	T(G3)
イベントのタイプ	高	OK
TimeSinceReset	T(P3) - T(0)	T(G3) - T(P3)
HighQDepth	3	3
MsgEnqCount	3	0
MsgDeqCount	0	3

キュー・サイズ・イベント

キュー・サイズ・イベントとは、キューのサイズ、つまり、キューのメッセージの数に関連したイベントです。

IBM MQ アプリケーションでは、キューが満杯になってはなりません。満杯になると、アプリケーションはそれ自体が指定したキューに、メッセージを書き込めなくなります。このような場合でもメッセージは失われませんが、キューが満杯になるために問題が発生することがあります。メッセージを処理するアプリケーションによって、メッセージが取り出される速度よりも高速にメッセージがキューに書き込まれると、キュー上のメッセージの数が増えることがあります。

この問題の解決策は、個々の状況によって異なりますが、次のような方法があります。

- 一部のメッセージを別のキューに方向転換する。
- より多くのメッセージをキューから取り出すために、新しいアプリケーションを始動する。
- 重要でないメッセージの送受信を停止する。
- 瞬間的ピークに対応できるよう、キューのサイズを増加する。

問題が起きかかっているという警告を事前に受け取ると、予防的処置を取りやすくなります。このため、IBM MQ では次のキュー・サイズ・イベントが用意されています。

キュー・サイズ上限イベント

キューのサイズが増加して、キュー・サイズ上限と呼ばれる事前定義のしきい値にまで既に達していることを示します。

キュー・サイズ下限イベント

キューのサイズが減少して、キュー・サイズ下限と呼ばれる事前定義のしきい値にまで既に達していることを示します。

キュー満杯イベント

キューがその最大サイズに到達した、つまり、キューが満杯であることを示します。

最大サイズに到達したキューに、アプリケーションがメッセージを書き込もうとすると、キュー満杯イベントが生成されます。キュー・サイズ上限イベントは、キューが満杯になりかかっていることを事前に警告するものです。つまり、このイベントを受信したとき、システム管理者は、なんらかの予防的処置を取る必要があります。予防的処置が成功して、キュー・サイズが安全なレベルにまで減少したら、キュー・サイズ下限イベントを生成するように、キュー・マネージャーを構成できます。

最初のキュー・サイズ・イベントの例で、キューを満杯にしないようにする予防的処置の効果を示します。

関連概念

38 ページの『[キュー・サイズ・イベントの例](#)』

これらの例を使用して、キュー・サイズ・イベントから入手できる情報を理解します。

関連資料

[キュー満杯](#)

[キュー・サイズ上限](#)



[キュー・サイズ下限](#)

キュー・サイズ・イベントの有効化

キュー・サイズ・イベントのキューを構成するには、適切なキュー・マネージャーとキュー属性を設定します。

このタスクについて

デフォルトでは、すべてのキュー・サイズ・イベントが無効になります。有効である場合、キュー・サイズ・イベントは次のように生成されます。

- メッセージがキューに書き込まれた結果として、**QDepthHighLimit** に設定された値以上のキュー・サイズになったとき、キュー・サイズ上限イベントが生成されます。
 - キュー・サイズ上限イベントは、同じキューのキュー・サイズ下限イベントによって自動的に有効になります。
 - キュー・サイズ上限イベントは、同じキューのキュー・サイズ下限イベントとキュー満杯イベントの両方を自動的に有効にします。
-  GET 操作でメッセージがキューから取り出された結果、**QDepthLowLimit** に設定された値以下のキュー・サイズになったとき、キュー・サイズ下限イベントが生成されます。
 -  GET 操作でメッセージがキューから取り出された結果、または取り出される予定だったが期限切れとなっている結果、**QDepthLowLimit** に設定された値以下のキュー・サイズになったとき、キュー・サイズ下限イベントが生成されます。
 - キュー・サイズ下限イベントは、同じキューのキュー・サイズ上限イベントまたはキュー満杯イベントによって自動的に有効になります。
 - キュー・サイズ下限イベントは、同じキューのキュー・サイズ上限イベントとキュー満杯イベントの両方を自動的に有効にします。
- キューが満杯であるためにアプリケーションがメッセージをキューに書き込むことができないとき、キュー満杯イベントが生成されます。
 - キュー満杯イベントは、同じキューのキュー・サイズ上限イベントまたはキュー・サイズ下限イベントによって自動的に有効になります。
 - キュー満杯イベントは、同じキューのキュー・サイズ下限イベントを自動的に有効にします。

キュー・サイズ・イベントのいずれかについてキューを構成するには、次の手順を実行する必要があります。

手順

1. キュー・マネージャー属性 **PERFMEV** を使用して、キュー・マネージャーのパフォーマンス・イベントを有効にします。
イベントは `SYSTEM.ADMIN.PERFM.EVENT` キューに入ります。
2. 必要なキューに関するイベントを有効にするには、次のいずれかの属性を設定します。
 - `QDepthHighEvent` (MQSC の **QDPHIEV**)
 - `QDepthLowEvent` (MQSC の **QDPLOEV**)
 - `QDepthMaxEvent` (MQSC の **QDPMAXEV**)
3. オプション: 限度を設定するには、最大キュー・サイズに対する割合 (%) として、次の属性を割り当てます。
 - `QDepthHighLimit` (MQSC の **QDEPTHHI**)
 - `QDepthLowLimit` (MQSC の **QDEPTHLO**)

制約事項: **QDEPTHHI** は **QDEPTHLO** 以上にする必要があります。

QDEPTHHI が **QDEPTHLO** に等しいと、キュー・サイズが値を下回るまたは上回るたびにイベント・メッセージが生成されます。これは、キュー・サイズが値を下回ると上限しきい値が有効になり、キュー・サイズが値を上回ると下限しきい値が有効になるためです。

タスクの結果

注:

Multi **QDepthLowLimit** に設定された値以下のキュー・サイズになる GET 操作で、有効期限が切れたメッセージがキューから取り出されるとき、キュー・サイズ下限イベントは生成されません。IBM MQ は、正常な GET 操作の間のみ、キュー・サイズ下限イベント・メッセージを生成します。そのため、有効期限が切れたメッセージがキューから取り出されるとき、キュー・サイズ下限イベント・メッセージは生成されません。また、これらの有効期限が切れたメッセージをキューから取り出した後、`QDepthHighEvent` および `QDepthLowEvent` はリセットされません。

z/OS IBM MQ は、正常に終了した破壊 GET 操作中、または正常に終了する予定だったが一致するメッセージが期限切れにならない破壊 GET 操作中に、キュー・サイズ下限イベント・メッセージを生成します。それ以外の場合は、汎用バックグラウンド・プロセスの際に有効期限が切れたメッセージがキューから取り出されるとき、キュー・サイズ下限イベント・メッセージは生成されません。また、汎用バックグラウンド・プロセスの際に有効期限が切れたメッセージをキューから取り出した後、`QDepthHighEvent` および `QDepthLowEvent` はリセットされません。期限切れメッセージ処理の詳細については、[IBM MQ for z/OS](#) でのキュー・マネージャー調整を参照してください。

例

キュー MYQUEUE の限度を 80% に設定してキュー・サイズ上限イベントを有効にするには、次の MQSC コマンドを使用します。

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

キュー MYQUEUE の限度を 20% に設定してキュー・サイズ下限イベントを有効にするには、次の MQSC コマンドを使用します。

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

キュー MYQUEUE に関してキュー満杯イベントを有効にするには、次の MQSC コマンドを使用します。


```
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* (**PERFMVEV** 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 **PERFMVEV** on the queue manager is ENABLED. This means that if, for example, the queue manager's **PERFMVEV** 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.

キュー・サイズ・イベントの例

これらの例を使用して、キュー・サイズ・イベントから入手できる情報を理解します。

最初の例では、キュー・サイズ・イベントの基本的な図を示します。2つ目の例はより長くなりますが、原則は最初の例と同じです。どちらの例も、次のような同じキュー定義を使用します。

キュー MYQUEUE1 の最大サイズはメッセージ数 1000 です。キュー・サイズ上限は 80%、キュー・サイズ下限は 20% です。最初は、キュー・サイズ上限イベントが有効になっており、その他のキュー・サイズ・イベントは無効になっています。

このキューを設定するための IBM MQ コマンド (MQSC) は次のとおりです。

```
ALTER QMGR PERFMVEV(ENABLED)

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

関連概念

[34 ページの『キュー・サイズ・イベント』](#)

キュー・サイズ・イベントとは、キューのサイズ、つまり、キューのメッセージの数に関連したイベントです。

関連タスク

[35 ページの『キュー・サイズ・イベントの有効化』](#)

キュー・サイズ・イベントのキューを構成するには、適切なキュー・マネージャーとキュー属性を設定します。

関連資料

[MQSC コマンド](#)

[キュー・サイズ・イベント - 例 1](#)

[キュー・サイズ・イベントの基本的な順序](#)

[39 ページの図 7](#) は、時間の経過とキュー・サイズの変化を示しています。

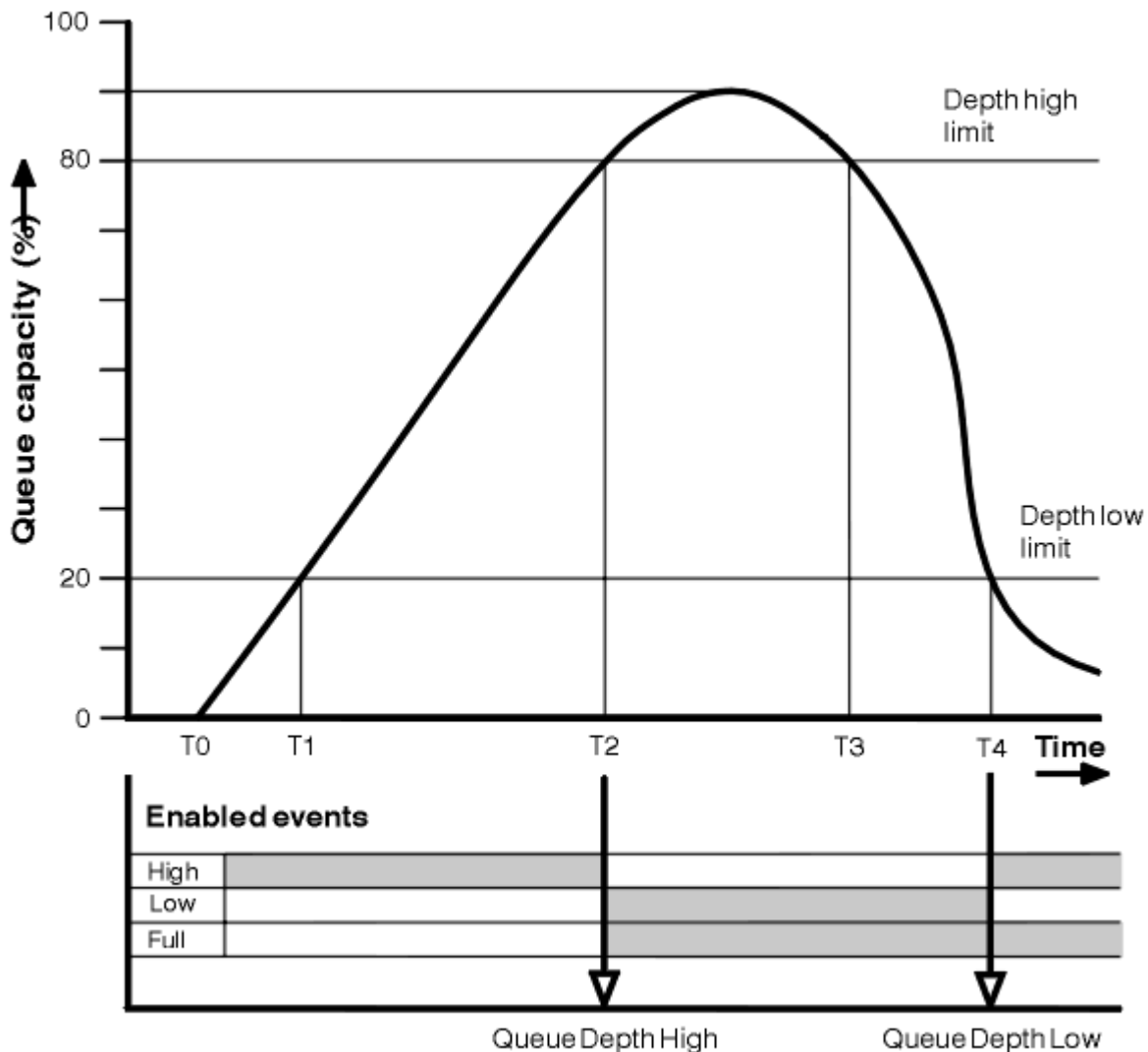


図 7. キュー・サイズ・イベント (1)

解説

1. T(1) の時点では、キュー・サイズは増加中であり (MQGET 呼び出しよりも MQPUT 呼び出しの方が多
い)、キュー・サイズ下限と交差しています。この時点では、イベントは生成されません。
2. キュー・サイズは T(2) まで増加し続け、サイズの上限 (80%) に達するとキュー・サイズ上限イベント
が生成されます。
これによって、キュー満杯イベントとキュー・サイズ下限イベントの両方が有効になります。
3. イベントによる影響のために取った予防的処置により、キューが満杯になることを防止します。T(3) の
時点で、再びキュー・サイズ上限に達します。この時点では、イベントは生成されません。
4. キュー・サイズは T(4) まで減少し続け、サイズ下限 (20%) に達するとキュー・サイズ下限イベントが
生成されます。
これによって、キュー満杯イベントとキュー・サイズ上限イベントの両方が有効になります。

イベント統計の要約

40 ページの表 8 は、キュー・イベント統計の要約であり、40 ページの表 9 は、どのイベントが有効かを要約したものです。

表 8. キュー・サイズ・イベントのイベント統計の要約 (例 1)

イベント統計	イベント 2	イベント 4
イベントの時点	T(2)	T(4)
イベントのタイプ	キュー・サイズ上限	キュー・サイズ下限
TimeSinceReset	T(2) - T(0)	T(4) - T(2)
HighQDepth (リセット以降の最大キュー・サイズ)	800	900
MsgEnqCount	1157	1220
MsgDeqCount	357	1820

表 9. 有効なイベントに関する要約

時間枠	キュー・サイズ上限イベント	キュー・サイズ下限イベント	キュー満杯イベント
T(1) 以前	ENABLED	-	-
T(1) から T(2)	ENABLED	-	-
T(2) から T(3)	-	ENABLED	ENABLED
T(3) から T(4)	-	ENABLED	ENABLED
T(4) 以降	ENABLED	-	ENABLED

キュー・サイズ・イベント - 例 2

キュー・サイズ・イベントのさらに長い順序

41 ページの図 8 は、時間の経過とキュー・サイズの変化を示しています。

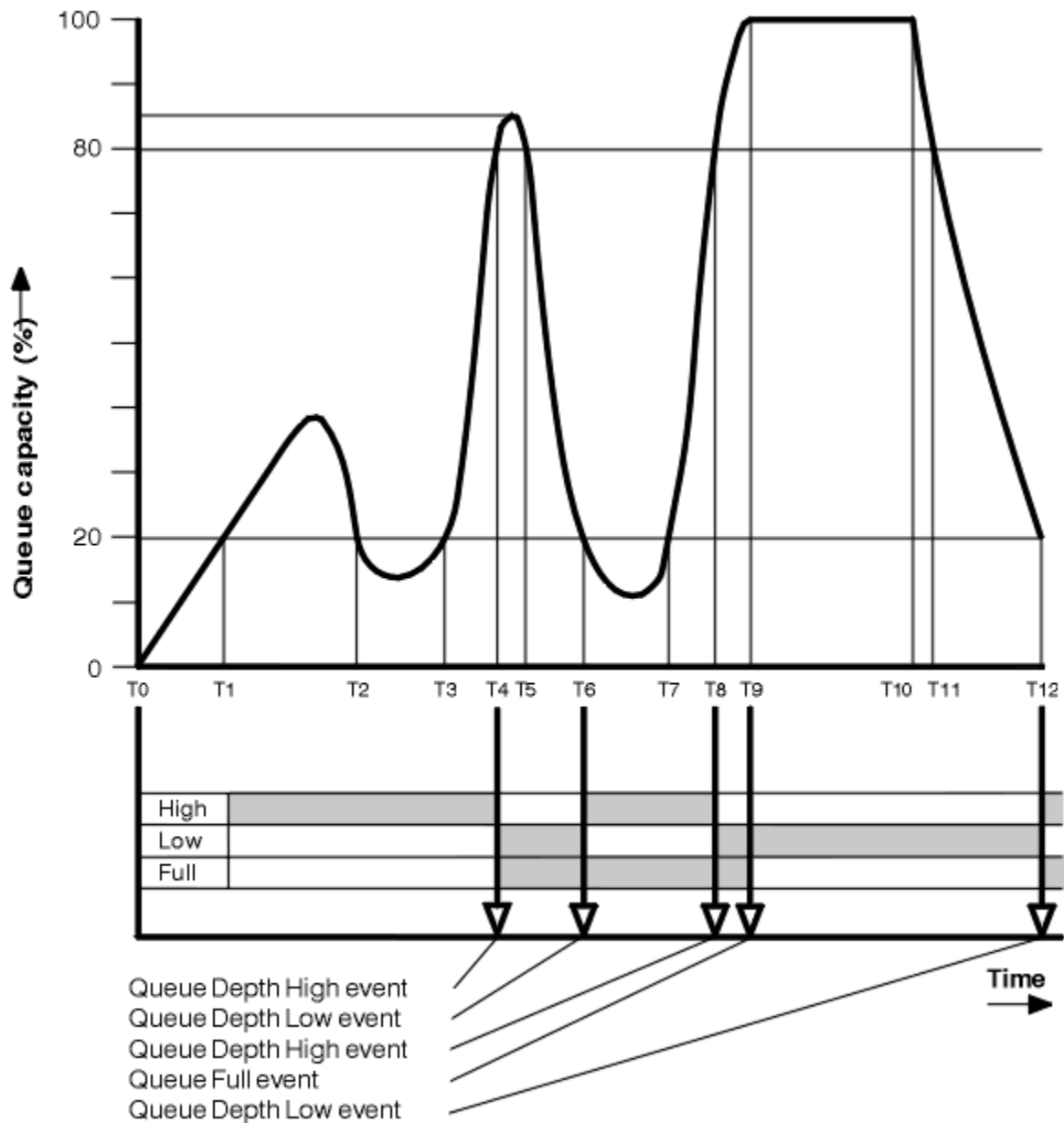


図 8. キュー・サイズ・イベント (2)

解説

1. キュー・サイズ下限イベントは、次の時点では生成されません。
 - T(1) (キュー・サイズが増加中で、イベントは無効)
 - T(2) (イベントは無効)
 - T(3) (キュー・サイズが増加中で、イベントは無効)
2. T(4) で、キュー・サイズ上限イベントが発生します。これによって、キュー満杯イベントとキュー・サイズ下限イベントの両方が有効になります。
3. T(9) では、キューが満杯であるためにキューに書き込むことができない最初のメッセージの後、キュー満杯イベントが発生します。
4. T(12) では、キュー・サイズ下限イベントが発生します。

イベント統計の要約

42 ページの表 10 はキュー・イベント統計を要約したものであり、42 ページの表 11 はこの例のさまざまな時点でどのイベントが有効かを要約したものです。

イベント統計	イベント 4	イベント 6	イベント 8	イベント 9	イベント 12
イベントの時点	T(4)	T(6)	T(8)	T(9)	T(12)
イベントのタイプ	キュー・サイズ上限	キュー・サイズ下限	キュー・サイズ上限	キュー満杯	キュー・サイズ下限
TimeSinceReset	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

時間枠	キュー・サイズ上限イベント	キュー・サイズ下限イベント	キュー満杯イベント
T(0) から T(4)	ENABLED	-	-
T(4) から T(6)	-	ENABLED	ENABLED
T(6) から T(8)	ENABLED	-	ENABLED
T(8) から T(9)	-	ENABLED	ENABLED
T(9) から T(12)	-	ENABLED	-
T(12) 以降	ENABLED	-	ENABLED

注：イベントは同期点外にあります。したがって、キューが空になった後、そのキューを満杯にしてイベントを引き起こし、その後、同期点管理プログラムの制御下ですべてのメッセージを取り消すことができます。ただし、イベントの有効化は自動的に設定されているので、次回キューが満杯になったとき、イベントは生成されません。

構成イベント

構成イベントとは、オブジェクトが作成、変更、または削除されたときに生成される通知であり、明示的な要求によって生成されることもあります。

イベントをオンにする方法については、22 ページの『[構成イベント、コマンド・イベント、およびロガー・イベントの制御](#)』を参照してください。

構成イベントは、オブジェクトの属性の変更を通知します。構成イベントには、次の 4 つのタイプがあります。

- オブジェクトの作成イベント
- オブジェクトの変更イベント
- オブジェクトの削除イベント
- オブジェクトのリフレッシュ・イベント

イベント・データには、以下の情報が含まれています。

発生元情報

変更が行われたキュー・マネージャー、変更を行ったユーザーの ID、および行われた変更方法 (コンソール・コマンドなどによる) について説明します。

コンテキスト情報

コマンド・メッセージからのメッセージ・データ内のコンテキスト情報のレプリカです。

コマンドが SYSTEM.COMMAND.INPUT キューにメッセージとして入力された場合、コンテキスト情報はイベント・データ内にのみ含まれます。

オブジェクト識別

オブジェクトの名前、タイプ、およびファイル属性指定について説明します。

オブジェクトの属性

オブジェクト内のすべての属性の値について説明します。

イベントは、PCF 形式のメッセージです。詳細については、以下を参照してください。

- [オブジェクトの変更](#)
- [オブジェクトの作成](#)
- [オブジェクトの削除](#)
- [オブジェクトのリフレッシュ](#)

オブジェクト変更イベントの場合は、変更前と変更後のメッセージが作成されます。以下のフィールドは共通なので、メッセージ同士の対応関係を把握できます。

- 同じ相関 ID
- PCF ヘッダー:
 - 変更前のメッセージ・ヘッダーは MsgSeqNumber': 1, 'Control': 'NOT_LAST' になります。
 - 変更後のメッセージ・ヘッダーは MsgSeqNumber': 2, 'Control': 'LAST' になります。

生成されるすべての構成イベント・メッセージは、キュー SYSTEM.ADMIN.CONFIG.EVENT に入れられません。

関連概念

[17 ページの『構成イベント』](#)

構成イベントは、構成イベントが明示的に要求されたときに生成されます。また、オブジェクトが作成、変更、または削除されたときには自動的に生成されます。

関連資料

[オブジェクトの作成](#)

[オブジェクトの変更](#)

[オブジェクトの削除](#)

[オブジェクトのリフレッシュ](#)

[12 ページの『イベント・タイプ』](#)

このページでは、キュー・マネージャーまたはチャンネル・インスタンスが報告できる観測イベントのタイプを示します。

構成イベント生成

このページでは、構成イベントを生成するコマンドを示し、構成イベントが生成されない環境について説明します。

構成イベント・メッセージは、CONFIGEV キュー・マネージャー属性が ENABLED で、かつ以下の場合に構成イベント・キューに書き込まれます。

- 以下のいずれかのコマンド、または PCF でこれらに相当するコマンドが発行された場合。
 - DELETE AUTHINFO
 - DELETE CFSTRUCT
 - DELETE CHANNEL
 - DELETE NAMELIST
 - DELETE PROCESS
 - DELETE QMODEL/QALIAS/QREMOTE

- DELETE STGCLASS
- DELETE TOPIC
- REFRESH QMGR
- オブジェクトに対する変更がなくても、以下のいずれかのコマンド、または PCF でこれらに相当するコマンドが発行された場合。
 - 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 (CONFIGEV 属性が DISABLED であり、ENABLED に変更されない場合を除く)
- キューに対する変更がなくても、以下のいずれかのコマンド、または PCF でこれらに相当するコマンドが、一時動的ではない ローカル・キューに対して発行された場合。
 - DELETE QLOCAL
 - DEFINE/ALTER QLOCAL
- オブジェクトに対する変更がなくても、一時動的キュー以外に対して MQSET 呼び出しが発行された場合。

構成イベントが生成されない場合

構成イベント・メッセージは、以下の環境では生成されません。

- コマンドまたは MQSET 呼び出しが失敗した場合。
- キュー・マネージャーがイベント・キューに構成イベントを書き込もうとして、エラーが検出された場合。この場合、コマンドまたは MQSET 呼び出しは完了しますが、イベント・メッセージは生成されません。
- 一時動的キュー。
- TRIGGER キュー属性に対して内部変更が行われた場合。
- 構成イベント・キュー SYSTEM.ADMIN.CONFIG.EVENT (ただし、REFRESH QMGR コマンドによる場合は除く)。
- クラスタリングを変更する REFRESH/RESET CLUSTER コマンドと RESUME/SUSPEND QMGR コマンド。
- キュー・マネージャーを作成または削除している場合。

関連概念

[プログラマブル・コマンド・フォーマットの概要](#)

42 ページの『[構成イベント](#)』

構成イベントとは、オブジェクトが作成、変更、または削除されたときに生成される通知であり、明示的な要求によって生成されることもあります。

関連資料

[MQSC コマンド](#)

[MQSET - オブジェクト属性の設定](#)

[MQSET - オブジェクト属性の設定](#)

構成イベントの使用法

このページでは、システムに関する情報を入手するために、どのように構成イベントを使用できるかを示し、構成イベントの使用に影響する可能性がある、CMDSCOPE などの要因について説明します。

構成イベントは、次の目的に使用できます。

1. レポートを作成し、システムの構造に関する情報を生成できる、中央構成リポジトリを作成および保守する。
2. 監査証跡を生成する。例えば、オブジェクトが予期せずに変更される場合、変更を行った人物および変更が行われた時刻に関する情報を保管することができます。

これは特に、コマンド・イベントも有効にされるときに役立ちます。MQSC または PCF コマンドを実行した結果、構成イベントおよびコマンド・イベントが生成された場合、両方のイベント・メッセージが同じ相関 ID をそのメッセージ記述子で共有します。

MQSET 呼び出しまたは以下のいずれかのコマンドでは、

- DEFINE object
- ALTER object
- DELETE object

キュー・マネージャー属性 CONFIGEV が有効になっているが、構成イベント・メッセージを構成イベント・キューに書き込むことができない場合(イベント・キューが定義されていない場合など)でも、コマンドまたは MQSET 呼び出しは無関係に実行されます。

CMDSCOPE の効果

CMDSCOPE が使用されるコマンドの場合、構成イベント・メッセージは、コマンドが入力されるキュー・マネージャーではなく、コマンドが実行されるキュー・マネージャー上で生成されます。ただし、CMDSCOPE を使用するコマンドがソース・キュー・マネージャーによって生成されたコマンドであっても、イベント・データ内のすべての発生元情報およびコンテキスト情報は、入力されたとおりに元のコマンドに関連付けます。

キュー共有グループに、現行バージョンではないキュー・マネージャーが含まれている場合、現行バージョンであり、以前のバージョンではないキュー・マネージャー上で CMDSCOPE によって実行されるコマンド用のイベントが生成されます。これは、コマンドが入力されるキュー・マネージャーが以前のバージョンであるかのようにして行われます。しかし、そのような場合には、イベント・データ内にはコンテキスト情報が含まれません。

関連概念

[プログラマブル・コマンド・フォーマットの概要](#)

42 ページの『[構成イベント](#)』

構成イベントとは、オブジェクトが作成、変更、または削除されたときに生成される通知であり、明示的な要求によって生成されることもあります。

関連資料

[MQSET - オブジェクト属性の設定](#)

[MQSET - オブジェクト属性の設定](#)

「オブジェクトのリフレッシュ」 構成イベント

「オブジェクトのリフレッシュ」 構成イベントは、明示的に要求されたときのみ発生するため、他の構成イベントとは異なります。

作成イベント、変更イベント、および削除イベントは、MQSET 呼び出しまたはオブジェクトを変更するコマンドによって生成されますが、オブジェクトのリフレッシュ・イベントは、MQSC コマンド REFRESH QMGR、または PCF でそれに相当するコマンドによって明示的に要求されたときのみ発生します。

REFRESH QMGR コマンドは、構成イベントを生成するその他のすべてのコマンドとは異なります。その他のすべてのコマンドは特定のオブジェクトに適用され、そのオブジェクト用の単一の構成イベントを生成します。REFRESH QMGR コマンドは、キュー・マネージャーによって保管されるすべてのオブジェクト定

義を潜在的に表す、多数の構成イベント・メッセージを作成することができます。選択されたオブジェクトごとに、1つのイベント・メッセージが生成されます。

REFRESH QMGR コマンドは、含まれているオブジェクトの数をフィルターに掛けるために3つの選択基準の組み合わせを使用します。

- オブジェクト名
- オブジェクト・タイプ
- リフレッシュ間隔

REFRESH QMGR コマンドに選択基準が指定されていない場合には、それぞれの選択基準ごとにデフォルト値が使用され、キュー・マネージャーによって保管されるすべてのオブジェクト定義についてリフレッシュ構成イベント・メッセージが生成されます。このため、許容できないほど処理時間が長くなり、イベント・メッセージが生成される可能性があります。いくつかの選択基準を指定することを検討してください。

リフレッシュ・イベントを生成する REFRESH QMGR コマンドは、以下の状態で使用できます。

- オブジェクトが最近操作されたかどうかにかかわらず、システム内のすべてまたはいくつかのオブジェクトに関する構成データが必要であるとき (例えば、構成イベントが初めて使用可能になるとき)

複数のコマンドを使用して、各コマンドでオブジェクトの選択は異なるとしても、全体ですべてのオブジェクトが含まれるようにすることを検討してください。

- SYSTEM.ADMIN.CONFIG.EVENT キューにエラーがあった場合。この状態では、作成、変更、または削除イベント用の構成イベント・メッセージは生成されません。キュー上のエラーが訂正されたとき、イベント・メッセージの生成を要求するために Refresh Queue Manager コマンドを使用することができます。これは、キューにエラーがあった間は失われていたものです。この場合は、リフレッシュ間隔を、キューが使用不能だった時間に設定することを検討してください。

関連概念

42 ページの『[構成イベント](#)』

構成イベントとは、オブジェクトが作成、変更、または削除されたときに生成される通知であり、明示的な要求によって生成されることもあります。

関連資料

[REFRESH QMGR](#)

[キュー・マネージャーのリフレッシュ](#)

コマンド・イベント

コマンド・イベントは、MQSC または PCF コマンドが正常に実行されたことを示す通知です。

イベント・データには、以下の情報が含まれています。

発生元情報

コマンドの発行元であるキュー・マネージャー、コマンドを発行したユーザー ID、およびコマンドの発行方法 (例えばコンソール・コマンドによるものなど) を記述します。

コンテキスト情報

コマンド・メッセージからのメッセージ・データ内のコンテキスト情報のレプリカです。コマンドがメッセージを使用して入力されていない場合、コンテキスト情報は省略されます。

コマンドが SYSTEM.COMMAND.INPUT キューにメッセージとして入力された場合、コンテキスト情報はイベント・データ内にも含まれます。

コマンド情報

発行されたコマンドのタイプを記述します。

コマンド・データ

- PCF コマンドでは、コマンド・データのレプリカ
- MQSC コマンドでは、コマンド・テキスト

コマンド・データ形式は、必ずしも元のコマンドの形式と一致しません。例えば、[マルチプラットフォーム](#)では、元の要求が MQSC コマンドの場合でも、コマンド・データ形式は常に PCF 形式になります。

生成される各コマンド・イベント・メッセージは、コマンド・イベント・キュー SYSTEM.ADMIN.COMMAND.EVENT に入れます。

関連資料

コマンド

[12 ページの『イベント・タイプ』](#)

このページでは、キュー・マネージャーまたはチャンネル・インスタンスが報告できる観測イベントのタイプを示します。

コマンド・イベント生成

このページでは、コマンド・イベントが生成される状況を示し、コマンド・イベントが生成されない環境について説明します。

コマンド・イベント・メッセージは以下の場合に生成されます。

- CMDEV キュー・マネージャー属性に ENABLED が指定され、MQSC コマンドまたは PCF コマンドが正常に実行された場合。
- CMDEV キュー・マネージャー属性に NODISPLAY が指定され、DISPLAY コマンド (MQSC) および照会コマンド (PCF) 以外のコマンドが正常に実行された場合。
- MQSC コマンド ALTER QMGR、または PCF コマンド Change Queue Manager を実行するときに、CMDEV キュー・マネージャー属性が次のどちらかの条件を満たしている場合。
 - 変更後に、CMDEV に DISABLED が指定されていない
 - 変更前に、CMDEV に DISABLED が指定されていなかった

コマンド・イベント・キュー SYSTEM.ADMIN.COMMAND.EVENT に対してコマンドを実行する際に、キューが引き続き存在し、書き込み禁止に設定されていなければ、コマンド・イベントは生成されます。

コマンド・イベントが生成されない場合

コマンド・イベント・メッセージは以下の状況では生成されません。

- コマンドが失敗した場合。
- キュー・マネージャーがイベント・キューにコマンド・イベントを書き込もうとして、エラーが検出された場合。この場合、コマンドは関係なく実行されますが、イベント・メッセージは生成されません。
- MQSC コマンド REFRESH QMGR TYPE (EARLY)。
- MQSC コマンド START QMGR MQSC。
- MQSC コマンド SUSPEND QMGR (パラメーター LOG が指定された場合)。
- MQSC コマンド RESUME QMGR (パラメーター LOG が指定された場合)。

関連概念

[46 ページの『コマンド・イベント』](#)

コマンド・イベントは、MQSC または PCF コマンドが正常に実行されたことを示す通知です。

関連資料

[REFRESH QMGR](#)

[START QMGR](#)

[SUSPEND QMGR](#)

[RESUME QMGR](#)

[SUSPEND QMGR、RESUME QMGR およびクラスター](#)

コマンド・イベントの使用法

このページでは、実行されたコマンドの監査証跡を生成するために、どのようにコマンド・イベントを使用できるかを示します。

例えば、オブジェクトが予期せずに変更される場合、変更を行った人物および変更が行われた時刻に関する情報を保管することができます。これは特に、構成イベントも有効にされるときに役立ちます。MQSC または PCF コマンドを実行した結果、コマンド・イベントおよび構成イベントが生成された場合、両方のイベント・メッセージが同じ関連 ID をそのメッセージ記述子で共有します。

コマンド・イベント・メッセージが生成されたが、そのメッセージをコマンド・イベント・キューに書き込むことができない場合 (例えばコマンド・イベント・キューが定義されていない場合) でも、コマンド・イベントが生成されたコマンドは無関係に実行されます。

CMDSCOPE の効果

CMDSCOPE が使用されるコマンドの場合、コマンド・イベント・メッセージは、コマンドが入力される キュー・マネージャーではなく、コマンドが実行されるキュー・マネージャー上で生成されます。ただし、CMDSCOPE を使用するコマンドがソース・キュー・マネージャーによって生成されたコマンドであっても、イベント・データ内のすべての発生元情報およびコンテキスト情報は、入力されたとおりに元のコマンドに関連付けます。

関連概念

[46 ページの『コマンド・イベント』](#)

コマンド・イベントは、MQSC または PCF コマンドが正常に実行されたことを示す通知です。

[47 ページの『コマンド・イベント生成』](#)



このページでは、コマンド・イベントが生成される状況を示し、コマンド・イベントが生成されない環境について説明します。

関連資料

[MQSC コマンド](#)

[グループ別の PCF コマンドと応答](#)

Multi ログ・イベント

ログ・イベントとは、キュー・マネージャーが新しいログ・エクステント  (または IBM i の場合はジャーナル・レシーバー) への書き込みを開始したという通知のことです。  ログ・イベント・メッセージは、IBM MQ for z/OS では使用できません。

イベント・データには、以下の情報が含まれています。

- 現行ログ・エクステントの名前。
- 再始動リカバリーに必要な最も古いログ・エクステントの名前。
- メディア・リカバリーに必要な最も古いログ・エクステントの名前。
- ログ・エクステントのあるディレクトリー。
- アーカイブ通知を必要とする最も古いログ・エクステントの名前。

アーカイブ・ログ管理用にログ・イベントが生成されます。つまり、**ARCHLOG** が変更されると、**ARCHLOG** 値がログ・イベント・メッセージに組み込まれます。

これらの全パラメーターについて詳しくは、[DISPLAY QMSTATUS](#) を参照してください。

CURRLOG 値および **ARCHLOG** 値を使用して、アーカイブ対象を判別できます。 **CURRLOG** が変わるたびに、**CURRLOG** より小さい番号のファイルをアーカイブ用に送信できます。エクステントのアーカイブが完了したら、[SET LOG](#) を呼び出してキュー・マネージャーに通知する必要があります。

ARCHLOG は、アーカイブが必要な最も古いエクステントです。そのエクステントをアーカイブし、[SET LOG](#) を呼び出してキュー・マネージャーにエクステントがアーカイブされたことを通知すると、キュー・マネージャーは **ARCHLOG** をシーケンスの次のエクステントに移動します。キュー・マネージャーは、**ARCHLOG** より古いエクステントを削除または再使用するようスケジュールします。

このエクステントに対して **SET LOG** が呼び出されると、**ARCHLOG** が変更されるため、新しいイベントが発行されます。新しいイベントの後に何をアーカイブする必要があるかを知っておく必要があります。

何らかの理由でアーカイブ・プロセスが失敗し、大量の通知が発生した場合、管理者はコマンド `RESET QMGR TYPE (ARCHLOG)` を手動で発行できます。これにより、指定したエクステントより古いすべてのエクステント (指定したエクステントを含む) を再使用または削除できることがキュー・マネージャーに通知されます。

生成される各ロガー・イベント・メッセージは、ロガー・イベント・キュー `SYSTEM.ADMIN.LOGGER.EVENT` に入れられます。

関連資料

ロガー

12 ページの『イベント・タイプ』

このページでは、キュー・マネージャーまたはチャンネル・インスタンスが報告できる観測イベントのタイプを示します。

Multi ログ・イベント生成

このページでは、ロガー・イベントが生成される状況を示し、ロガー・イベントが生成されない環境について理解します。

ロガー・イベント・メッセージは以下の場合に生成されます。

- LOGGEREV キュー・マネージャー属性に `ENABLED` が指定され、キュー・マネージャーが新しいログ・エクステント (または `IBM i` の場合はジャーナル・レシーバー) に書き込みを開始した場合。
- LOGGEREV キュー・マネージャー属性に `ENABLED` が指定され、キュー・マネージャーが開始された場合。
- LOGGEREV キュー・マネージャー属性が `DISABLED` から `ENABLED` に変更された場合。
- キュー・マネージャーがアーカイブ通知 (`ARCHLOG`) を待機している最も古いログ・エクステントの名前がアーカイブされたことが、LOGGEREV キュー・マネージャー属性に通知された場合。

ヒント: `RESET QMGR MQSC` コマンドを使用して、キュー・マネージャーに新しいログ・エクステントへの書き込みを開始することを要求できます。

ロガー・イベントが生成されない場合

ロガー・イベント・メッセージは以下の状況では生成されません。



- キュー・マネージャーが循環ロギングを使用するように構成されている場合。

この場合、LOGGEREV キュー・マネージャー属性は `DISABLED` に設定され、変更はできません。

- キュー・マネージャーがイベント・キューにロガー・イベントを書き込もうとして、エラーが検出された場合。この場合、イベントが発生させたアクションは完了しますが、イベント・メッセージは生成されません。

関連概念

48 ページの『ロガー・イベント』

ロガー・イベントとは、キュー・マネージャーが新しいログ・エクステント  (または `IBM i` の場合はジャーナル・レシーバー) への書き込みを開始したという通知のことです。  ロガー・イベント・メッセージは、`IBM MQ for z/OS` では使用できません。

関連資料

LoggerEvent (MQLONG)

LoggerEvent (10 桁の符号付き整数)

RESET QMGR

Multi ログ・イベントの使用法



このページでは、キュー・マネージャー再始動またはメディア・リカバリーには不要になったログ・エクステントを判別するために、どのようにロガー・イベントを使用できるかを説明します。

余分なログ・エクステントは、災害復旧用の磁気テープなどのメディアにアーカイブした後、アクティブなログ・ディレクトリーから除去することができます。余分なログ・エクステントを定期的に削除すると、ディスク・スペースの使用量が最小限に保たれます。

LOGGEREV キュー・マネージャー属性が有効になっているが、ロガー・イベント・メッセージをロガー・イベント・キューに書き込むことができない場合(イベント・キューが定義されていない場合など)でも、イベントを発生させるアクションは無関係に続行されます。

関連概念

[48 ページの『ロガー・イベント』](#)

ロガー・イベントとは、キュー・マネージャーが新しいログ・エクステント  (または IBM i の場合はジャーナル・レシーバー) への書き込みを開始したという通知のことです。  ロガー・イベント・メッセージは、IBM MQ for z/OS では使用できません。

関連資料

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(10 桁の符号付き整数\)](#)

[49 ページの『ロガー・イベント生成』](#)

このページでは、ロガー・イベントが生成される状況を示し、ロガー・イベントが生成されない環境について理解します。

ロガー・イベント・キューをモニターするためのサンプル C プログラム

このページを使用して、新規イベント・メッセージについてロガー・イベント・キューをモニターし、それらのメッセージを読み取り、メッセージの内容を標準出力に出力するサンプルの C プログラムを確認します。

```

/*****
/*
/* Program name: AMQSLOG0.C
/*
/* Description: Sample C program to monitor the logger event
/* 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');
}
}

```

出力例

このアプリケーションは、次のような形式の出力を生成します。

```

/*****
/* 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
-----

```

関連概念

[49 ページの『ロガー・イベントの使用法』](#)

このページでは、キュー・マネージャー再始動またはメディア・リカバリーには不要になったログ・エクステンツを判別するために、どのようにロガー・イベントを使用できるかを説明します。

[47 ページの『コマンド・イベントの使用法』](#)

このページでは、実行されたコマンドの監査証跡を生成するために、どのようにコマンド・イベントを使用できるかを示します。

関連資料

[49 ページの『ロガー・イベント生成』](#)

このページでは、ロガー・イベントが生成される状況を示し、ロガー・イベントが生成されない環境について理解します。

権限構成イベント

権限構成イベントは、コマンド・ライン、MQSC、PCF、または対応する IBM i コマンドにより、何らかのセキュリティ管理操作から変更が加えられる際の出力です。

イベント・データには、以下の情報が含まれています。

発生元情報

変更が行われたキュー・マネージャー、変更を行ったユーザーの ID、および行われた変更方法 (コンソール・コマンドなどによる) について説明します。

コンテキスト情報

コマンド・メッセージからのメッセージ・データ内のコンテキスト情報のレプリカです。

コマンドが SYSTEM.ADMIN.COMMAND.QUEUE キューにメッセージとして入力された場合、コンテキスト情報はイベント・データ内に含まれます。

権限レコード ID

プロファイル名と、権限レコードのオブジェクト・タイプで構成されます。

オブジェクトの属性

権限レコード内のすべての属性の値で構成されます。

権限レコードの変更イベントの場合には 2 つのメッセージが生成されます。1 つは変更前の情報で、もう 1 つは変更後の情報が含まれています。

生成される各イベント・メッセージは、キュー SYSTEM.ADMIN.CONFIG.EVENT に入れられます。

関連資料

12 ページの『イベント・タイプ』

このページでは、キュー・マネージャーまたはチャンネル・インスタンスが報告できる観測イベントのタイプを示します。

権限構成イベントの生成

このページでは、権限構成イベントが生成される状況を示し、権限構成イベントが生成されない環境について説明します。

権限構成イベントは、権限レコードの属性の変更を通知します。権限構成イベントには、以下の 3 つのタイプがあります。

- [変更権限レコード](#)
- [削除権限レコード](#)
- [更新権限レコード](#)

権限イベント・メッセージは、**CONFIGEV** キュー・マネージャー属性が **ENABLED** に設定されており、かつ、以下のいずれかのコマンドまたは MQSC でこれらに相当するコマンドが発行された場合に (権限レコードに対する実際の変更がなくても)、構成イベント・キューに書き込まれます。

- [Delete Authority Record PCF コマンド](#)
- [Set Authority Record PCF コマンド](#)
- [setmqaut 制御コマンド](#)
- [RVKMQMAUT CL コマンド](#)
- [GRTMQMAUT CL コマンド](#)

権限構成イベントが生成されない場合

権限構成イベント・メッセージは、以下の環境では生成されません。

- コマンドが失敗した場合。
- キュー・マネージャーがイベント・キューにメッセージを書き込もうとして、エラーが検出された場合。この場合、コマンドは完了しますが、イベント・メッセージは生成されません。
- キュー・マネージャーを作成または削除している場合。

- 削除コマンド上の **AUTHREC** オプションとは無関係に、オブジェクトが削除される場合。対応するコマンド・イベントによりこの操作が示されますが、個々のユーザーに関する権限レコードには適用されません。

関連概念

46 ページの『コマンド・イベント』

コマンド・イベントは、MQSC または PCF コマンドが正常に実行されたことを示す通知です。

関連資料

[REFRESH QMGR](#)

Multi 観測イベントをモニターするためのサンプル・プログラム (Multiplatforms)

amqsevt は IBM MQ for Multiplatforms に付属しており、キュー・マネージャーが作成できる観測イベントをフォーマットします。このプログラムは、イベント・キューからメッセージを読み取り、人が読めるストリングにフォーマットします。

ソースとバイナリーの両方がサンプル・プログラムとして提供されています。このサンプルは、IBMi を含むすべての Multiplatforms で利用できます。

単一のバイナリー・ファイル **amqsevt** (または **amqsevt.exe**) は、サンプル・ファイル・セットで出荷され、サンプルの bin (`tools\c\samples\bin` または `bin64`) ディレクトリーにインストールされます。

ソース・ファイル **amqsevta.c** もサンプル・ファイル・セットに同梱されており、サンプル・ディレクトリー (Windows の場合は `tools\c\samples`) にインストールされています。

なお、このプログラムは、[MQCB](#) を使用してメッセージを取得することにより、複数のイベント・キューから読み取ったり、複数のトピックにサブスクライブしたりできます。

クライアントとして実行している場合、このサンプルは z/OS を含むどのキュー・マネージャーにも接続できます。

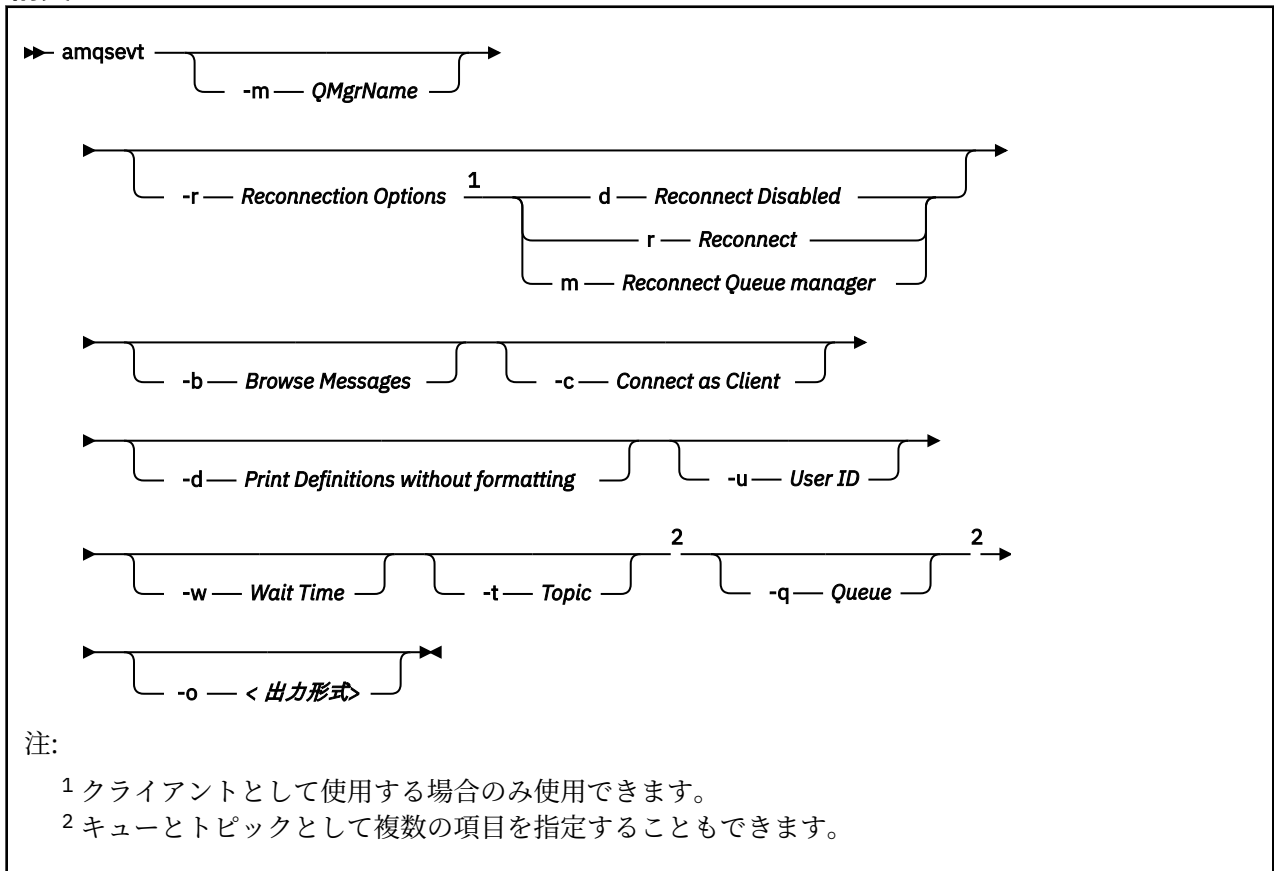


重要: パラメーターを指定せずにこのプログラムを使用できます。その場合、このプログラムはデフォルトのキュー・マネージャーに接続し、標準的なイベント・キューのセット (SYSTEM.ADMIN.*.EVENT) からメッセージを読み取ろうとします。

この状態では、Enter キーを押してこのプログラムを終了するまで、このプログラムは永続的にメッセージを待機します。

しかし、さまざまなオプションを記述してこのプログラムを使用する場合の方が多くは必ずです。

構文



オプション・パラメーター

-m *QueueManagerName*

イベントの読み取りに固有のキュー・マネージャーを指定します。

-r *Reconnection Options*

クライアントとして使用する場合の自動再接続オプション。指定できる値は以下のとおりです。

d

クライアントを再接続できません。

r

クライアントを再接続します。

m

キュー・マネージャーを再接続します。

-b

レコードのブラウズのみ行い、メッセージを破壊的に読み取ることはありません。

-c

接続をクライアントとして選択します。

-d

2 番目の例で使用される印刷モードを選択します。ヘッダー・ファイル内にあるとおりに、MQI 定数が正確に印刷されます。

-u *User ID*

特定のユーザーを指定し、パスワードを要求するプロンプトが出されるようにします。

-w *Wait*

指定された秒数の間にイベント・メッセージが着信しないと、プログラムが終了するようにします。

時刻を指定しない場合、Enter キーを押さないとプログラムが正常終了しないことに注意してください。

-t Topic と

-q Queue

-q オプションと -t オプションは両方とも複数回コマンド・ライン上に指定できます。

したがって、プログラムの1回の実行で、複数の標準キューやトピック (イベントがトピックに送信される場合) から読み取ることができます。

コマンド・ライン上でキューやトピックの名前を指定しないと、デフォルトのイベント・キューが開かれます。

注: プログラムは、z/OS キュー・マネージャーにクライアントとして接続しているかどうかを検出し、z/OS に SYSTEM.ADMIN.LOGGER.EVENT キュー。

トピックの使用時には、終了時にすべてがクリーンアウトされるように、プログラムは非永続サブスクリプションと管理対象キューを使用します。

-o <output format>

出力形式。指定可能な値は以下のとおりです。

text

標準テキスト形式。これがデフォルト値です。

json

標準 JSON 形式。JSON を理解できるすべてのアプリケーションがこの出力を取得して、直接処理できます。

出力例

以下の3つの例は、このプログラムからの出力を示しています。

最初の例は、デフォルトのフォーマット・オプションを使用します。この場合、プログラムはフィールドの MQI 定義を採用し、出力をフォーマットして読みやすくします。

```
**** 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
```

2番目の例では、-d オプションを使用した代替フォーマットを示します。この場合、MQI 定数を変換しようとしません。特定の MQI 値を検索するスクリプト・ツールの場合は、こちらの方が望ましいことがあります。

```
**** 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
```

3番目の例は、コマンドの JSON 出力を示しています。

```
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,
    ...
  }
}

```

使用例

次の例は、複数のキューを使用する方法を示しています。

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

関連概念

9 ページの『イベント・モニター』

イベント・モニターは、キュー・マネージャー・ネットワーク内で発生する観測イベントを検出する処理です。観測イベントとは、キュー・マネージャーまたはチャンネル・インスタンスが検出するイベントが論理的に組み合わせられたものです。このようなイベントが発生すると、イベント・メッセージという特殊なメッセージがキュー・マネージャーまたはチャンネル・インスタンスによってイベント・キューに書き込まれます。

10 ページの『観測イベント』

観測イベントは、キュー・マネージャーまたはチャンネル・インスタンスが検出する条件と、イベント・メッセージと呼ばれる特別メッセージをイベント・キューに入れる条件を論理的に組み合わせたものです。

関連資料

C プログラミング

50 ページの『ロガー・イベント・キューをモニターするためのサンプル C プログラム』

このページを使用して、新規イベント・メッセージについてロガー・イベント・キューをモニターし、これらのメッセージを読み取り、メッセージの内容を標準出力に出力するサンプルの C プログラムを確認します。

メッセージ・モニター

メッセージ・モニターは、メッセージがキュー・マネージャー・ネットワークを通過した経路を特定する処理です。メッセージに対して実行されたアクティビティーのタイプと順序を特定することで、そのメッセージの経路を判別できます。

メッセージがキュー・マネージャー・ネットワークを通過するとき、そのメッセージに基づいて、さまざまなプロセスによってアクティビティーが実行されます。メッセージの経路を判別するには、以下のいずれかの手法を使用してください。

- IBM MQ 経路表示アプリケーション (dspmqrte)
- アクティビティー記録
- 経路トレース・メッセージング

これらの技法はすべて、特別なメッセージを生成します。この特別なメッセージには、そのメッセージがキュー・マネージャー・ネットワークを通過したときに、それに対して実行されたアクティビティーに関

する情報が含まれます。これらの特別なメッセージで返される情報を使用することで、以下の目的を達成することができます。

- メッセージ・アクティビティを記録する。
- メッセージが最後に確認された場所を特定する。
- キュー・マネージャー・ネットワークにあるルーティング問題を検出する。
- キュー・マネージャー・ネットワークにあるルーティング問題の原因の判別を支援する。
- キュー・マネージャー・ネットワークが正しく実行されていることを確認する。
- キュー・マネージャー・ネットワークの実行方法を習得できるようにする。
- パブリッシュされたメッセージをトレースする。

関連概念

[メッセージのタイプ](#)

アクティビティと操作

アクティビティとは、メッセージに対してアプリケーションが実行する個別のアクションです。アクティビティは、アプリケーションが実行する処理の1つの部分である操作から構成されます。

アクティビティの例としては、以下のようなものがあります。

- メッセージ・チャンネル・エージェント (MCA) が伝送キューからチャンネルを通じてメッセージを送信する。
- MCA がチャンネルからメッセージを受信し、それをターゲット・キューに書き込む。
- アプリケーションがキューからメッセージを読み取り、応答として返信メッセージを書き込む。
- IBM MQ パブリッシュ/サブスクライブ・エンジンがメッセージを処理する。

アクティビティは、1つ以上の操作から構成されます。操作とは、アプリケーションが実行する処理の1つの部分です。例えば、伝送キューからチャンネルを通じてメッセージを送信する MCA のアクティビティは、以下の操作から構成されます。

1. 伝送キューからメッセージを読み取る (読み取り 操作)。
2. チャンネルを通じてメッセージを送信する (送信 操作)。

パブリッシュ/サブスクライブ・ネットワークでは、IBM MQ パブリッシュ/サブスクライブ・エンジンによって処理されているメッセージのアクティビティは、以下の複数の操作から成る可能性があります。

1. メッセージをトピック・ストリングに書き込む (書き込み 操作)。
2. メッセージ受信の対象とみなされるサブスクライバーごとに 0 個以上の操作 (パブリッシュ 操作、パブリッシュ廃棄 操作、またはパブリッシュ除外 操作)。

アクティビティからの情報

メッセージがキュー・マネージャー・ネットワーク経由で送信されるときに情報を記録することにより、メッセージに対して実行されたアクティビティの順序を特定できます。メッセージに対して実行されたアクティビティの順序から、キュー・マネージャー・ネットワークを通るメッセージの経路を判別できます。入手できる情報は、次のとおりです。

メッセージが最後に確認された場所

メッセージが意図された宛先に届いていない場合、完全なメッセージ経路または部分的なメッセージ経路から、メッセージが最後に確認された場所を特定できます。

キュー・マネージャー・ネットワークの構成問題

キュー・マネージャー・ネットワークを通ったメッセージの経路を調べると、メッセージが予定された場所に届かなかったことを確認できます。例えば、チャンネルがアクティブでないために、メッセージが代替経路を取った可能性があるなど、これには多くの原因が考えられます。

パブリッシュ/サブスクライブ・アプリケーションの場合は、トピックにパブリッシュされるメッセージの経路や、サブスクライバーにパブリッシュされた結果としてキュー・マネージャー・ネットワークを流れるメッセージの経路も特定できます。

このような場合、システム管理者は、キュー・マネージャー・ネットワークに問題があるかどうかを判断し、必要に応じて、それを修正することができます。

メッセージ経路

メッセージ経路を判別する理由に応じて、以下の2つの一般的な方式を使用できます。

経路トレース・メッセージについて記録されたアクティビティー情報を使用する

経路トレース・メッセージでは、特定の目的のためにアクティビティー情報が記録されます。これらのメッセージを使用して、キュー・マネージャー・ネットワークの構成問題を判別したり、メッセージが最後に確認された場所を特定することができます。意図された宛先に届かなかったメッセージが最後に確認された場所を特定するために経路トレース・メッセージを生成する場合、元のメッセージを模倣することができます。これにより、元のメッセージが取った経路に従っている可能性が最も高い経路トレース・メッセージが得られます。

経路トレース・メッセージは、IBM MQ 経路表示アプリケーションを使用して生成することができます。

元のメッセージについて記録されたアクティビティー情報を使用する

アクティビティー・レコーディングでメッセージを使用可能にし、メッセージについて記録されたアクティビティー情報を入手することができます。メッセージが意図された宛先に届いていない場合は、記録されたアクティビティー情報を使用して、メッセージが最後に確認された場所を特定できます。元のメッセージのアクティビティー情報を使用することで、メッセージが最後に確認された場所に至る、考えられる限り最も正確なメッセージ経路を判別できます。この方式を使用するには、元のメッセージがアクティビティー・レコーディングで使用できるようになっていることが必要です。

警告: アクティビティー・レコーディングでは、キュー・マネージャー・ネットワーク内のすべてのメッセージを使用可能にしないようにしてください。アクティビティー・レコーディングで使用可能なメッセージは、それ自体について多数のアクティビティー・レポートが生成される可能性があります。アクティビティー・レコーディングでキュー・マネージャー・ネットワーク内のすべてのメッセージを使用可能にすると、キュー・マネージャー・ネットワークのトラフィックが許容できないレベルまで増加する可能性があります。

関連概念

[59 ページの『メッセージ・モニター』](#)

メッセージ・モニターは、メッセージがキュー・マネージャー・ネットワークを通過した経路を特定する処理です。メッセージに対して実行されたアクティビティーのタイプと順序を特定することで、そのメッセージの経路を判別できます。

[61 ページの『メッセージ経路の手法』](#)

アクティビティー・レコーディングおよび経路トレース・メッセージングは、メッセージがキュー・マネージャー・ネットワーク経由で送信されるときに、そのメッセージのアクティビティー情報を記録することができる手法です。

[68 ページの『経路トレース・メッセージング』](#)

経路トレース・メッセージングとは、経路トレース・メッセージを使用して、メッセージのアクティビティー情報を記録する技法です。経路トレース・メッセージングには、経路トレース・メッセージのキュー・マネージャー・ネットワークへの送信が含まれます。

関連タスク

[独自のメッセージ・チャンネル・エージェントの作成](#)

メッセージ経路の手法

アクティビティー・レコーディングおよび経路トレース・メッセージングは、メッセージがキュー・マネージャー・ネットワーク経由で送信されるときに、そのメッセージのアクティビティー情報を記録することができる手法です。

アクティビティー記録

適切なレポート・オプションがメッセージに指定されている場合、このメッセージはアプリケーションに対して、このメッセージがキュー・マネージャー・ネットワーク経由で送信されたときにアクティビティー・レポートを生成するように要求します。アプリケーションがメッセージに対してアクティ

ビティを実行すると、アクティビティ・レポートが生成され、適切な場所に送信されます。アクティビティ・レポートには、このメッセージに対して実行されたアクティビティに関する情報が含まれます。

メッセージ経路を判別するには、アクティビティ・レポートを使用して収集されたアクティビティ情報を、正しい順序に並べ替えてください。

経路トレース・メッセージング

経路トレース・メッセージングは、経路トレース・メッセージをキュー・マネージャー・ネットワークに送信することを伴う技法です。アプリケーションが経路トレース・メッセージに対してアクティビティを実行するときに、アクティビティ情報を経路トレース・メッセージのメッセージ・データに累積するか、またはアクティビティ・レポートを生成することができます。アクティビティ情報を経路トレース・メッセージのメッセージ・データに累積する場合、メッセージがターゲット・キューに到着すると、経路トレース・メッセージの情報をすべて含む経路トレース応答メッセージが生成され、適切な場所に送信されます。

経路トレース・メッセージは、経路トレース・メッセージに対して実行されたアクティビティの順序を記録するためにのみ使用されるので、アクティビティ・レポートを要求する通常のメッセージに比べてより多くの処理オプションが使用できます。

アクティビティ・レコーディングと経路トレース・メッセージングの比較

アクティビティ・レコーディングと経路トレース・メッセージングはどちらも、メッセージがキュー・マネージャー・ネットワークを通過した経路を判別するためのアクティビティ情報を提供します。どちらの方式にもそれぞれ利点があります。

利点	アクティビティ レコード	経路トレース・メ ッセージング
メッセージが最後に確認された場所を特定可能	はい	はい
キュー・マネージャー・ネットワークの構成問題を判別可能	はい	はい
どのメッセージからでも要求可能 (経路トレース・メッセージとの使用に限りません)	はい	いいえ
メッセージ・データは未変更のままになる	はい	いいえ
メッセージは通常どおり処理される	はい	いいえ
アクティビティ情報をメッセージ・データに累積可能	いいえ	はい
オプションでターゲット・キューへのメッセージ配信が可能	いいえ	はい
メッセージが無限ループにキャッチされた場合に、それを検出して 対処することが可能	いいえ	はい
アクティビティ情報を確実に正しい順序に並べ替えることが可能	いいえ	はい
アクティビティ情報を表示するためのアプリケーションが用意 されている	いいえ	はい

メッセージ経路の完全性

メッセージに対して実行されたアクティビティの完全な順序を特定できない場合があります。その場合は、部分的なメッセージ経路しか判別できません。メッセージ経路の完全性は、メッセージが通過するキュー・マネージャー・ネットワークにより直接に影響を受けます。

キュー・マネージャーに接続された MCA およびユーザー作成アプリケーションは、メッセージの代わりに実行されたアクティビティに関連する情報を記録できます。アクティビティ情報のレコーディングは、キュー・マネージャー属性 ACTIVREC および ROUTEREC によって制御されます。キュー・マネージャー・ネットワークは、完全なメッセージ経路を判別できます。

アクティビティー情報の保管方法

IBM MQ では、アクティビティー情報は、アクティビティー・レポート、経路トレース・メッセージ、または経路トレース応答メッセージに保管されます。いずれの場合も、情報は *Activity PCF* グループと呼ばれる構造に保管されます。経路トレース・メッセージや経路トレース応答メッセージには、メッセージに対して実行されたアクティビティーの数に応じて、多数の *Activity PCF* グループを入れることができます。アクティビティー・レポートには、1つの *Activity PCF* グループが含まれます。これは、記録されたアクティビティーごとに、個別にアクティビティー・レポートが生成されるためです。

経路トレース・メッセージングで、その他の情報も記録できます。この追加情報は、*TraceRoute PCF* グループと呼ばれる構造に保管されます。*TraceRoute PCF* グループには、多数の *PCF* 構造が含まれます。この *PCF* 構造は、追加のアクティビティー情報の保管や、経路トレース・メッセージがキュー・マネージャー・ネットワークを通過するときの処理方法を決定するオプションの指定に使用されます。

関連概念

[63 ページの『アクティビティー記録』](#)

アクティビティー・レコーディングは、メッセージがキュー・マネージャー・ネットワークを通った経路を判別するための手法です。メッセージがたどった経路を判別するために、メッセージに対して実行されたアクティビティーが記録されます。

[68 ページの『経路トレース・メッセージング』](#)

経路トレース・メッセージングとは、経路トレース・メッセージを使用して、メッセージのアクティビティー情報を記録する技法です。経路トレース・メッセージングには、経路トレース・メッセージのキュー・マネージャー・ネットワークへの送信が含まれます。

関連資料

[74 ページの『TraceRoute PCF グループ』](#)

TraceRoute PCF グループの属性によって、経路トレース・メッセージの振る舞いが制御されます。*TraceRoute PCF* グループは、各経路トレース・メッセージのメッセージ・データ内に存在します。

[109 ページの『アクティビティー・レポート・メッセージ・データ』](#)

このページを使用して、アクティビティー・レポート・メッセージ内の *Activity PCF* グループに含まれたパラメーターを確認します。一部のパラメーターは、特定の操作が実行された場合にのみ返されます。

アクティビティー記録

アクティビティー・レコーディングは、メッセージがキュー・マネージャー・ネットワークを通った経路を判別するための手法です。メッセージがたどった経路を判別するために、メッセージに対して実行されたアクティビティーが記録されます。

アクティビティー・レコーディングを使用すると、メッセージに対して実行されたアクティビティーをそれぞれアクティビティー・レポートに記録することができます。アクティビティー・レポートはレポート・メッセージの一種です。アクティビティー・レポートにはそれぞれ、メッセージに対してアクティビティーを実行したアプリケーションに関する情報、アクティビティーが実行された時期、およびアクティビティーの一部として実行された操作に関する情報が記録されます。通常、アクティビティー・レポートは応答先キューに送信され、ここで1つにまとめられます。メッセージに関連するアクティビティー・レポートを調べることで、そのメッセージがキュー・マネージャー・ネットワークを通った経路を判別できます。

アクティビティー・レポートの使用法

メッセージがキュー・マネージャー・ネットワークを通過するときに、アクティビティー・レポートを生成できます。アクティビティー・レポートの情報は、次のように使用することができます。

メッセージが最後に確認された場所を特定する

アクティビティー・レコーディングで使用可能にされたメッセージが、意図された宛先に届いていない場合に、メッセージがキュー・マネージャー・ネットワーク経由で送信されたときにそのメッセージについて生成されたアクティビティー・レポートを調べて、メッセージが最後に確認された場所を特定することができます。

キュー・マネージャー・ネットワークの構成問題を判別する

アクティビティ・レコーディングで使用可能な多数のメッセージを、キュー・マネージャー・ネットワークに送信することができます。各メッセージに関連するアクティビティ・レポートを調べること、これらのメッセージが期待された経路を通っていないことが明らかになります。原因として、チャンネルが停止していたために、メッセージは強制的に代替経路を取らされた可能性があるなど、多くのことが考えられます。この場合、システム管理者は、キュー・マネージャー・ネットワークに問題があるかどうかを判断し、問題がある場合はそれを修正することができます。

注：IBM MQ 経路表示アプリケーションを使用すると、アクティビティ・レコーディングを経路トレース・メッセージとともに使用できます。

アクティビティ・レポート形式

アクティビティ・レポートは、メッセージに対してアクティビティを実行したアプリケーションによって生成される PCF メッセージです。アクティビティ・レポートは、以下のようなメッセージ記述子とメッセージ・データを含む標準の IBM MQ レポートです。

メッセージ記述子

- MQMD 構造

メッセージ・データ

- 組み込み PCF ヘッダー (MQEPH)
- アクティビティ・レポート・メッセージ・データ

アクティビティ報告書のレポート・メッセージ・データは Activity PCF グループから構成されます。ただし、経路トレース・メッセージについて生成された場合は TraceRoute PCF グループから構成されます。

関連資料

[MQMD - メッセージ記述子](#)

[MQEPH - 組み込み PCF ヘッダー](#)

アクティビティ・レコーディングの制御

キュー・マネージャー・レベルでアクティビティ・レコーディングを使用可能にします。キュー・マネージャー・ネットワーク全体を使用可能にするには、ネットワーク内のすべてのキュー・マネージャーをそれぞれ個別に、アクティビティ・レコーディングで使用可能にします。より多くのキュー・マネージャーを使用可能にすればするほど、アクティビティ・レポートも生成されていきます。

このタスクについて

メッセージをキュー・マネージャー経由で送信するときに、そのメッセージのアクティビティ・レポートを生成するには、アクティビティ・レポートを要求するようにメッセージを定義し、アクティビティ・レコーディングでキュー・マネージャーを使用可能にし、メッセージに対してアクティビティを実行しているアプリケーションがアクティビティ・レポートを確実に生成できるようにします。

メッセージをキュー・マネージャー経由で送信するときに、そのメッセージのアクティビティ・レポートを生成しないようにする場合は、アクティビティ・レコーディングでキュー・マネージャーを使用不可にします。

手順

1. メッセージのアクティビティ・レポートを要求

- a) メッセージのメッセージ記述子の *Report* フィールドに `MQRO_ACTIVITY` を指定します。
- b) メッセージのメッセージ記述子の *ReplyToQ* フィールドに、応答先キューの名前を指定します。

警告：アクティビティ・レコーディングでは、キュー・マネージャー・ネットワーク内のすべてのメッセージを使用可能にしないようにしてください。アクティビティ・レコーディングで使用可能なメッセージは、それ自体について多数のアクティビティ・レポートが生成される可能性があります。アクティビティ・レコーディングでキュー・マネージャー・ネットワーク内のすべてのメッセージを使

用可能にすると、キュー・マネージャー・ネットワークのトラフィックが許容できないレベルまで増加する可能性があります。

2. アクティビティ・レコーディングでキュー・マネージャーを使用可能または使用不可にします。

このキュー・マネージャー属性の値は、MQSC コマンド ALTER QMGR にパラメーター ACTIVREC を指定して変更できます。値は次のいずれかです。

MSG

アクティビティ・レコーディングでキュー・マネージャーを使用可能にします。生成されたアクティビティ・レポートはすべて、メッセージのメッセージ記述子で指定された応答先キューに送信されます。これがデフォルト値です。

QUEUE

アクティビティ・レコーディングでキュー・マネージャーを使用可能にします。生成されたアクティビティ・レポートはすべて、ローカル・システム・キュー SYSTEM.ADMIN.ACTIVITY.QUEUE に送信されます。また、このシステム・キューは、共通キューへのアクティビティ・レポートの転送にも使用できます。

DISABLED

アクティビティ・レコーディングでキュー・マネージャーを使用不可にします。このキュー・マネージャーの有効範囲内にいる間は、アクティビティ・レポートは生成されません。

例えば、キュー・マネージャーをアクティビティ・レコーディングで使用可能にし、生成されたアクティビティ・レポートはすべてローカル・システム・キュー SYSTEM.ADMIN.ACTIVITY.QUEUE に送信されるように指定するには、次の MQSC コマンドを使用します。

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

要確認: ACTIVREC キュー・マネージャー属性を変更すると、実行中の MCA では、チャンネルが再始動されるまでこの変更を検出しません。

3. MCA がメッセージのアクティビティ・レポートを生成するかどうかを判別する場合に使用するものと同じアルゴリズムを、アプリケーションが確実に使用するようにします。
 - a) メッセージでアクティビティ・レポートの生成が要求されたことを確認します。
 - b) 現在、メッセージがあるキュー・マネージャーがアクティビティ・レコーディングで使用可能であることを確認します。
 - c) アクティビティ・レポートを、ACTIVREC キュー・マネージャー属性で決定されたキューに書き込みます。

アクティビティ・レポートでの共通キューのセットアップ

アクティビティ・レポートがローカル・システム・キューに送信されるときに、特定のメッセージに関連するアクティビティ・レポートの場所を判別するには、単一ノード上の共通のキューを使用するとより効率的です。

始める前に

アクティビティ・レコーディングでキュー・マネージャーを使用可能にし、生成されたアクティビティ・レポートをすべてローカル・システム・キュー SYSTEM.ADMIN.ACTIVITY.QUEUE に送信するよう指定するには、**ACTIVREC** パラメーターを設定します。

このタスクについて

キュー・マネージャー・ネットワーク内の数多くのキュー・マネージャーで、アクティビティ・レポートをローカル・システム・キューに送信するように設定すると、特定のメッセージに関連するアクティビティ・レポートの場所を特定することに時間がかかることがあります。代替策としては、単一ノードを使用します。単一ノードとは、共通キューをホストするキュー・マネージャーです。キュー・マネージャー・ネットワーク内のキュー・マネージャーはどれも、この共通キューにアクティビティ・レポートを送信できます。共通キューの使用には、キュー・マネージャーが、メッセージで指定された応答先キューにアクティビティ・レポートを送信する必要がなくなり、また特定のメッセージに関連するアクティビティ・レポートの場所を特定する際に、1つのキューを照会するだけで済むという利点があります。

共通キューをセットアップするには、以下のステップを実行します。

手順

1. キュー・マネージャーを単一ノードとして選択または定義します。
2. 単一ノードで、共通キューとして使用するキューを選択または定義します。
3. アクティビティ・レポートを共通キューに送信するすべてのキュー・マネージャーで、ローカル・システム・キュー `SYSTEM.ADMIN.ACTIVITY.QUEUE` を、リモート・キュー定義として再定義します。
 - a) リモート・キュー・マネージャー名としての単一ノードの名前を指定します。
 - b) リモート・キュー名としての共通キューの名前を指定します。

メッセージ経路情報の判別

メッセージ経路を判別するには、収集されたアクティビティ・レポートから情報を取得します。必要な情報を調べて、アクティビティ・レポートを正しい順番に並べ替えることができるための、十分なアクティビティ・レポートが応答先キューにあるかどうかを確認します。

このタスクについて

アクティビティ・レポートが応答先キューに書き込まれた順序は、アクティビティが実行された順序と必ずしも関連しているわけではありません。アクティビティ・レポートは手動で並べ替える必要があります。ただし、アクティビティ・レポートが経路トレース・メッセージに生成されている場合は例外で、この場合は IBM MQ 経路表示アプリケーションを使用して、アクティビティ・レポートを並べ替えることができます。

必要な情報を取得するための、十分なアクティビティ・レポートが応答先キューにあるかどうかを確認します。

手順

1. アクティビティ・レポートと元のメッセージの ID を比較して、応答先キュー上の関連するアクティビティ・レポートをすべて特定します。アクティビティ・レポートと元のメッセージを関連付けられるようなレポート・オプションを元のメッセージに設定しておくことが必要です。
2. 応答先キューの特定された各アクティビティ・レポートを並べ替えます。
アクティビティ・レポートの以下のパラメーターを使用できます。

OperationType

実行された操作のタイプにより、現在のアクティビティ・レポートの直前、または直後に生成されたアクティビティ・レポートを判別できる場合があります。

例えば、あるアクティビティ・レポートに、MCA が伝送キューからチャンネルを通じてメッセージを送信したに関する詳細が示されているとします。このアクティビティ・レポートには、最後の操作として `send` という *OperationType* が記録され、このメッセージはチャンネル CH1 を使用して、宛先キュー・マネージャー QM1 に送信されたことが詳述されています。つまり、このメッセージに対して実行される次のアクティビティはキュー・マネージャー QM1 で発生し、チャンネル CH1 からの `receive` 操作で始まります。この情報を使用して、次のアクティビティ・レポートを特定することができます。ただし、次のアクティビティ・レポートが存在し、取得済みである場合に限りです。

OperationDate と OperationTime

各アクティビティ・レポートにある操作の日時により、アクティビティの大まかな順序を判別できます。

警告: キュー・マネージャー・ネットワーク内の各キュー・マネージャーでシステム・クロックが同期されている場合を除き、日時で並べ替えてもアクティビティ・レポートが正しい順序で並ぶ保証はありません。順序付けは手動で行うことが必要です。

アクティビティ・レポートの順序は、メッセージがキュー・マネージャー・ネットワークを通過した経路、または経路の一部を表します。

3. 順序付けられたアクティビティ・レポート内にあるアクティビティ情報から必要な情報を入手します。
メッセージに関する情報が不十分な場合は、アクティビティ・レポートをさらに取得できる場合があります。

アクティビティ・レポートをさらに取得

メッセージ経路を判別するには、収集されたアクティビティ・レポートから十分な情報が入手できる必要があります。メッセージで指定された応答先キューからメッセージに関連するアクティビティ・レポートを取得したが、必要な情報がない場合には、さらにアクティビティ・レポートを探します。

このタスクについて

追加のアクティビティ・レポートの場所を特定するには、以下の手順を行います。

手順

1. アクティビティ・レポートを共通キューに送信する、キュー・マネージャー・ネットワーク内にあるすべてのキュー・マネージャーについて、元のメッセージの *MsgId* と一致する *CorrelId* を持つアクティビティ・レポートをこの共通キューから取得します。
2. アクティビティ・レポートを共通キューに送信しない、キュー・マネージャー・ネットワーク内にあるすべてのキュー・マネージャーについて、アクティビティ・レポートを以下のようにして取得します。
 - a) 既存のアクティビティ・レポートを調べて、メッセージの送信に使用されたキュー・マネージャーを特定します。
 - b) これらのキュー・マネージャーで、アクティビティ・レコーディングで使用可能なキュー・マネージャーを特定します。
 - c) これらのキュー・マネージャーで、指定された応答先キューにアクティビティ・レポートを返さなかったものを特定します。
 - d) 特定したキュー・マネージャーごとに、システム・キュー `SYSTEM.ADMIN.ACTIVITY.QUEUE` を確認し、元のメッセージの *MsgId* と一致する *CorrelId* を持つアクティビティ・レポートをすべて取得します。
 - e) システム・キューにアクティビティ・レポートが見つからない場合、キュー・マネージャーの送達不能キューにアクティビティ・レポートがないかどうかを確認します。
アクティビティ・レポートが送達不能キューに送信されるのは、レポート・オプション `MQRO_DEAD_LETTER_Q` が設定されている場合だけです。
3. 取得したアクティビティ・レポートを正しい順序に並べ替えます。
つまり、このアクティビティ・レポートの順序が、メッセージがキュー・マネージャー・ネットワークを通過した経路、または経路の一部を表します。
4. 順序付けられたアクティビティ・レポート内にあるアクティビティ情報から必要な情報を入手します。
状況によっては、記録されたアクティビティ情報が、指定された応答先キュー、共通キュー、またはシステム・キューに届かない場合があります。

アクティビティ情報を取得できない状況

メッセージに対して実行された、完全なアクティビティの順序をすべて判別するには、各アクティビティに関連する情報を取得する必要があります。アクティビティに関連する情報が記録されていなかった場合、またはそうした情報が得られなかった場合は、アクティビティの順序の一部だけを判別することができます。

アクティビティ情報は以下の状況では記録されません。

- IBM WebSphere® MQ 6.0 より前のキュー・マネージャーによってメッセージが処理されている。
- アクティビティ・レコーディングで使用可能になっていないキュー・マネージャーによってメッセージが処理されている。
- メッセージを処理するはずのアプリケーションが実行されていない。

記録されたアクティビティ情報は、以下のような状況では、指定された応答先キューに届きません。

- アクティビティ・レポートを応答先キューに送信するように定義されているチャンネルがない。
- アクティビティ・レポートを応答先キューに送信するチャンネルが稼働していない。
- 応答先キューのあるキュー・マネージャー (キュー・マネージャー別名) へアクティビティ・レポートを返すためのリモート・キュー定義がない。
- 元のメッセージを生成したユーザーが、キュー・マネージャー別名に対するオープン権限または書き込み権限を持っていない。
- 元のメッセージを生成したユーザーが、応答先キューに対するオープン権限または書き込み権限を持っていない。
- 応答先キューが書き込み禁止になっている。

記録されたアクティビティ情報は、以下のような状況では、システム・キューまたは共通キューに届きません。

- 共通キューを使用する場合に、アクティビティ・レポートを共通キューに送信するように定義されたチャンネルがない。
- 共通キューを使用する場合に、アクティビティ・レポートを共通キューに送信するためのチャンネルが稼働していない。
- 共通キューを使用する場合に、誤ってシステム・キューが定義されている。
- 元のメッセージを生成したユーザーが、システム・キューに対するオープン権限または書き込み権限を持っていない。
- システム・キューが書き込み禁止になっている。
- 共通キューを使用する場合に、元のメッセージを生成したユーザーが共通キューに対するオープン権限または書き込み権限を持っていない。
- 共通キューを使用する場合に、共通キューが書き込み禁止になっている。

このような状況では、アクティビティ・レポートにレポート・オプション `MQRO_DISCARD_MSG` が指定されていなければ、アクティビティ・レポートを送達不能キューから取り出すことができます。ただし、アクティビティ・レポートが拒否されたキュー・マネージャーで送達不能キューが定義されていた場合に限りです。アクティビティ・レポートにこのレポート・オプションが指定されるのは、アクティビティ・レポートが生成された元のメッセージで、メッセージ記述子のレポート・フィールドに `MQRO_PASS_DISCARD_AND_EXPIRY` と `MQRO_DISCARD_MSG` の両方が指定されていた場合だけです。

経路トレース・メッセージング

経路トレース・メッセージングとは、経路トレース・メッセージを使用して、メッセージのアクティビティ情報を記録する技法です。経路トレース・メッセージングには、経路トレース・メッセージのキュー・マネージャー・ネットワークへの送信が含まれます。

経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信されると、アクティビティ情報が記録されます。このアクティビティ情報には、アクティビティを実行したアプリケーション、それらのアクティビティが実行された時期、およびアクティビティの一部として実行された操作に関する情報が含まれます。経路トレース・メッセージングを使用して記録された情報は、以下のような目的で使用できます。

メッセージが最後に確認された場所を特定する

メッセージが意図された宛先に届いていない場合に、経路トレース・メッセージに記録されたアクティビティ情報を使用して、メッセージが最後に確認された場所を特定できます。経路トレース・メッセージは、元のメッセージと同じ経路をたどるように意図して、元のメッセージと同じターゲット宛先を持つキュー・マネージャー・ネットワークに送信されます。アクティビティ情報は、経路トレース・メッセージのメッセージ・データに累積するか、またはアクティビティ・レポートを使用して記録することができます。経路トレース・メッセージが元のメッセージと同じ経路をたどる可能性を高めるために、経路トレース・メッセージを変更して、元のメッセージに似せることができます。

キュー・マネージャー・ネットワークの構成問題を判別する

経路トレース・メッセージがキュー・マネージャー・ネットワークに送信され、アクティビティ情報が記録されます。経路トレース・メッセージについて記録されたアクティビティ情報を調べること

で、経路トレース・メッセージが期待された経路をたどらなかったことが明確になります。原因として、チャンネルが非アクティブであるために、メッセージは強制的に代替経路を取らされた可能性があるなど、多くのことが考えられます。この場合、システム管理者は、キュー・マネージャー・ネットワークに問題があるかどうかを判断し、問題がある場合はそれを修正することができます。

IBM MQ 経路表示アプリケーションを使用して、経路トレース・メッセージを構成、生成、およびキュー・マネージャー・ネットワークに入れます。

警告: 経路トレース・メッセージを配布リストに入れた場合、その結果は不確定なものになります。

関連概念

126 ページの『[経路トレース・メッセージ参照](#)』

このページを使用して、経路トレース・メッセージの形式についての概要を説明します。経路トレース・メッセージ・データには、経路トレース・メッセージによって引き起こされたアクティビティを記述したパラメーターが含まれています。

アクティビティ情報の記録方法

経路トレース・メッセージングによって、アクティビティ情報を経路トレース・メッセージのメッセージ・データに記録するか、またはアクティビティ・レポートを使用することができます。あるいは、両方の手法を使用することもできます。

経路トレース・メッセージのメッセージ・データにアクティビティ情報を累積する

経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信されると、経路トレース・メッセージに対して実行されたアクティビティについての情報が、経路トレース・メッセージのメッセージ・データに累積されます。アクティビティ情報は *Activity PCF* グループに保管されます。経路トレース・メッセージに対して実行されたアクティビティごとに、*Activity PCF* グループが、経路トレース・メッセージのメッセージ・データの *PCF* ブロックの末尾に書き込まれます。

追加のアクティビティ情報が、経路トレース・メッセージング中に、*TraceRoute PCF* グループと呼ばれる *PCF* グループに記録されます。追加のアクティビティ情報は *PCF* グループに保管され、記録されたアクティビティの順序を判別するために使用できます。この技法は、*TraceRoute PCF* グループの *Accumulate* パラメーターによって制御されます。

アクティビティ・レポートを使用してアクティビティ情報を記録する

経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信されると、経路トレース・メッセージに対して実行されたアクティビティごとに、アクティビティ・レポートを生成できます。アクティビティ情報は *Activity PCF* グループに保管されます。経路トレース・メッセージに対して実行されたアクティビティごとに、*Activity PCF* グループが含まれたアクティビティ・レポートが生成されます。経路トレース・メッセージのアクティビティ・レコーディングは、他のメッセージと同じ方法で行われます。

経路トレース・メッセージについて生成されたアクティビティ・レポートには、他のメッセージについて生成されたレポートに比べて、追加のアクティビティ情報が含まれています。この追加情報は *TraceRoute PCF* グループに返されます。*TraceRoute PCF* グループに含まれている情報は、アクティビティ・レポートが生成された時点以降についてのみ正確です。この追加情報は、経路トレース・メッセージに対して実行されたアクティビティの順序を判別するため使用できます。

記録されたアクティビティ情報の取得

経路トレース・メッセージがその意図した宛先に届くか、または廃棄される場合、アクティビティ情報の取得に使用する方法は、その情報の記録方法によって決まります。

始める前に

アクティビティ情報に詳しくない場合は、69 ページの『[アクティビティ情報の記録方法](#)』を参照してください。

このタスクについて

経路トレース・メッセージがその意図された宛先に届くか、または廃棄される場合は、以下のアクティビティ情報の取得方法を使用してください。

手順

- 経路トレース・メッセージを取得する。

TraceRoute PCF グループの *Deliver* パラメーターは、経路トレース・メッセージの到着時にこれをターゲット・キューに入れるか、または廃棄するかを制御します。経路トレース・メッセージがターゲット・キューに送信された場合、その経路トレース・メッセージはこのキューから取得できます。そうすれば、IBM MQ 経路表示アプリケーションを使用して、アクティビティ情報を表示できます。

アクティビティ情報を経路トレース・メッセージのメッセージ・データに累積するように要求するには、*TraceRoute* PCF グループの *Accumulate* パラメーターを `MQROUTE_ACCUMULATE_IN_MSG` に設定します。

- 経路トレース応答メッセージを使用する。

経路トレース・メッセージがその意図された宛先に届いたとき、または経路トレース・メッセージをキュー・マネージャー・ネットワーク内でさらに転送することができないときは、経路トレース応答メッセージを生成できます。経路トレース応答メッセージには、経路トレース・メッセージのすべてのアクティビティ情報の複製が含まれています。経路トレース応答メッセージは、指定された応答先キュー、またはシステム・キュー `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE` のいずれかに送信されます。IBM MQ 経路表示アプリケーションを使用して、アクティビティ情報を表示します。

経路トレース応答メッセージを要求するには、*TraceRoute* PCF グループの *Accumulate* パラメーターを `MQROUTE_ACCUMULATE_AND_REPLY` に設定します。

- アクティビティ・レポートを使用する。

経路トレース・メッセージにアクティビティ・レポートが生成された場合、アクティビティ情報を取得する前に、そのアクティビティ・レポートを見つけておく必要があります。この後、アクティビティの順序を調べるために、アクティビティ・レポートを並べ替える必要があります。

経路トレース・メッセージングの制御

経路トレース・メッセージングをキュー・マネージャー・レベルで使用可能にすることにより、そのキュー・マネージャーの有効範囲内にあるアプリケーションがアクティビティ情報を経路トレース・メッセージに書き込めるようにします。キュー・マネージャー・ネットワーク全体を使用可能にするには、ネットワーク内のすべてのキュー・マネージャーをそれぞれ個別に、経路トレース・メッセージングで使用可能にします。より多くのキュー・マネージャーを使用可能にすればするほど、アクティビティ・レポートも生成されていきます。

始める前に

アクティビティ・レポートを使用して経路トレース・メッセージにアクティビティ情報を記録する場合は、64 ページの『[アクティビティ・レコーディングの制御](#)』を参照してください。

このタスクについて

経路トレース・メッセージがキュー・マネージャー経由で送信される際に、そのメッセージにアクティビティ情報を記録するには、以下の手順を実行してください。

手順

- アクティビティ情報を経路トレース・メッセージに記録する方法を定義します。
73 ページの『[経路トレース・メッセージの生成および構成](#)』を参照してください。
- アクティビティ情報を経路トレース・メッセージに累積する場合には必ず、キュー・マネージャーを経路トレース・メッセージングで使用可能にしてください。
- アクティビティ情報を経路トレース・メッセージに累積する場合には必ず、経路トレース・メッセージ上のアクティビティを実行するアプリケーションで、経路トレース・メッセージのメッセージ・データにアクティビティ情報を書き込めるようにしてください。

関連概念

73 ページの『[経路トレース・メッセージの生成および構成](#)』

経路トレース・メッセージは、固有のメッセージ記述子の部分とメッセージ・データの部分から構成されます。経路トレース・メッセージを生成するには、メッセージを手動で作成するか、または IBM MQ 経路表示アプリケーションを使用します。

関連タスク

64 ページの『アクティビティ・レコーディングの制御』

キュー・マネージャー・レベルでアクティビティ・レコーディングを使用可能にします。キュー・マネージャー・ネットワーク全体を使用可能にするには、ネットワーク内のすべてのキュー・マネージャーをそれぞれ個別に、アクティビティ・レコーディングで使用可能にします。より多くのキュー・マネージャーを使用可能にすればするほど、アクティビティ・レポートも生成されていきます。

経路トレース・メッセージングでキュー・マネージャーを使用可能化

経路トレース・メッセージングでキュー・マネージャーを使用可能にするか使用不可にするかを制御するには、キュー・マネージャー属性 ROUTEREC を使用します。

このキュー・マネージャー属性の値は、MQSC コマンド ALTER QMGR にパラメーター ROUTEREC を指定して変更できます。値には以下のいずれかの値を指定できます。

MSG

経路トレース・メッセージングでキュー・マネージャーを使用可能にします。キュー・マネージャーの有効範囲内のアプリケーションは、アクティビティ情報を経路トレース・メッセージに書き込むことができます。

TraceRoute PCF グループの *Accumulate* パラメーターが MQROUTE_ACCUMULATE_AND_REPLY に設定されている場合、経路トレース・メッセージに対して次に実行されるアクティビティは、以下のいずれかになります。

- 廃棄
- ローカル・キュー (ターゲット・キューまたは送達不能キュー) への書き込み
- 経路トレース・メッセージに対して実行されたアクティビティの総数が *TraceRoute PCF* グループのパラメーター *MaxActivities* の値を超える

経路トレース応答メッセージが生成され、経路トレース・メッセージのメッセージ記述子で指定された応答先キューに送信されます。

QUEUE

経路トレース・メッセージングでキュー・マネージャーを使用可能にします。キュー・マネージャーの有効範囲内のアプリケーションは、アクティビティ情報を経路トレース・メッセージに書き込むことができます。

TraceRoute PCF グループの *Accumulate* パラメーターが MQROUTE_ACCUMULATE_AND_REPLY に設定されている場合、経路トレース・メッセージに対して次に実行されるアクティビティは、以下のいずれかになります。

- 廃棄
- ローカル・キュー (ターゲット・キューまたは送達不能キュー) への書き込み
- 経路トレース・メッセージに対して実行されたアクティビティの総数が *TraceRoute PCF* グループのパラメーター *MaxActivities* の値を超える

経路トレース応答メッセージが生成され、ローカル・システム・キュー SYSTEM.ADMIN.TRACE.ROUTE.QUEUE に送信されます。

DISABLED

経路トレース・メッセージングでキュー・マネージャーを使用不可にします。アクティビティ情報は経路トレース・メッセージに累積されませんが、*TraceRoute PCF* グループは、このキュー・マネージャーの有効範囲内にある間は更新可能です。

例えば、キュー・マネージャーを経路トレース・メッセージングで使用不可にするには、次の MQSC コマンドを使用します。

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

要確認: ROUTEREC キュー・マネージャー属性を変更すると、実行中の MCA では、チャンネルが再始動されるまでこの変更を検出しません。

経路トレース・メッセージングでアプリケーションを使用可能化

ユーザー・アプリケーションが経路トレース・メッセージングを使用できるようにするには、メッセージ・チャンネル・エージェント (MCA) で使用されるアルゴリズムをユーザーのアルゴリズムの基礎にします。

始める前に

経路トレース・メッセージの形式に詳しくない場合は、[126 ページの『経路トレース・メッセージ参照』](#)を参照してください。

このタスクについて

メッセージ・チャンネル・エージェント (MCA) が経路トレース・メッセージングで使用可能になっています。経路トレース・メッセージングでユーザー・アプリケーションを使用可能にするには、MCA で使用されるアルゴリズムの、以下の手順を使用してください。

手順

1. 処理対象のメッセージが経路トレース・メッセージであるかどうかを判別します。
メッセージが経路トレース・メッセージの形式に合致していない場合、そのメッセージは経路トレース・メッセージとして処理されません。
2. アクティビティー情報を記録するかどうかを判別します。
実行されたアクティビティーの詳細レベルが *Detail* パラメーターで指定された詳細レベル以上であれば、特定の環境下でアクティビティー情報が記録されます。経路トレース・メッセージで累積が要求され、キュー・マネージャーが経路トレース・メッセージングで使用可能になっている場合、または経路トレース・メッセージでアクティビティー・レポートが要求され、キュー・マネージャーがアクティビティー・レコーディングで使用可能になっている場合にのみ、この情報が記録されます。
 - アクティビティー情報が記録される場合は、*RecordedActivities* パラメーターの値を増やします。
 - アクティビティー情報が記録されない場合は、*UnrecordedActivities* の値を増やします。
3. 経路トレース・メッセージに対して実行されたアクティビティーの総数が *MaxActivities* パラメーターの値を超えているかどうかを判別します。
アクティビティーの総数は *RecordedActivities*、*UnrecordedActivities*、および *DiscontinuityCount* の和です。
アクティビティーの総数が *MaxActivities* を超える場合は、フィードバック MQFB_MAX_ACTIVITIES でメッセージを拒否します。
4. *Accumulate* の値が MQROUTE_ACCUMULATE_IN_MSG または MQROUTE_ACCUMULATE_AND_REPLY に設定され、キュー・マネージャーが経路トレース・メッセージングで使用可能にされている場合は、経路トレース・メッセージのメッセージ・データの PCF ブロックの末尾に Activity PCF グループを書き込みます。
5. 経路トレース・メッセージをローカル・キューに送信します。
 - パラメーター *Deliver* が MQROUTE_DELIVER_NO に設定されている場合は、フィードバック MQFB_NOT_DELIVERED で経路トレース・メッセージを拒否します。
 - パラメーター *Deliver* が MQROUTE_DELIVER_YES に設定されている場合は、経路トレース・メッセージをローカル・キューに送信します。
6. 以下のすべての条件に合致する場合は、経路トレース応答メッセージを生成します。
 - 経路トレース・メッセージがローカル・キューに送信されたか、または拒否された
 - パラメーター *Accumulate* の値が MQROUTE_ACCUMULATE_AND_REPLY である
 - 経路トレース・メッセージングでキュー・マネージャーが使用可能になっている経路トレース応答メッセージは、ROUTEREC キュー・マネージャー属性で決定されたキューに入れられます。

7. 経路トレース・メッセージでアクティビティ・レポートが要求され、キュー・マネージャーがアクティビティ・レコーディングで使用可能になっている場合は、アクティビティ・レポートを生成します。

アクティビティ・レポートは、ACTIVREC キュー・マネージャー属性で決定されたキューに入れられます。

経路トレース・メッセージの生成および構成

経路トレース・メッセージは、固有のメッセージ記述子の部分とメッセージ・データの部分から構成されます。経路トレース・メッセージを生成するには、メッセージを手動で作成するか、または IBM MQ 経路表示アプリケーションを使用します。

経路トレース・メッセージは以下の部分から構成されます。

メッセージ記述子

Format フィールドが MQFMT_ADMIN または MQFMT_EMBEDDED_PCF に設定された MQMD 構造

メッセージ・データ

以下のいずれかの組み合わせで構成されます。

- PCF ヘッダー (MQCFH) および経路トレース・メッセージ・データ (*Format* が MQFMT_ADMIN に設定されている場合)
- 組み込み PCF ヘッダー (MQEPH)、経路トレース・メッセージ・データ、および追加のユーザー指定メッセージ・データ (*Format* が MQFMT_EMBEDDED_PCF に設定されている場合)

経路トレース・メッセージ・データは、TraceRoute PCF グループと 1 つ以上の Activity PCF グループで構成されます。

手動生成

手動で経路トレース・メッセージを生成する場合、Activity PCF グループは不要です。Activity PCF グループは、MCA またはユーザー作成アプリケーションが経路トレース・メッセージに対してアクティビティを実行するときに、経路トレース・メッセージのメッセージ・データに書き込まれます。

IBM MQ 経路表示アプリケーション

IBM MQ 経路表示アプリケーション **dspmqrte** を使用して、経路トレース・メッセージを構成および生成し、キュー・マネージャー・ネットワークに入れることができます。メッセージ記述子の *Format* パラメーターを MQFMT_ADMIN に設定します。IBM MQ 経路表示アプリケーションで生成された経路トレース・メッセージには、ユーザー・データを追加できません。

制約事項: dspmqrte は、IBM MQ for z/OS キュー・マネージャーでは発行できません。経路トレース・メッセージが経由する最初のキュー・マネージャーをこのタイプのキュー・マネージャーにする場合は、オプション・パラメーター **-c** を使用して、クライアントとしてキュー・マネージャーに接続します。

元のメッセージの模倣

経路トレース・メッセージを使用して、別のメッセージがキュー・マネージャー・ネットワークを通過した経路を判別する場合、経路トレース・メッセージが元のメッセージに似ているほど、経路トレース・メッセージが元のメッセージと同じ経路をたどる可能性が高くなります。

メッセージがキュー・マネージャー・ネットワーク内のどこに転送されるかは、以下のメッセージ特性によって影響を受けます。

優先順位

優先度は、メッセージのメッセージ記述子で指定できます。

Persistence

持続性は、メッセージのメッセージ記述子で指定できます。

有効期限

有効期限は、メッセージのメッセージ記述子で指定できます。

レポート・オプション

レポート・オプションは、メッセージのメッセージ記述子で指定できます。

メッセージ・サイズ

メッセージのサイズを模倣するために、メッセージのメッセージ・データに追加データを書き込むことができます。この場合、追加メッセージ・データは特に意味があるものでなくてもかまいません。

ヒント: IBM MQ 経路表示アプリケーションでは、メッセージ・サイズは指定できません。

メッセージ・データ

一部のキュー・マネージャー・ネットワークでは、メッセージが転送される場所を特定するために、コンテンツ・ベースのルーティングが使用されます。この場合、元のメッセージのメッセージ・データを模倣するために、経路トレース・メッセージのメッセージ・データを書き込む必要があります。

ヒント: IBM MQ 経路表示アプリケーションでは、メッセージ・データは指定できません。

TraceRoute PCF グループ

TraceRoute PCF グループの属性によって、経路トレース・メッセージの振る舞いが制御されます。

TraceRoute PCF グループは、各経路トレース・メッセージのメッセージ・データ内に存在します。

下の表には、MCA によって認識される、TraceRoute グループの各パラメーターがリストされています。79 ページの『追加のアクティビティー情報』で説明されているように、ユーザー作成アプリケーションがパラメーターを認識するように作成されている場合は、さらにパラメーターを追加できます。

パラメーター	タイプ
TraceRoute	MQCFGR
詳細	MQCFIN
RecordedActivities	MQCFIN
UnrecordedActivities	MQCFIN
DiscontinuityCount	MQCFIN
MaxActivities	MQCFIN
累積	MQCFIN
Forward	MQCFIN
Deliver	MQCFIN

以下は、TraceRoute PCF グループの各パラメーターの説明です。

Detail

記録するアクティビティー情報の詳細レベルを指定します。値には以下のいずれかの値を指定できます。

MQROUTE_DETAIL_LOW

ユーザー・アプリケーションで実行されたアクティビティーのみが記録されます。

MQROUTE_DETAIL_MEDIUM

MQROUTE_DETAIL_LOW で指定されたアクティビティーが記録されます。さらに、MCA によって実行されたアクティビティーも記録されます。

MQROUTE_DETAIL_HIGH

MQROUTE_DETAIL_LOW および MQROUTE_DETAIL_MEDIUM で指定されたアクティビティーが記録されます。この詳細レベルでは、MCA はこれ以上のアクティビティー情報を記録しません。このオプションは、ユーザー・アプリケーションが追加のアクティビティー情報を記録する場合にのみ有効です。例えば、ユーザー・アプリケーションが特定のメッセージ特性を考慮してメッセージがとる経路を決定する場合は、ルーティング・ロジックに関する情報がこの詳細レベルに含まれることがあります。

RecordedActivities

経路トレース・メッセージに対して実行された、記録されたアクティビティーの数を示します。アクティビティーは、そのアクティビティーについての情報が経路トレース・メッセージに書き込まれた場合、またはアクティビティー・レポートが生成された場合に、記録されたものと見なされます。記録されたアクティビティーごとに、RecordedActivities は 1 ずつ増加します。

UnrecordedActivities

経路トレース・メッセージに対して実行された、記録されていないアクティビティーの数を示します。アクティビティーは、経路トレース・メッセージングで使用可能なアプリケーションが、関連するアクティビティー情報を累積せず、またアクティビティー・レポートに書き込むこともしない場合に、記録されないものと見なされます。

経路トレース・メッセージに対して実行されたアクティビティーは、以下の状況においては記録されません。

- 実行されたアクティビティーの詳細レベルが、パラメーター *Detail* で指定された 詳細レベルより低い場合。
- 経路トレース・メッセージでアクティビティー・レポートが要求されたが累積は要求されず、キュー・マネージャーがアクティビティー・レコーディングで使用可能になっていない場合。
- 経路トレース・メッセージで累積が要求されたが、アクティビティー・レポートは要求されず、キュー・マネージャーが経路トレース・メッセージングで使用可能になっていない場合。
- 経路トレース・メッセージで累積とアクティビティー・レポートの両方が要求されたが、キュー・マネージャーがアクティビティー・レコーディングおよび経路トレース・メッセージングで使用可能になっていない場合。
- 経路トレース・メッセージで累積もアクティビティー・レポートも要求されない場合。

記録されないアクティビティーごとに、パラメーター *UnrecordedActivities* は 1 ずつ増加します。

DiscontinuityCount

経路トレース・メッセージングで使用可能になっていなかったアプリケーションのあるキュー・マネージャー経由で経路トレース・メッセージが送信された回数を指定します。この値は、キュー・マネージャーによって増加されます。この値が 0 より大きい場合は、メッセージ経路の一部のみが判別可能です。

MaxActivities

経路トレース・メッセージに対して実行可能なアクティビティーの最大数を指定します。

アクティビティーの総数は *RecordedActivities*、*UnrecordedActivities*、および *DiscontinuityCount* の和です。アクティビティーの総数は、*MaxActivities* の値を超えてはなりません。

MaxActivities の値は以下のとおりです。

正整数

アクティビティーの最大数。

アクティビティーの最大数を超えると、経路トレース・メッセージはフィードバック MQFB_MAX_ACTIVITIES で拒否されます。これにより、経路トレース・メッセージが無限ループに入った場合に、無制限に転送されるのを防ぐことができます。

MQROUTE_UNLIMITED_ACTIVITIES

経路トレース・メッセージに対して、アクティビティーを無制限に実行できます。

Accumulate

アクティビティー情報を累積するために使用する方法を指定します。値には以下のいずれかの値を指定できます。

MQROUTE_ACCUMULATE_IN_MSG

経路トレース・メッセージングでキュー・マネージャーが使用可能になっている場合、アクティビティー情報は経路トレース・メッセージのメッセージ・データに累積されます。

この値が指定されている場合、経路トレース・メッセージ・データは以下から構成されます。

- *TraceRoute* PCF グループ。
- ゼロ以上の *Activity* PCF グループ。

MQROUTE_ACCUMULATE_AND_REPLY

キュー・マネージャーが経路トレース・メッセージングで使用可能になっている場合、アクティビティー情報は経路トレース・メッセージのメッセージ・データに累積されます。また、以下が行われると、経路トレース応答メッセージが生成されます。

- 経路トレース・メッセージが IBM MQ キュー・マネージャーによって廃棄された。
- 経路トレース・メッセージが IBM MQ キュー・マネージャーによってローカル・キュー (ターゲット・キューまたは送達不能キュー) に入れられた。
- 経路トレース・メッセージに対して実行されたアクティビティーの数が *MaxActivities* の値を超えた。

この値が指定されている場合、経路トレース・メッセージ・データは以下から構成されます。

- *TraceRoute* PCF グループ。
- ゼロ以上の *Activity* PCF グループ。

MQRROUTE_ACCUMULATE_NONE

経路トレース・メッセージのメッセージ・データにアクティビティー情報を累積しません。

この値が指定されている場合、経路トレース・メッセージ・データは以下から構成されます。

- *TraceRoute* PCF グループ。

Forward

経路トレース・メッセージを転送可能な宛先を指定します。値は次のいずれかです。

MQRROUTE_FORWARD_IF_SUPPORTED

経路トレース・メッセージは、*TraceRoute* グループから *Deliver* パラメーターの値を受け取るキュー・マネージャーにのみ転送されます。

MQRROUTE_FORWARD_ALL

経路トレース・メッセージは、*Deliver* パラメーターの値が渡されるかどうかにかかわらず、すべてのキュー・マネージャーに転送されます。

キュー・マネージャーは、経路トレース・メッセージをリモート・キュー・マネージャーに転送するかどうかを決定するときに、次のアルゴリズムを使用します。

1. リモート・キュー・マネージャーが経路トレース・メッセージングをサポート可能であるかどうかを判別します。
 - リモート・キュー・マネージャーが経路トレース・メッセージングをサポート可能である場合、アルゴリズムは [ステップ 76 ページの『4』](#) に進みます。
 - リモート・キュー・マネージャーが経路トレース・メッセージングをサポート可能でない場合、アルゴリズムは [ステップ 76 ページの『2』](#) に進みます。
2. *TraceRoute* グループからの *Deliver* パラメーターにある *MQRROUTE_DELIVER_REJ_UNSUP_MASK* ビット・マスクに、認識されない配信オプションが含まれているかどうかを判別します。
 - 認識されない配信オプションが見つかったら、経路トレース・メッセージはフィードバック *MQFB_UNSUPPORTED_DELIVERY* で拒否されます。
 - 認識されない配信オプションが見つからない場合、アルゴリズムは [ステップ 76 ページの『3』](#) に進みます。
3. 経路トレース・メッセージにある *TraceRoute* PCF グループからのパラメーター *Deliver* の値を判別します。
 - *Deliver* に *MQRROUTE_DELIVER_YES* が指定されている場合、経路トレース・メッセージはリモート・キュー・マネージャーに転送されます。
 - *Deliver* に *MQRROUTE_DELIVER_NO* が指定されている場合、アルゴリズムは [ステップ 76 ページの『4』](#) に進みます。
4. *TraceRoute* グループからの *Forward* パラメーターにある *MQRROUTE_FORWARDING_REJ_UNSUP_MASK* ビット・マスクに、認識されない転送オプションが含まれているかどうかを判別します。
 - 認識されない転送オプションが見つかったら、経路トレース・メッセージはフィードバック *MQFB_UNSUPPORTED_FORWARDING* で拒否されます。
 - 認識されない転送オプションが見つからない場合、アルゴリズムは [ステップ 77 ページの『5』](#) に進みます。

5. 経路トレース・メッセージにある *TraceRoute* PCF グループからのパラメーター *Forward* の値を判別します。
- *Forward* に `MQROUTE_FORWARD_IF_SUPPORTED` が指定されている場合、経路トレース・メッセージはフィードバック `MQFB_NOT_FORWARDED` で拒否されます。
 - *Forward* に `MQROUTE_FORWARD_ALL` が指定されている場合、経路トレース・メッセージはリモート・キュー・マネージャーに転送できます。

Deliver

意図された宛先に経路トレース・メッセージが届いた場合に実行するアクションを指定します。ユーザー作成アプリケーションは、経路トレース・メッセージをターゲット・キューに入れる前にこの属性を検査する必要があります。値には以下のいずれかの値を指定できます。

MQROUTE_DELIVER_YES

到着すると、経路トレース・メッセージはターゲット・キューに入れられます。ターゲット・キューで取得操作を実行しているどのアプリケーションも、経路トレース・メッセージを取り出すことができます。

MQROUTE_DELIVER_NO

到着すると、経路トレース・メッセージはターゲット・キューに入れられません。メッセージは、そのレポート・オプションに従って処理されます。

経路トレース応答メッセージ用の共通キューのセットアップ

レポートがローカル・システム・キューに送信されるときに、特定のメッセージに関連する経路トレース応答メッセージの場所を判別するには、単一ノード上の共通キューを使用するとより効率的です。

始める前に

経路トレース・メッセージングでキュー・マネージャーを使用可能にし、生成された経路トレース応答メッセージをすべてローカル・システム・キュー `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE` に送信するよう指定するには、**ROUTEREC** パラメーターを設定します。

このタスクについて

キュー・マネージャー・ネットワーク内の数多くのキュー・マネージャーで、経路トレース応答メッセージをローカル・システム・キューに送信するように設定すると、特定のメッセージに関連する経路トレース応答メッセージの場所を特定することに時間がかかることがあります。代替策としては、単一ノードを使用します。単一ノードとは、共通キューをホストするキュー・マネージャーです。キュー・マネージャー・ネットワーク内のキュー・マネージャーはどれも、この共通キューに経路トレース応答メッセージを送信できます。共通キューの使用には、キュー・マネージャーが、メッセージで指定された応答先キューに経路トレース応答メッセージを送信する必要がなくなり、また特定のメッセージに関連する経路トレース応答メッセージの場所を特定する際に、1つのキューを照会するだけで済むという利点があります。

共通キューをセットアップするには、以下のステップを実行します。

手順

1. キュー・マネージャーを単一ノードとして選択または定義します。
2. 単一ノードで、共通キューとして使用するキューを選択または定義します。
3. 経路トレース応答メッセージを共通キューに転送するすべてのキュー・マネージャーで、ローカル・システム・キュー `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE` を、リモート・キュー定義として再定義します。
 - a) リモート・キュー・マネージャー名としての単一ノードの名前を指定します。
 - b) リモート・キュー名としての共通キューの名前を指定します。

記録された情報の取得および使用

以下の手法のいずれかを使用して、経路トレース・メッセージに記録されたアクティビティ情報を取得します。

アクティビティ情報が取得されない状況では、これが経路トレース応答メッセージにも当てはまることに注意してください。

経路トレース・メッセージが、アクティビティ・レコーディングと経路トレース・メッセージングの両方で使用不可に設定されているキュー・マネージャーによって処理される場合、アクティビティ情報は記録されません。

経路トレース応答メッセージからの情報の取得

アクティビティ情報を取得するには、経路トレース応答メッセージを見つけます。この後、メッセージを取得してアクティビティ情報を分析します。

このタスクについて

アクティビティ情報を経路トレース応答メッセージから取得できるのは、経路トレース応答メッセージの場所がわかっている場合のみです。メッセージを見つけて、以下のようにアクティビティ情報を処理します。

手順

1. 経路トレース・メッセージのメッセージ記述子に指定された応答先キューを確認します。経路トレース応答メッセージがこの応答先キューにない場合は、以下の場所を調べてください。
 - 経路トレース・メッセージのターゲット・キュー・マネージャー上のローカル・システム・キュー `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE`
 - 共通キュー (経路トレース応答メッセージ用の共通キューをセットアップしている場合)
 - キュー・マネージャー・ネットワーク内の他のすべてのキュー・マネージャー上にあるローカル・システム・キュー `SYSTEM.ADMIN.TRACE.ROUTE.QUEUE` (経路トレース・メッセージが送達不能キューに入れられた場合、またはアクティビティが最大数を超過した場合)
2. 経路トレース応答メッセージを取得します。
3. IBM MQ 経路表示アプリケーションを使用して、記録されたアクティビティ情報を表示します。
4. アクティビティ情報を調べて、必要な情報を入手します。

経路トレース・メッセージからの情報の取得

アクティビティ情報を取得するには、経路トレース・メッセージを見つけます。このメッセージには、`TraceRoute` PCF グループに適切なパラメーターが保持されている必要があります。この後、メッセージを取得してアクティビティ情報を分析します。

このタスクについて

アクティビティ情報を経路トレース・メッセージから取得できるのは、経路トレース・メッセージの場所がわかっていて、その経路トレース・メッセージの `TraceRoute` PCF グループにパラメーター `Accumulate` があり、`MQRROUTE_ACCUMULATE_IN_MSG` または `MQRROUTE_ACCUMULATE_AND_REPLY` に設定されている場合のみです。

ターゲット・キューに送信される経路トレース・メッセージの場合、`TraceRoute` PCF グループの `Deliver` パラメーターは `MQRROUTE_DELIVER_YES` に設定されていることが必要です。

手順

1. ターゲット・キューを調べます。経路トレース・メッセージがターゲット・キューにない場合は、アクティビティ・レコーディングで使用可能になっている経路トレース・メッセージを使用して、経路トレース・メッセージの場所を探索することができます。生成されたアクティビティ・レポートを使用して、経路トレース・メッセージが最後に確認された場所を特定してください。
2. 経路トレース・メッセージの取得
3. IBM MQ 経路表示アプリケーションを使用して、記録されたアクティビティ情報を表示します。
4. アクティビティ情報を調べて、必要な情報を入手します。

アクティビティ・レポートからの情報の取得

アクティビティ情報を取得するには、アクティビティ・レポートを見つけます。このレポートのメッセージ記述子には、レポート・オプションが指定されている必要があります。この後、アクティビティ・レポートを取得してアクティビティ情報を分析します。

このタスクについて

アクティビティ情報をアクティビティ・レポートから取得できるのは、アクティビティ・レポートの場所がわかっていて、レポート・オプション MQRO_ACTIVITY が、経路トレース・メッセージのメッセージ記述子に指定されている場合のみです。

手順

1. 経路トレース・メッセージに生成されたアクティビティ・レポートを見つけて、正しい順序に並べ替えます。
アクティビティ・レポートを見つけたら、手動で並べ替えることができ、また IBM MQ 経路表示アプリケーションを使用して、アクティビティ情報を自動的に並べ替えて表示することもできます。
2. アクティビティ情報を調べて、必要な情報を入手します。

追加のアクティビティ情報

経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信される時、ユーザー・アプリケーションは、Activity グループを経路トレース・メッセージまたはアクティビティ・レポートのメッセージ・データに書き込む際に、1 つ以上の追加 PCF パラメーターを含めることによって追加情報を記録することができます。

追加のアクティビティ情報は、経路トレース・メッセージがとった経路や、その経路をとった理由をシステム管理者が特定するために役立ちます。

IBM MQ 経路表示アプリケーションを使用して経路トレース・メッセージについて記録された情報を表示する場合、各パラメーターのパラメーター ID が IBM MQ 経路表示アプリケーションによって認識されていなければ、追加の PCF パラメーターに伴って表示されるのは数値 ID だけです。パラメーター ID を認識するには、以下の PCF パラメーターを使用して追加情報を記録する必要があります。これらの PCF パラメーターを、Activity PCF グループ内の適切な位置に組み込んでください。

GroupName

表 13. グループ名	
説明	追加情報を指定するグループ化されたパラメーター。
ID	MQGACF_VALUE_NAMING
データ・タイプ	MQCFGR
グループ内のパラメーター	<i>ParameterName</i> <i>ParameterValue</i>

ParameterName

表 14. パラメーター名	
説明	IBM MQ 経路表示アプリケーションで表示される名前が含まれます。このアプリケーションにより、 <i>ParameterValue</i> の値が取り込まれます。
ID	MQCA_VALUE_NAME
データ・タイプ	MQCFST
組み込まれる PCF グループ:	<i>GroupName</i> .
値:	表示される名前。

ParameterValue

表 15. パラメーター値	
説明	IBM MQ 経路表示アプリケーションで表示される値が含まれます。
ID:	追加情報の PCF 構造 ID。
データ型:	追加情報の PCF 構造データ型。
組み込まれる PCF グループ:	GroupName.
値:	表示される値。

追加アクティビティー情報の記録例

以下の各例では、経路トレース・メッセージに対してアクティビティーが実行されるときにユーザー・アプリケーションで追加情報を記録する方法を示しています。いずれの例でも、IBM MQ 経路表示アプリケーションを使用して経路トレース・メッセージを生成し、返されたアクティビティー情報を表示します。

追加のアクティビティー情報の記録: 例 1

追加のアクティビティー情報は、パラメーター ID が IBM MQ 経路表示アプリケーションによって認識されない形式で、ユーザー・アプリケーションによって記録されます。

1. IBM MQ 経路表示アプリケーションを使用して経路トレース・メッセージを生成し、キュー・マネージャー・ネットワークに入れます。以下を要求するために必要なオプションが設定されます。
 - 経路トレース・メッセージのメッセージ・データにアクティビティー情報を累積します。
 - ターゲット・キューへ到着したときに経路トレース・メッセージを廃棄し、経路トレース応答メッセージを生成して指定した 応答先キューへ送信します。
 - 経路トレース応答メッセージを受信すると、IBM MQ 経路表示アプリケーションは累積されたアクティビティー情報を表示します。

経路トレース・メッセージがキュー・マネージャー・ネットワークに入れられます。

2. 経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信されると、経路トレース・メッセージングで使用可能になっているユーザー・アプリケーションが、メッセージに対して詳細度の低いアクティビティーを実行します。標準的なアクティビティー情報を経路トレース・メッセージに書き込むのに加えて、ユーザー・アプリケーションは以下の PCF パラメーターを Activity グループの末尾に書き込みます。

ColorValue

ID

65536

データ・タイプ

MQCFST

値

'Red'

この追加 PCF パラメーターは、実行されたアクティビティーに関する詳細情報を提供しますが、パラメーター ID が IBM MQ 経路表示アプリケーションによって認識されない形式で書き込まれます。

3. 経路トレース・メッセージがターゲット・キューに到着し、経路トレース応答メッセージが IBM MQ 経路表示アプリケーションに返されます。追加のアクティビティー情報は以下のように表示されます。

```
65536: 'Red'
```

IBM MQ 経路表示アプリケーションは PCF パラメーターのパラメーター ID を認識せず、これを数値として表示します。追加情報の内容は明確には分かりません。

IBM MQ 経路表示アプリケーションが PCF パラメーターのパラメーター ID を認識する場合の例については、81 ページの『追加のアクティビティー情報の記録: 例 2』を参照してください。

追加のアクティビティー情報の記録: 例 2

パラメーター ID が IBM MQ 経路表示アプリケーションによって認識される形式で、追加のアクティビティー情報がユーザー・アプリケーションにより記録されます。

1. IBM MQ 経路表示アプリケーションを使用して、80 ページの『追加のアクティビティー情報の記録: 例 1』と同じ方法で経路トレース・メッセージを生成し、キュー・マネージャー・ネットワークに入れます。
2. 経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信されると、経路トレース・メッセージングで使用可能になっているユーザー・アプリケーションが、メッセージに対して詳細度の低いアクティビティーを実行します。標準的なアクティビティー情報を経路トレース・メッセージに書き込むのに加えて、ユーザー・アプリケーションは以下の PCF パラメーターを Activity グループの末尾に書き込みます。

ColorInfo

表 16. 色情報	
説明	色についての情報を指定するグループ化されたパラメーター。
ID:	MQGACF_VALUE_NAMING
データ型:	MQCFGR
グループ内のパラメーター:	<i>ColorName</i> <i>ColorValue</i>

ColorName

表 17. 色の名前	
説明	IBM MQ 経路表示アプリケーション で表示される名前が含まれます。このアプリケーションにより、 ColorValue の値が取り込まれます。
ID:	MQCA_VALUE_NAME
データ型:	MQCFST
組み込まれる PCF グループ:	<i>ColorInfo</i> .
値:	'Color'

ColorValue

表 18. 色の値	
説明	IBM MQ 経路表示アプリケーション で表示される値が含まれます。
ID:	65536
データ型:	MQCFST
組み込まれる PCF グループ:	<i>ColorInfo</i> .
値:	'Red'

これらの追加 PCF パラメーターは、実行されたアクティビティーについての追加情報を提供します。これらの PCF パラメーターは、パラメーター ID が IBM MQ 経路表示アプリケーションによって認識される形式で書かれています。

3. 経路トレース・メッセージがターゲット・キューに到着し、経路トレース応答メッセージが IBM MQ 経路表示アプリケーションに返されます。追加のアクティビティ情報は以下のように表示されます。

```
Color: 'Red'
```

IBM MQ 経路表示アプリケーションは、追加アクティビティ情報の値を含む PCF 構造のパラメーター ID に対応する名前があることを認識します。対応する名前が、数値の代わりに表示されます。

IBM MQ 経路表示アプリケーション

コマンド行インターフェースで、IBM MQ 経路表示アプリケーション (**dspmqrte**) を使用して、経路トレース・メッセージと経路トレース・メッセージに関連するアクティビティ情報を処理します。

IBM MQ 経路表示アプリケーションは、IBM MQ for z/OS では出荷されませんが、**dspmqrte** コマンドを発行するときに **-c** パラメーターを指定することにより、分散インストール済み環境から実行し、クライアントとして IBM MQ for z/OS キュー・マネージャーに接続することができます。

IBM MQ **dspmqrte** 経路表示アプリケーションは、以下の目的で使用できます。

- 経路トレース・メッセージを構成および生成し、これをキュー・マネージャー・ネットワークに入れます。

経路トレース・メッセージをキュー・マネージャー・ネットワークに入れることにより、アクティビティ情報を収集し、これを経路トレース・メッセージがとった経路の判別に使用できます。経路トレース・メッセージの特性は以下のように指定できます。

- 経路トレース・メッセージの宛先。
- 経路トレース・メッセージが別のメッセージを模倣する方法。
- 経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信されたときの処理方法。
- アクティビティ情報の記録にアクティビティ・レコーディングまたは経路トレース・メッセージングのどちらを使用するか。

- 経路トレース・メッセージに関連するアクティビティ情報を正しく並べ替えて表示します。

関連するアクティビティ情報が返された後で IBM MQ 経路表示アプリケーションが経路トレース・メッセージをキュー・マネージャー・ネットワークに入れた場合、その情報は直ちに並べ替えて表示することができます。また、IBM MQ 経路表示アプリケーションを使用して、以前に生成された経路トレース・メッセージに関連するアクティビティ情報を並べ替えて表示することも可能です。

関連資料

[dspmqrte](#)

経路トレース・メッセージのパラメーター

このページを使用して、IBM MQ 経路表示アプリケーション **dspmqrte** で提供されているパラメーターの概要を取得し、経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信される際の処理方法を含む、経路トレース・メッセージの特性を決定します。

関連資料

[dspmqrte](#)

キュー・マネージャー接続

このページを使用して、IBM MQ 経路表示アプリケーションの接続先キュー・マネージャーを指定します。

-c

IBM MQ 経路表示アプリケーションがクライアント・アプリケーションとして接続することを指定します。

このパラメーターを指定しない場合、IBM MQ 経路表示アプリケーションはクライアント・アプリケーションとして接続されません。

-m QMgrName

IBM MQ 経路表示アプリケーションの接続先となるキュー・マネージャーの名前。名前の長さは 48 文字までです。

このパラメーターを指定しない場合は、デフォルトのキュー・マネージャーが使用されます。

ターゲット宛先

このページを使用して、経路トレース・メッセージのターゲット宛先を指定します。

-q TargetQName

経路トレース・メッセージをキュー・マネージャー・ネットワークに送信するために IBM MQ 経路表示アプリケーションが使用されている場合は、*TargetQName* でターゲット・キューの名前を指定します。

-ts TargetTopicString

トピック・ストリングを指定します。

-qm TargetQMGr

ターゲット宛先の名前を修飾します。通常のキュー・マネージャーのネーム・レゾリューションが適用されます。ターゲット宛先は *-q TargetQName* または *-ts TargetTopicString* で指定します。

このパラメーターを指定しない場合は、IBM MQ 経路表示アプリケーションが接続しているキュー・マネージャーがターゲット・キュー・マネージャーとして使用されます。

-o

ターゲット宛先が特定の宛先にバインドされないことを指定します。通常このパラメーターは、経路トレース・メッセージをクラスター全体に書き込むときに使用されます。ターゲット宛先は、オプション *MQOO_BIND_NOT_FIXED* でオープンされます。

このパラメーターを指定しない場合、ターゲット宛先は特定の宛先にバインドされます。

パブリケーション・トピック

パブリッシュ/サブスクライブ・アプリケーションの場合、このページを使用して、IBM MQ 経路表示アプリケーションでパブリッシュするための経路トレース・メッセージのトピック・ストリングを指定します。

-ts TopicName

IBM MQ 経路表示アプリケーションが経路トレース・メッセージをパブリッシュする宛先となるトピック・ストリングを指定して、このアプリケーションをトピック・モードに設定します。このモードでは、そのアプリケーションは、パブリッシュ要求の結果として得られるすべてのメッセージをトレースします。

また、IBM MQ 経路表示アプリケーションを使用して、パブリッシュ・メッセージに対して生成されたアクティビティ・レポートから結果を表示することもできます。

メッセージの模倣

例えば、元のメッセージがその意図された宛先に届かなかった場合などに、このページを使用して、メッセージを模倣するように経路トレース・メッセージを構成します。

経路トレース・メッセージングの用途の 1 つに、意図された宛先に届かなかったメッセージが最後に確認された場所の特定を支援するというものがあります。IBM MQ 経路表示アプリケーションには、経路トレース・メッセージを元のメッセージに似せて構成できるパラメーターが用意されています。メッセージを模倣するときは、以下のパラメーターを使用できます。

-l Persistence

生成された経路トレース・メッセージの持続性を指定します。*Persistence* の可能な値は、次のとおりです。

yes

生成される経路トレース・メッセージは持続します (MQPER_PERSISTENT)。

no

生成する経路トレース・メッセージは持続しません (MQPER_NOT_PERSISTENT)。

q

生成される経路トレース・メッセージは、その持続性値を、*-q TargetQName* または *-ts TargetTopicString* で指定された宛先から継承します。(MQPER_PERSISTENCE_AS_Q_DEF)。

戻された経路トレース応答メッセージまたはレポート・メッセージは、元の経路トレース・メッセージと同じ持続性を共有します。

Persistence を **yes** と指定する場合は、*-rq ReplyToQ* パラメーターを指定する必要があります。応答先キューは一時動的キューに解決できません。

このパラメーターを指定しない場合は、生成された経路トレース・メッセージが持続しません。

-p Priority

経路トレース・メッセージの優先順位を指定します。Priority の値は、0 以上かまたは MQPRI_PRIORITY_AS_Q_DEF です。MQPRI_PRIORITY_AS_Q_DEF は、*-q TargetQName* または *-ts TargetTopicString* で指定された宛先から優先度の値を取得することを指定します。

このパラメーターを指定しない場合、優先度の値は、*-q TargetQName* または *-ts TargetTopicString* で指定された宛先から取得されます。

-xs Expiry

トレース・メッセージの有効期限を秒単位で指定します。

このパラメーターを指定しない場合は、有効期限時間が 60 秒に指定されます。

-ro none |ReportOption

none

レポート・オプションを設定しないように指定します。

ReportOption

経路トレース・メッセージのレポート・オプションを指定します。コンマを分離文字として使用すると、複数のレポート・オプションを指定できます。ReportOption の可能な値は、次のとおりです。

アクティビティー

レポート・オプション MQRO_ACTIVITY を設定します。

coa

レポート・オプション MQRO_COA_WITH_FULL_DATA を設定します。

cod

レポート・オプション MQRO_COD_WITH_FULL_DATA を設定します。

exception

レポート・オプション MQRO_EXCEPTION_WITH_FULL_DATA を設定します。

expiration

レポート・オプション MQRO_EXPIRATION_WITH_FULL_DATA を設定します。

discard

レポート・オプション MQRO_DISCARD_MSG を設定します。

-ro ReportOption も *-ro none* も指定しない場合は、MQRO_ACTIVITY および MQRO_DISCARD_MSG レポート・オプションが指定されます。

IBM MQ 経路表示アプリケーションでは、経路トレース・メッセージにユーザー・データを追加することはできません。経路トレース・メッセージにユーザー・データを追加する必要がある場合は、経路トレース・メッセージを手動で生成する必要があります。

記録されたアクティビティー情報

このページを使用して、記録されたアクティビティー情報を返すために使用する方法を指定します。このアクティビティー情報は、次に経路トレース・メッセージが通過した経路を判別するために使用できます。

記録されたアクティビティー情報は、以下を使用して返すことができます。

- アクティビティー・レポート。
- 経路トレース応答メッセージ。
- 経路トレース・メッセージ自体 (ターゲット・キューに入れられたメッセージ)。

dspmqrte を使用するとき、記録されたアクティビティー情報を返すために使用する方法は、以下のパラメーターを使用して決定されます。

-ro activity

アクティビティー情報をアクティビティー・レポートを使用して返すことを指定します。デフォルトでは、アクティビティー・レコーディングは使用可能に設定されています。

-ac -ar

アクティビティー情報を経路トレース・メッセージに累積し、経路トレース応答メッセージを生成することを指定します。

-ac

アクティビティー情報が経路トレース・メッセージ内に累積されるように指定します。

このパラメーターを指定しない場合は、アクティビティー情報は経路トレース・メッセージ内に累積されません。

-ar

次の事情において、累積されたすべてのアクティビティー情報を含んだ経路トレース応答メッセージを生成するよう要求します。

- 経路トレース・メッセージが IBM MQ キュー・マネージャーによって廃棄された。
- 経路トレース・メッセージが IBM MQ キュー・マネージャーによってローカル・キュー (ターゲット・キューまたは送達不能キュー) に入れられた。
- 経路トレース・メッセージで行われたアクティビティーの数が、**-s Activities** に指定されている値を超えた。

-ac -d yes

アクティビティー情報を経路トレース・メッセージに累積し、到着時に経路トレース・メッセージをターゲット・キューに入れることを指定します。

-ac

アクティビティー情報が経路トレース・メッセージ内に累積されるように指定します。

このパラメーターを指定しない場合は、アクティビティー情報は経路トレース・メッセージ内に累積されません。

-d yes

キュー・マネージャーが経路トレース・メッセージングをサポートしていない場合でも、経路トレース・メッセージは、到着するとターゲット・キューに書き込まれます。

このパラメーターを指定しない場合、経路トレース・メッセージはターゲット・キューに入れられません。

経路トレース・メッセージはターゲット・キューから取り出すことができ、記録されたアクティビティー情報を得ることができます。

これらの方法は必要に応じて組み合わせることができます。

また、記録されるアクティビティー情報の詳細レベルは、以下のパラメーターを使用して指定できます。

-t Detail

記録されるアクティビティーを指定します。 *Detail* の可能な値は、次のとおりです。

low

ユーザー定義アプリケーションによって実行されるアクティビティーのみが記録されます。

ミディアム

low で指定されているアクティビティーが記録されます。さらに、パブリッシュ・アクティビティーおよび MCA で実行されたアクティビティーも記録されます。

high

low および **medium** で指定されたアクティビティーが記録されます。この詳細レベルでは、詳細なアクティビティー情報が MCA によって公開されることはありません。このオプションが使用可能なのは、詳細なアクティビティー情報を公開するユーザー定義アプリケーションに限定されます。例えば、ユーザー定義アプリケーションが、特定のメッセージ特性を考慮することによってメッセージの経路を決定する場合は、この詳細レベルで経路指定ロジックを組み込むことができます。

このパラメーターを指定しない場合は、中間レベルのアクティビティーが記録されます。

デフォルトでは、IBM MQ 経路表示アプリケーションは返されたメッセージを、一時動的キューを使用して保管します。IBM MQ 経路表示アプリケーションが終了すると、この一時動的キューはクローズされ、メッセージはすべて消去されます。現在の IBM MQ 経路表示アプリケーションの実行が終了した後も、返されたメッセージが必要な場合は、以下のパラメーターを使用して永続キューを指定する必要があります。

-rq ReplyToQ

経路トレース・メッセージへのすべての応答の送信先となる応答先キューの名前を指定します。経路トレース・メッセージが持続的であるか、または *-n* パラメーターを指定した場合は、一時動的キュー以外の応答先キューを指定する必要があります。

このパラメーターを指定しない場合は、システム・デフォルト・モデル・キュー SYSTEM.DEFAULT.MODEL.QUEUE を使用して 動的な応答先キューが作成されます。

-rqm ReplyToQMgr

応答先キューが存在するキュー・マネージャーの名前を指定します。名前の長さは 48 文字までです。

このパラメーターを指定しない場合は、IBM MQ 経路表示アプリケーションの接続先となるキュー・マネージャーが応答先キュー・マネージャーとして使用されます。

経路トレース・メッセージの処理方法

このページを使用して、経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信される際の処理方法を制御します。

以下のパラメーターは、キュー・マネージャー・ネットワーク内の経路トレース・メッセージが送信される場所を制限することができます。

-d Deliver

経路トレース・メッセージを到着時にターゲット・キューに配信するかどうかを指定します。 *Deliver* の可能な値は、次のとおりです。

yes

キュー・マネージャーが経路トレース・メッセージングをサポートしていない場合でも、経路トレース・メッセージは、到着するとターゲット・キューに書き込まれます。

no

到着時に、経路トレース・メッセージはターゲット・キューに入れられません。

このパラメーターを指定しない場合、経路トレース・メッセージはターゲット・キューに入れられません。

-f Forward

経路トレース・メッセージを転送できる先のキュー・マネージャーのタイプを指定します。メッセージをリモート・キュー・マネージャーに転送するかどうかを決定するために、キュー・マネージャーが使用するアルゴリズムについて詳しくは、74 ページの『[TraceRoute PCF グループ](#)』を参照してください。 *Forward* の可能な値は、次のとおりです。

すべて

経路トレース・メッセージは任意のキュー・マネージャーに転送されます。

警告 : IBM WebSphere MQ 6.0 より前のキュー・マネージャーに転送された場合、経路トレース・メッセージは認識されず、*-d Deliver* パラメーターの値にかかわらず、ローカル・キューに送信される可能性があります。

サポート済み

経路トレース・メッセージは、*TraceRoute PCF* グループからの *Deliver* パラメーターを認識するキュー・マネージャーにのみ転送されます。

このパラメーターを指定しない場合は、経路トレース・メッセージが、*Deliver* パラメーターを認識するキュー・マネージャーにのみ転送されます。

以下のパラメーターは、経路トレース・メッセージがキュー・マネージャー・ネットワークに無期限に残るのを防ぐことができます。

-s Activities

経路トレース・メッセージが廃棄される前に、このメッセージの代わりに実行できる記録済みのアクティビティの最大数を指定します。これにより、経路トレース・メッセージが無限ループに入ってし

まった場合、メッセージが無制限に転送されることを防止できます。Activities の値は、1 以上かまたは MQROUTE_UNLIMITED_ACTIVITIES です。MQROUTE_UNLIMITED_ACTIVITIES は、経路トレース・メッセージの代わりにアクティビティーを無制限に実行できるように指定します。

このパラメーターを指定しない場合は、経路トレース・メッセージの代わりにアクティビティーを無制限に実行できます。

-xs Expiry

トレース・メッセージの有効期限を秒単位で指定します。

このパラメーターを指定しない場合は、有効期限時間が 60 秒に指定されます。

-xp PassExpiry

経路トレース・メッセージの有効期間を経路トレース応答メッセージに渡すかどうかを指定します。PassExpiry の可能な値は、次のとおりです。

yes

レポート・オプション MQRO_PASS_DISCARD_AND_EXPIRY が経路トレース・メッセージのメッセージ記述子で指定されます。

経路トレース応答メッセージまたはアクティビティー・レポートが、経路トレース・メッセージについて生成される場合は、MQRO_DISCARD レポート・オプション (指定されている場合) および残りの有効期間が渡されます。

これがデフォルト値です。

no

レポート・オプション MQRO_PASS_DISCARD_AND_EXPIRY は指定されません。

経路トレース・メッセージについて経路トレース応答メッセージが生成される場合、経路トレース・メッセージの廃棄オプションと有効期間は渡されません。

このパラメーターを指定しない場合、MQRO_PASS_DISCARD_AND_EXPIRY は指定されません。

-ro discard

MQRO_DISCARD_MSG レポート・オプションを指定します。これにより、経路トレース・メッセージがキュー・マネージャー・ネットワークに無期限に残るのを防ぐことができます。

アクティビティー情報の表示

IBM MQ 経路表示アプリケーションにより、キュー・マネージャー・ネットワークに入れられたばかりの、または以前に生成された経路トレース・メッセージのアクティビティー情報を表示できます。また、ユーザー作成アプリケーションによって記録された追加情報を表示することもできます。

経路トレース・メッセージについて返されたアクティビティー情報を表示するかどうかを指定するには、以下のパラメーターを指定します。

-n

経路トレース・メッセージに返されるアクティビティー情報を表示しないように指定します。

このパラメーターに、経路トレース応答メッセージ (*-ar*) の要求、またはレポート生成オプション (*-ro ReportOption*) のいずれかが伴う場合は、*-rq ReplyToQ* を使用して特定の (非モデル) 応答先キューを指定する必要があります。デフォルトでは、アクティビティー・レポート・メッセージのみが要求されます。

指定されたターゲット・キューに経路トレース・メッセージが入れると、経路トレース・メッセージのメッセージ ID を含む 48 文字の 16 進数ストリングが表示されます。このメッセージ ID が IBM MQ 経路表示アプリケーションで使用されて、*-i CorrelId* パラメーターを使用して、後で経路トレース・メッセージのアクティビティー情報が表示されます。

このパラメーターを指定しない場合は、経路トレース・メッセージとして返されるアクティビティー情報が、*-v* パラメーターによって指定される形式で表示されます。

キュー・マネージャー・ネットワークに入れられたばかりの経路トレース・メッセージのアクティビティー情報を表示するには、以下のパラメーターを指定できます。

-w WaitTime

IBM MQ 経路表示アプリケーションが、指定された応答先キューに戻るまでに、アクティビティ報告書または経路トレース応答メッセージを待つときの時間を秒単位で指定します。

このパラメーターを指定しない場合は、待ち時間が、経路トレース・メッセージの有効期限時間に 60 秒を加えて指定されます。

以前に累積されたアクティビティ情報を表示する際には、以下のパラメーターを指定する必要があります。

-q TargetQName

IBM MQ 経路表示アプリケーションを使用して、前に収集済みのアクティビティ情報を表示する場合は、アクティビティ情報の格納先キューの名前を *TargetQName* に指定します。

-i CorrelId

このパラメーターを使用するのは、IBM MQ 経路表示アプリケーションを使用して、以前に累積したアクティビティ情報のみを表示する場合です。 *-q TargetQName* によって指定されるキューには、多くのアクティビティ報告書と経路トレース応答メッセージが存在する可能性があります。 *CorrelId* は、経路トレース・メッセージに関連したアクティビティ報告書または経路トレース応答メッセージを識別するために使用します。元の経路トレース・メッセージのメッセージ ID を *CorrelId* に指定します。

CorrelId の形式は、48 文字の 16 進数ストリングです。

以前に累積したアクティビティ情報を表示するとき、または経路トレース・メッセージの現在のアクティビティ情報を表示するときには、以下のパラメーターを使用できます。

-b

IBM MQ 経路表示アプリケーションが、メッセージに関連したアクティビティ報告書または経路トレース応答メッセージのみを表示するように指定します。このオプションを指定すると、後でもう一度アクティビティ情報を表示できます。

このパラメーターを指定しない場合は、IBM MQ 経路表示アプリケーションが、メッセージに関連するアクティビティ報告書または経路トレース応答メッセージを破壊的に取得します。

-v summary | all | none | outline DisplayOption

要約

経路指定された経路トレース・メッセージが表示されるキュー。

すべて

使用可能なすべての情報が表示されます。

なし

情報は表示されません。

outline DisplayOption

経路トレース・メッセージの表示オプションを指定します。コンマを分離文字として使用すると、複数の表示オプションを指定できます。

値が提供されない場合には、次の情報が表示されます。

- アプリケーション名。
- 各オペレーションのタイプ。
- オペレーションに固有のパラメーター

DisplayOption の可能な値は、次のとおりです。

アクティビティ

Activity PCF グループのすべての非 *PCF* グループ・パラメーターが表示されます。

ID

パラメーター識別子 *MQBACF_MSG_ID* または *MQBACF_CORREL_ID* による値が表示されます。これにより、*msgdelta* を指定変更します。

メッセージ

Message PCF グループのすべての非 *PCF* グループ・パラメーターが表示されます。この値を指定する場合は、*msgdelta* を指定できません。

msgdelta

最後のオペレーション以後に変更されている Message PCF グループのすべての非 PCF グループ・パラメーターが表示されます。この値を指定する場合は、*message* を指定できません。

operation

Operation PCF グループのすべての非 PCF グループ・パラメーターが表示されます。

traceroute

TraceRoute PCF グループのすべての非 PCF グループ・パラメーターが表示されます。

このパラメーターを指定しない場合は、メッセージ経路の要約が表示されます。

追加情報の表示

経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信される時、ユーザー作成アプリケーションは、経路トレース・メッセージまたはアクティビティ・レポートのメッセージ・データに1つ以上の追加 PCF パラメーターを書き込むことによって追加情報を記録することができます。読み取り可能な形式で IBM MQ 経路表示アプリケーションが追加情報を表示するには、その情報が、79 ページの『追加のアクティビティ情報』に詳述されている特定の形式で記録されていなければなりません。

IBM MQ 経路表示アプリケーションの例

次の例は、IBM MQ 経路表示アプリケーションの使用法を示しています。それぞれの例には、2つのチャネル (QM2.TO.QM1 および QM1.TO.QM2) で相互接続された2つのキュー・マネージャー (QM1 および QM2) があります。

例 1 - アクティビティ・レポートの要求

ターゲット・キューに送信された経路トレース・メッセージからアクティビティ情報を表示します。

この例では、IBM MQ 経路表示アプリケーションがキュー・マネージャー QM1 に接続しており、このアプリケーションを使用して経路トレース・メッセージを生成し、リモート・キュー・マネージャー QM2 にあるターゲット・キュー TARGET.Q に送信します。経路トレース応答メッセージの送信時にアクティビティ・レポートが要求されるように、必要なレポート・オプションが指定されています。ターゲット・キューに到着すると、経路トレース・メッセージは廃棄されます。アクティビティ・レポートを使用して IBM MQ 経路表示アプリケーションに返されたアクティビティ情報が、正しい順序に並べ替えられ、表示されます。

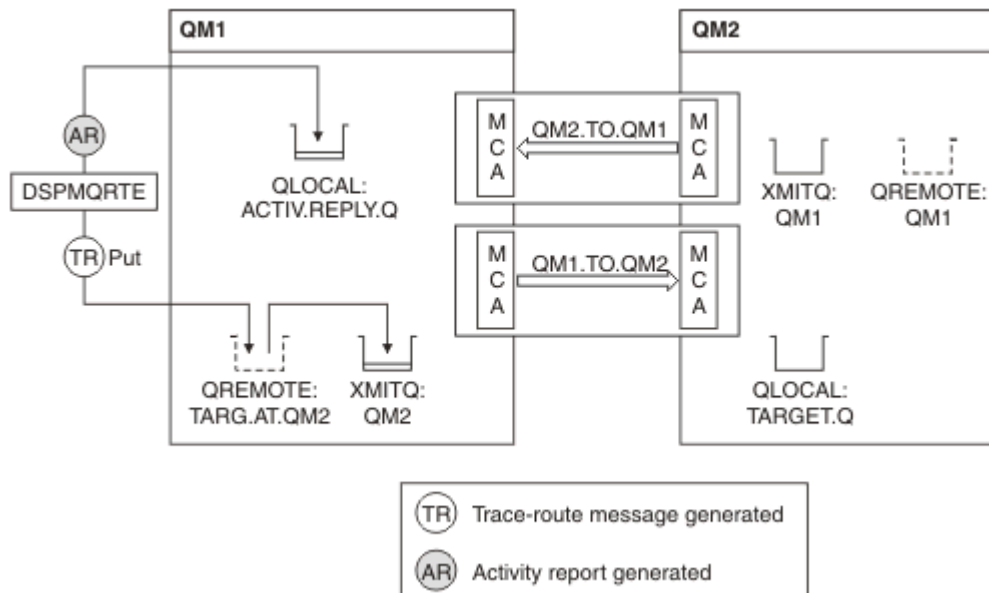


図 9. アクティビティ・レポートの要求、図 1

- 各キュー・マネージャー (QM1 および QM2) の ACTIVREC 属性は MSG に設定されています。
- 次のコマンドが発行されます。


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

QM1 は IBM MQ 経路表示アプリケーションが接続するキュー・マネージャーの名前、TARG.AT.QM2 はターゲット・キューの名前、ACTIV.REPLY.Q は経路トレース・メッセージに対するすべての応答が送信されるように要求されている送信先キューの名前です。

指定されていないすべてのオプションについてデフォルト値が想定されていますが、特に -f オプション (TraceRoute PCF グループの Deliver パラメーターを受け入れるキュー・マネージャーにのみ経路トレース・メッセージを転送)、-d オプション (到着時に経路トレース・メッセージをターゲット・キューに入れない)、-ro オプション (MQRO_ACTIVITY および MQRO_DISCARD_MSG レポート・オプションを指定)、および -t オプション (中程度の詳細レベルのアクティビティを記録) には注意してください。

- DSPMQRTE は経路トレース・メッセージを生成し、これをリモート・キュー TARG.AT.QM2 に入れます。
- DSPMQRTE は次に、キュー・マネージャー QM1 の ACTIVREC 属性の値を調べます。この値は MSG であるため、DSPMQRTE はアクティビティ・レポートを生成し、これを応答キュー ACTIV.REPLY.Q に入れます。

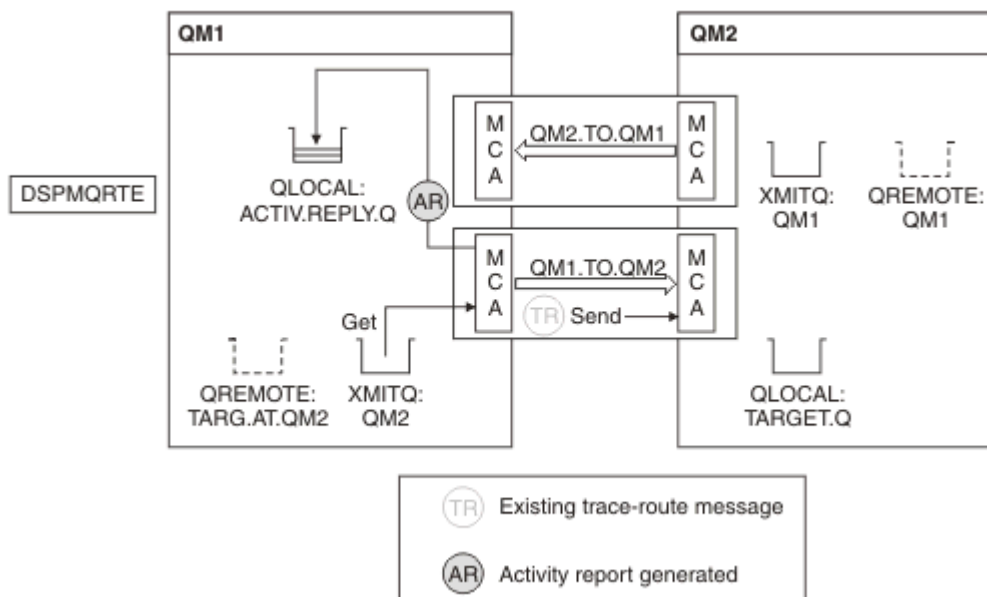


図 10. アクティビティ・レポートの要求、図 2

- 送信側メッセージ・チャンネル・エージェント (MCA) が伝送キューから経路トレース・メッセージを読み取ります。このメッセージは経路トレース・メッセージであるため、MCA はアクティビティ情報の記録を開始します。
- キュー・マネージャー (QM1) の ACTIVREC 属性は MSG であり、MQRO_ACTIVITY オプションがメッセージ記述子の Report フィールドで指定されているため、MCA は後でアクティビティ・レポートを作成します。TraceRoute PCF グループの RecordedActivities パラメーター値が 1 増加します。
- MCA は、TraceRoute PCF グループの MaxActivities 値を超過していないことを確認します。
- メッセージが QM2 に転送される前に、MCA は Forward (手順 76 ページの『1』、76 ページの『4』、および 77 ページの『5』) に記述されているアルゴリズムに従い、メッセージの送信を選択します。
- 次に MCA はアクティビティ・レポートを生成し、応答キュー (ACTIV.REPLY.Q) に入れます。

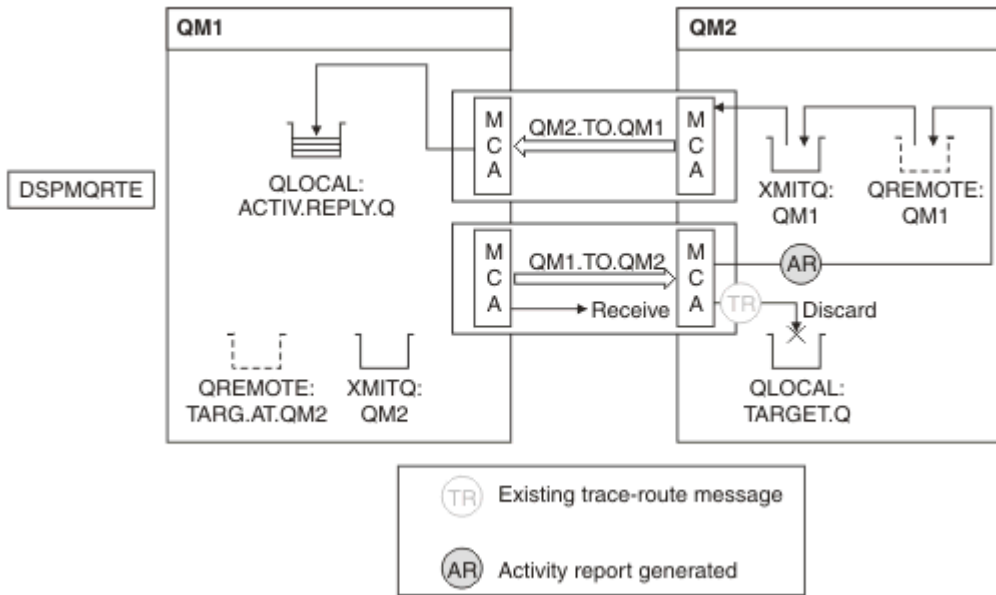


図 11. アクティビティ・レポートの要求、図 3

- 受信側 MCA は、チャンネルから経路トレース・メッセージを受け取ります。このメッセージは経路トレース・メッセージであるため、MCA はアクティビティに関する情報の記録を開始します。
- 経路トレース・メッセージの送信元キュー・マネージャーが IBM WebSphere MQ 5.3.1 以前のものであった場合、MCA は TraceRoute PCF の DiscontinuityCount パラメーターを 1 増加させます。この場合はそれに該当しません。
- キュー・マネージャー (QM2) の ACTIVREC 属性は MSG であり、MQRO_ACTIVITY オプションが指定されているため、MCA はアクティビティ・レポートを生成します。RecordedActivities パラメーターの値が 1 増加します。
- ターゲット・キューはローカル・キューであるため、メッセージは TraceRoute PCF グループの Deliver パラメーターの値を踏まえ、フィードバック MQFB_NOT_DELIVERED で廃棄されます。
- 次に MCA は最終アクティビティ・レポートを生成し、応答キューに入れます。これにより、キュー・マネージャー QM1 に関連付けられた伝送キューが解決され、アクティビティ・レポートがキュー・マネージャー QM1 (ACTIV.REPLY.Q) に返されます。

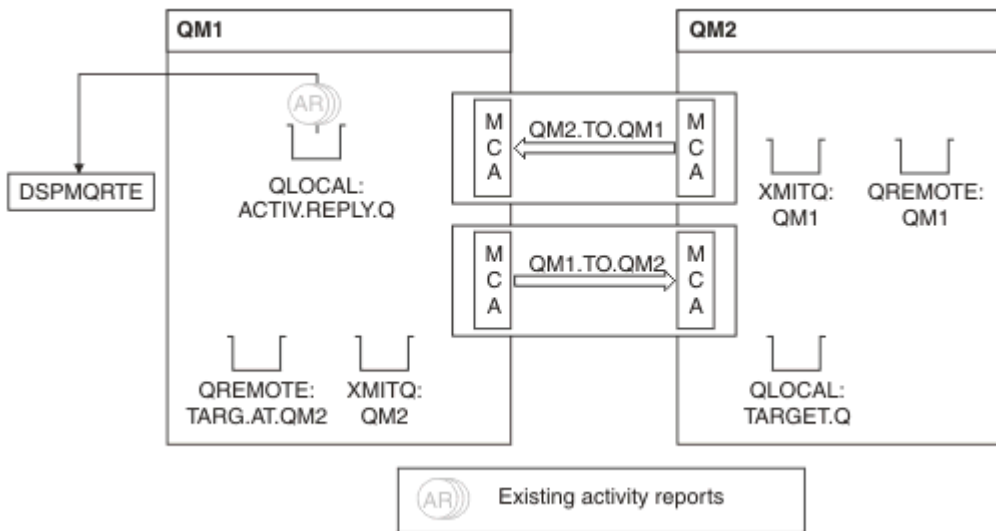


図 12. アクティビティ・レポートの要求、図 4

- その間に、DSPMQRTE はアクティビティー・レポートを待ちながら、応答キュー (ACTIV.REPLY.Q) に対する MQGET の実行を続けます。DSPMQRTE が開始されたときに -w が指定されなかったため、最大 120 秒 (経路トレース・メッセージの有効期間より 60 秒長い) まで待機します。
- DSPMQRTE は応答キューから 3 つのアクティビティー・レポートを取得します。
- アクティビティー・レポートは各アクティビティーの TraceRoute PCF グループの RecordedActivities、UnrecordedActivities、および DiscontinuityCount パラメーターを使用して正しい順序に並べ替えられます。この例でゼロ以外の値は RecordedActivities だけであるため、これが実際に使用される唯一のパラメーターです。
- プログラムは廃棄操作が表示されると直ちに終了します。最後の操作が廃棄であっても、フィードバックは MQFB_NOT_DELIVERED であるため、put が実行されたかのように処理されます。

表示される出力は以下のようになります。

```
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.
```

例 2 - 経路トレース応答メッセージの要求

経路トレース・メッセージを生成して、ターゲット・キューに送信します。

この例では、IBM MQ 経路表示アプリケーションがキュー・マネージャー QM1 に接続しており、このアプリケーションを使用して経路トレース・メッセージを生成し、リモート・キュー・マネージャー QM2 にあるターゲット・キュー TARGET.Q に送信します。アクティビティー情報が経路トレース・メッセージに累積されるように、必要なオプションが指定されています。ターゲット・キューに到着すると、経路トレース応答メッセージが要求され、経路トレース・メッセージは廃棄されます。

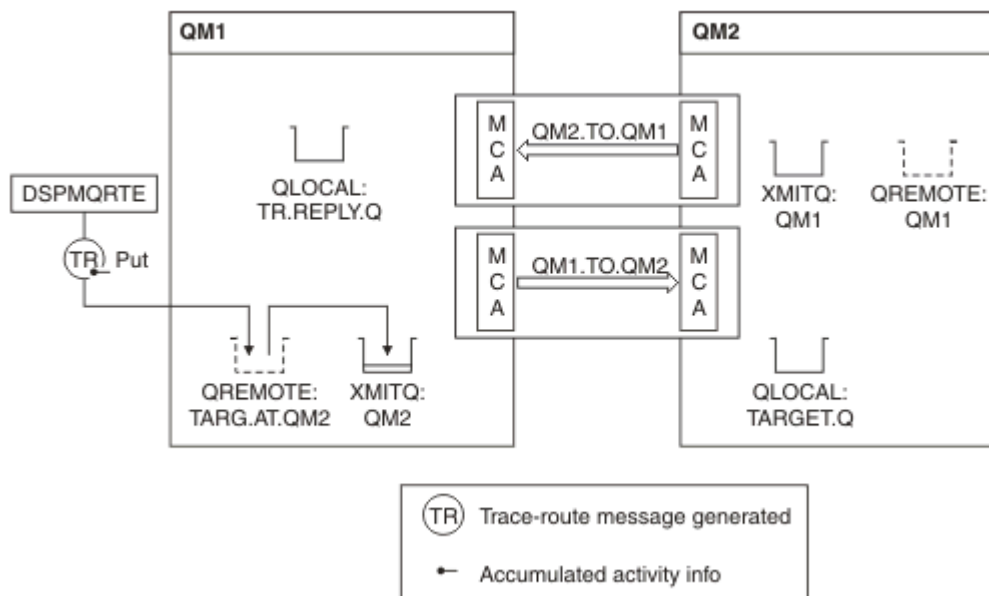


図 13. 経路トレース応答メッセージの要求、図 1

- 各キュー・マネージャー (QM1 および QM2) の ROUTEREC 属性は MSG に設定されています。
- 次のコマンドが発行されます。

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

QM1 は IBM MQ 経路表示アプリケーションが接続するキュー・マネージャーの名前、TARG.AT.QM2 はターゲット・キューの名前、ACTIV.REPLY.Q は経路トレース・メッセージに対するすべての応答が送信されるように要求されている送信先キューの名前です。-ac オプションはアクティビティー情報が経路トレース・メッセージに累積されることを指定し、-ar オプションは累積されたすべてのアクティビティーが、-rq オプションで指定された応答先キュー (TR.REPLY.Q) に送信されることを指定します。-ro オプションは、レポート・オプション MQRO_DISCARD_MSG を設定することを指定します。これは、アクティビティー・レポートがこの例では生成されないことを意味します。

- DSPMQRTE は、メッセージがターゲット経路に入れられる前に、アクティビティー情報を経路トレース・メッセージに累積します。これを行うためには、キュー・マネージャー属性 ROUTEREC を DISABLED に設定してはいけません。

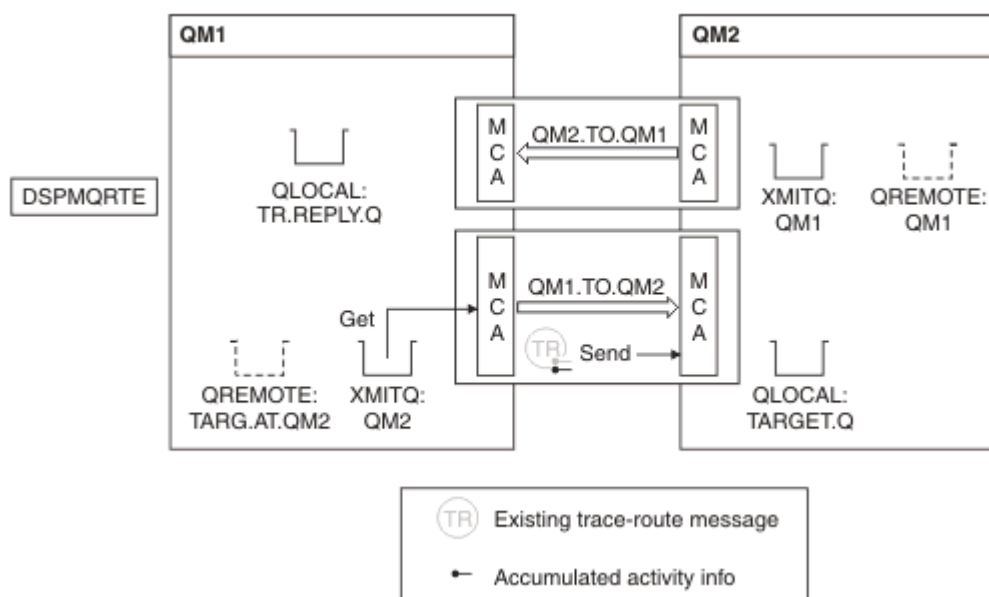


図 14. 経路トレース応答メッセージの要求、図 2

- このメッセージは経路トレース・メッセージであるため、送信側 MCA はアクティビティーに関する情報の記録を開始します。
- QM1 上のキュー・マネージャー属性 ROUTEREC は DISABLED ではないため、MCA はメッセージがキュー・マネージャー QM2 に転送される前に、メッセージ内にアクティビティー情報を累積します。

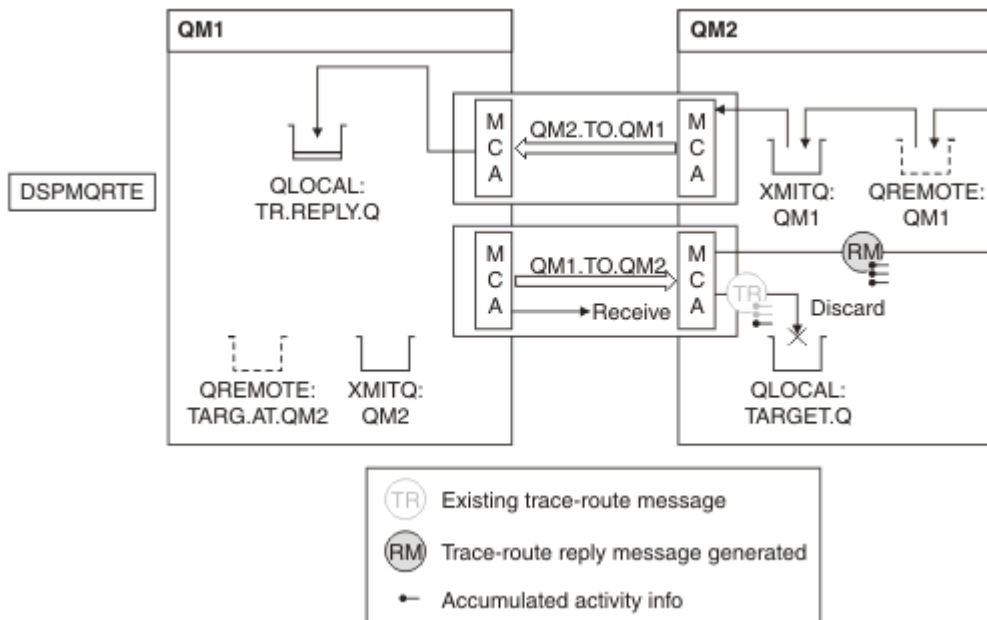


図 15. 経路トレース応答メッセージの要求、図 3

- このメッセージは経路トレース・メッセージであるため、受信側 MCA はアクティビティーに関する情報の記録を開始します。
- QM2 上のキュー・マネージャー属性 ROUTEREC は DISABLED ではないため、MCA はメッセージ内に情報を累積します。
- ターゲット・キューはローカル・キューであるため、メッセージは TraceRoute PCF グループの Deliver パラメーターの値を踏まえ、フィードバック MQFB_NOT_DELIVERED で廃棄されます。
- これは、メッセージで実行される最後のアクティビティーです。QM1 上のキュー・マネージャー属性 ROUTEREC が DISABLED ではないため、MCA は Accumulate 値に従って経路トレース応答メッセージを生成します。ROUTEREC の値は MSG であるため、応答メッセージは応答キューに入れられます。応答メッセージには経路トレース・メッセージから累積されたすべてのアクティビティー情報が含まれています。

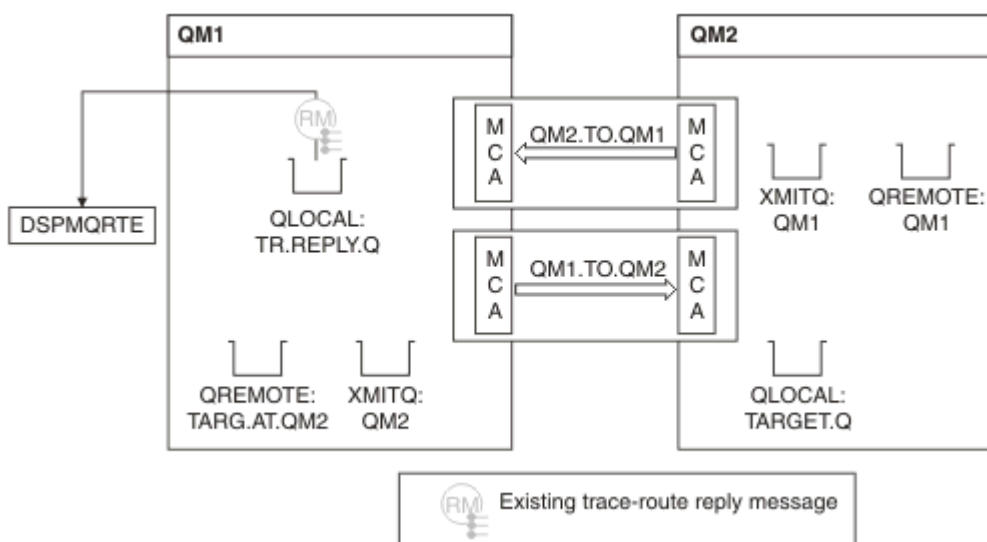


図 16. 経路トレース応答メッセージの要求、図 4

- この間、DSPMQRTE は経路トレース応答メッセージが応答キューに返されるのを待っています。応答メッセージが返されると、DSPMQRTE は含まれている各アクティビティを解析し、印刷します。最後の操作は廃棄操作です。印刷後、DSPMQRTE は終了します。

表示される出力は以下のようになります。

```
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.
```

例 3 - システム・キューへのアクティビティ・レポートの送信

アクティビティ・レポートが応答先キュー以外のキューに送信された時に検出し、IBM MQ 経路表示アプリケーションを使用して、他のキューからアクティビティ・レポートを読み取ります。

この例は、QM2 の ACTIVREC キュー・マネージャー属性の値が QUEUE に設定されていることを除けば、89 ページの『例 1 - アクティビティ・レポートの要求』と同じです。これを有効にするには、チャンネル QM1.TO.QM2 を再始動しておく必要があります。

この例は、応答先キュー以外のキューにアクティビティ・レポートが送信された場合の検出方法を示します。検出されると、アクティビティ・レポートを別のキューから読み取るために IBM MQ 経路表示アプリケーションが使用されます。

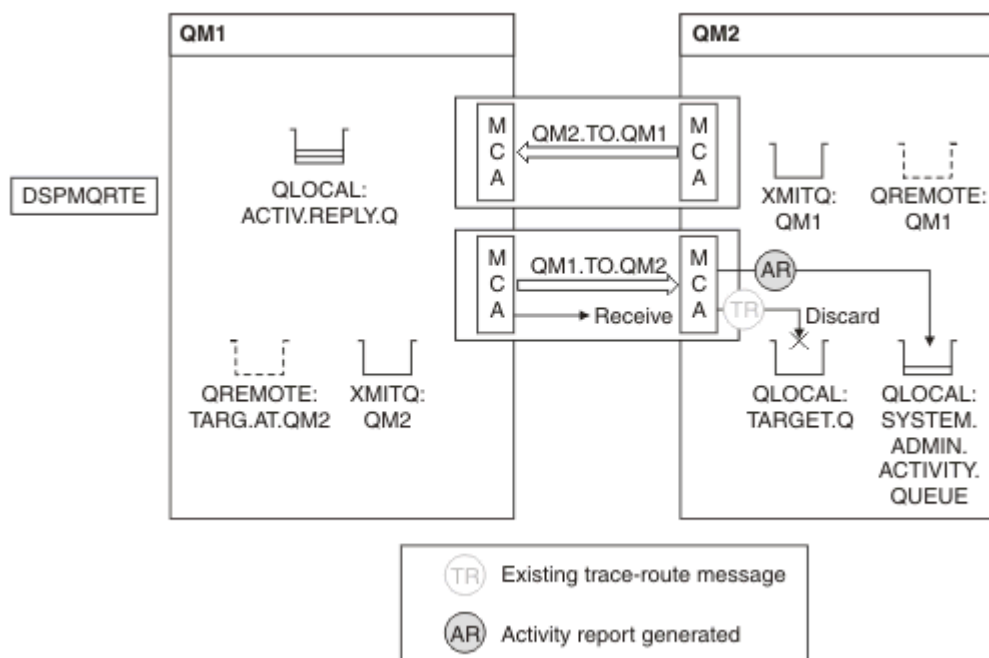


図 17. システム・キューへのアクティビティ・レポートの送信、図 1

- このメッセージは経路トレース・メッセージであるため、受信側 MCA はアクティビティに関する情報の記録を開始します。
- QM2 の ACTIVREC キュー・マネージャー属性の値が今回は QUEUE であるため、MCA はアクティビティ・レポートを生成しますが、これを応答キュー (ACTIV.REPLY.Q) ではなく、システム・キュー (SYSTEM.ADMIN.ACTIVITY.QUEUE) に入れます。

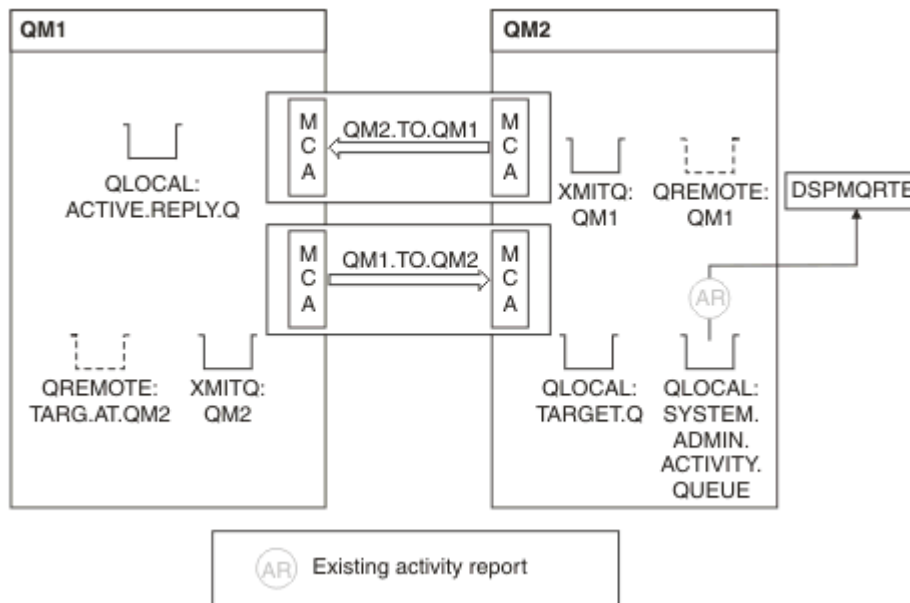


図 18. システム・キューへのアクティビティ・レポートの送信、図 2

- 一方、DSPMQRTE はアクティビティ・レポートが ACTIV.REPLY.Q に到着するのを待機しています。2 つだけが到着します。送信がまだ完了していないようなので、DSPMQRTE は 120 秒間待ち続けます。

表示される出力は以下のようになります。


```
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.
```

- DSPMQRTE が監視した最後の操作は Send であったため、チャンネルは稼働中です。ここで、キュー・マネージャー QM2 (RemoteQMgrName で識別) からもっと多くのアクティビティー・レポートが送られてこなかった理由を解明する必要があります。
- システム・キューにアクティビティー情報があるかどうかを確認するには、QM2 で DSPMQRTE を開始し、さらにアクティビティー・レポートを収集するよう試みます。以下のコマンドを使用して DSPMQRTE を開始します。

```
dspmqrte -m QM2 -q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C9154220001502 -v outline
```

414D51204C41524745512020202020A3C9154220001502 は、書き込まれた経路トレース・メッセージの MsgId です。

- DSPMQRTE は次に、一連の MQGET を再度実行し、システム・アクティビティー・キュー上で、指定された ID を持つ 経路トレース・メッセージに関連する応答を待ちます。

- DSPMQRTE はもう 1 つアクティビティー・レポートを取得し、それを表示します。DSPMQRTE は、先行するアクティビティー・レポートが欠落していると判断し、下記のようなメッセージを表示します。ただし、経路のこの部分については既に分かっています。

表示される出力は以下のようになります。

```

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.

```

- このアクティビティー・レポートは、経路情報がこれで完了したことを示しています。問題は発生しませんでした。
- 経路情報が使用できない、または DSPMQRTE がすべての経路を表示できないというだけで、メッセージが送信されなかったとは限りません。例えば、いくつかのキュー・マネージャーのキュー・マネージャー属性が異なっていたり、応答キューが応答を返すように定義されていなかった可能性があります。

例 4 - チャネル問題の診断

経路トレース・メッセージがターゲット・キューに届かない問題を診断します。

この例では、IBM MQ 経路表示アプリケーションがキュー・マネージャー QM1 に接続し、経路トレース・メッセージを生成して、リモート・キュー・マネージャー QM2 上のターゲット・キュー TARGET.Q への送信を試行します。この例では、経路トレース・メッセージはターゲット・キューへ届きません。有効なアクティビティー・レポートを使用して、問題を診断します。

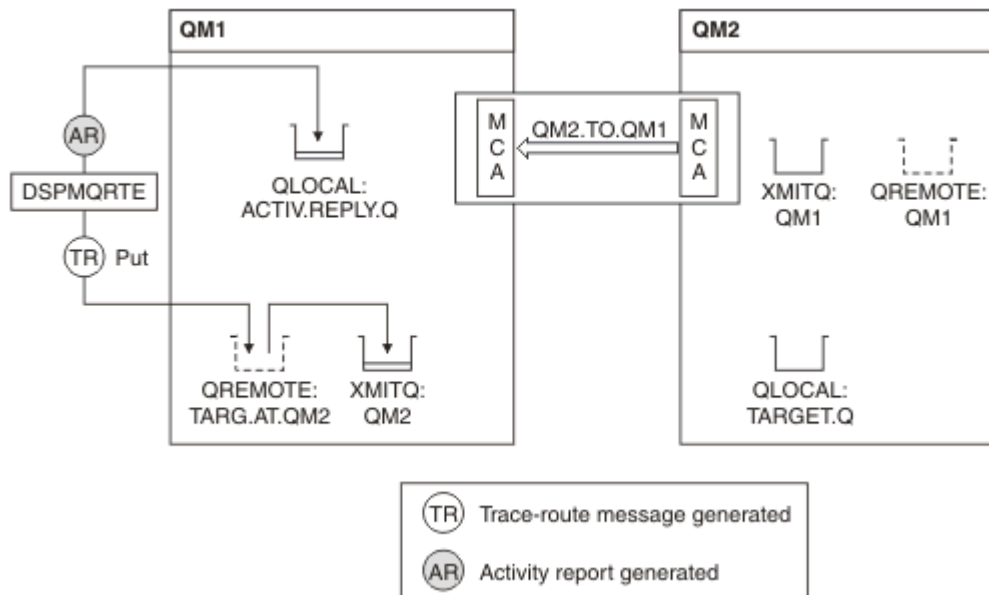


図 19. チャンネル問題の診断

- この例では、チャンネル QM1.TO.QM2 は稼働していません。
- DSPMQRTE は経路トレース・メッセージを (例 1 のように) ターゲット・キューに入れ、アクティビティ・レポートを生成します。
- 伝送キュー (QM2) からメッセージを取得する MCA はないため、これが DSPMQRTE が応答キューから取得する唯一のアクティビティ・レポートです。今回は、送信が完了していないという事実が問題を示しています。管理者は ResolvedQName にある伝送キューを使用して、伝送キューが作動していない理由を調査することができます。

表示される出力は以下のようになります。

```
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.
```

アクティビティ・レポート参照

このページを使用して、アクティビティ・レポート・メッセージの形式についての概要を説明します。アクティビティ・レポートのメッセージ・データには、そのアクティビティを記述するパラメーターが含まれています。

アクティビティ・レポート形式

アクティビティ・レポートは、メッセージ記述子とメッセージ・データを含む、標準的な IBM MQ レポート・メッセージです。アクティビティ・レポートは、メッセージがキュー・マネージャー・ネットワーク

ーク経由で送信されたときに、メッセージに対してアクティビティを実行したアプリケーションによって生成されます。

アクティビティ・レポートには以下の情報が含まれています。

メッセージ記述子

MQMD 構造

メッセージ・データ

以下により構成されます。

- 組み込み PCF ヘッダー (MQEPH)
- アクティビティ報告書のメッセージ・データ。

アクティビティ・レポート・メッセージ・データは、*Activity* PCF グループから構成されます。ただし、経路トレース・メッセージについては生成された場合は、*TraceRoute* PCF グループから構成されます。

101 ページの表 19 は、一定の条件下でのみ返されるパラメーターを含む、これらのレポートの構造を示しています。

表 19. アクティビティ・レポート形式		
MQMD 構造	組み込み PCF ヘッダー MQEPH 構造体	アクティビティ・レポート・メッセージ・データ
構造体 ID 構造バージョン レポート・オプション メッセージ・タイプ 満了時刻 Feedback Encoding コード化文字セット ID メッセージ形式 優先順位 Persistence メッセージ ID 関連 ID バックアウト・カウント 応答先キュー 応答先キュー・マネージャー ユーザー ID アカウンティング・トークン アプリケーション識別データ アプリケーション・タイプ アプリケーション名 書き込み日付 書き込み時刻 発生元アプリケーション・データ グループ ID メッセージ順序番号 オフセット メッセージ・フラグ 元の長さ	構造体 ID 構造バージョン 構造の長さ Encoding コード化文字セット ID メッセージ形式 フラグ PCF ヘッダー (MQCFH) 構造タイプ 構造の長さ 構造バージョン コマンド ID メッセージ順序番号 制御オプション 完了コード 理由コード パラメーター数	アクティビティ アクティビティ・アプリケーション名 アクティビティ・アプリケーション・タイプ アクティビティの記述 Operation 操作タイプ 操作日付 操作時刻 メッセージ メッセージ長 MQMD ⁸ EmbeddedMQMD キュー・マネージャー名 キュー共有グループ名 キュー名 ¹ ² ³ ⁷ 解決されたキュー名 ¹ ³ ⁷ リモート・キュー名 ³ ⁷ リモート・キュー・マネージャー名 ² ³ ⁴ ⁵ ⁷ サブスクリプション・レベル ⁹ サブスクリプション ID ⁹ フィードバック ² ¹⁰ チャンネル名 ⁴ ^{5つの} チャンネル・タイプ ⁴ ^{5つの} 伝送キュー名 ⁵ トレース経路 ⁶ 詳細 記録されたアクティビティ 記録されていないアクティビティ 不連続カウント 最大アクティビティ数 累積 Deliver

注:

1. Get 操作および Browse 操作に対して返されます。
2. Discard 操作に対して返されます。
3. Put 操作、Put Reply 操作、および Put Report 操作に対して返されます。

4. Receive 操作に対して返されます。
5. Send 操作に対して返されます。
6. 経路トレース・メッセージの場合に返されます。
7. トピックへの Put 操作に対しては返されず、パブリッシュ・アクティビティー内に含まれます。
8. Excluded Publish 操作に対しては返されません。Publish 操作および Discarded Publish 操作に対しては、パラメーターのサブセットとともに返されます。
9. Publish 操作、Discarded Publish 操作、および Excluded Publish 操作に対して返されます。
10. Discarded Publish 操作および Excluded Publish 操作に対して返されます。

アクティビティー・レポート MQMD (メッセージ記述子)

このページを使用して、アクティビティー・レポートの MQMD 構造に含まれた値を確認します。

StrucId

構造 ID:

データ・タイプ
MQCHAR4

値
MQMD_STRUC_ID

Version

構造体のバージョン番号

データ・タイプ
MQLONG

値
元のメッセージ記述子からコピーされます。指定可能な値は以下のとおりです。

MQMD_VERSION_1

Version-1 メッセージ記述子構造。すべての環境でサポートされます。

MQMD_VERSION_2

Version-2 メッセージ記述子構造。以下の環境でサポートされます。

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

およびこれらのシステムに接続されるすべての IBM MQ MQI clients。

Report

追加のレポート・メッセージのためのオプション

データ・タイプ
MQLONG

値
MQRO_PASS_DISCARD_AND_EXPIRY または MQRO_DISCARD_MSG が元のメッセージ記述子の Report フィールドに指定されていた場合:

MQRO_DISCARD

レポートは、宛先キューに送信できなかった場合は廃棄されます。

それ以外の場合:

MQRO_NONE

レポートは必要ありません。

MsgType

メッセージのタイプを示します

データ・タイプ

MQLONG

値

MQMT_REPORT

Expiry

レポート・メッセージの存続時間

データ・タイプ

MQLONG

値

元のメッセージ記述子の *Report* フィールドに **MQRO_PASS_DISCARD_AND_EXPIRY** が指定されている場合は、元のメッセージの残りの有効期間が使用されます。

それ以外の場合:

MQEI_UNLIMITED

レポートには有効期間がありません。

Feedback

説明: フィードバックまたは理由コード。

データ型: MQLONG

値: **MQFB_ACTIVITY**
アクティビティ・レポート。

Encoding

説明: レポート・メッセージ・データの数値エンコード。

データ型: MQLONG

値: MQENC_NATIVE

CodedCharSetId

説明: レポート・メッセージ・データの文字セット ID。

データ型: MQLONG

値: 必要に応じて設定します。

Format

説明: レポート・メッセージ・データの形式名。

データ型: MQCHAR8.

値: **MQFMT_EMBEDDED_PCF**
組み込み PCF メッセージ。

Priority

説明: レポート・メッセージ優先度。

データ型: MQLONG

値: 元のメッセージ記述子からコピーされます。

Persistence

説明: レポート・メッセージの持続性。
データ型: MQLONG
値: 元のメッセージ記述子からコピーされます。

MsgId

説明: メッセージ ID。
データ型: MQBYTE24.
値: 元のメッセージ記述子の *Report* フィールドに MQRO_PASS_MSG_ID が指定されている場合は、元のメッセージのメッセージ ID が使用されます。
そうでない場合は、キュー・マネージャーによって固有値が生成されます。

CorrelId

説明: 相関 ID。
データ型: MQBYTE24.
値: 元のメッセージ記述子の *Report* フィールドに MQRO_PASS_CORREL_ID が指定されている場合は、元のメッセージの相関 ID が使用されます。
そうでない場合は、元のメッセージからメッセージ ID がコピーされます。

BackoutCount

説明: バックアウトのカウンター。
データ型: MQLONG
値: 0。

ReplyToQ

説明: 応答キューの名前。
データ型: MQCHAR48.
値: ブランク。

ReplyToQMgr

説明: 応答キュー・マネージャーの名前。
データ型: MQCHAR48.
値: レポート・メッセージを生成したキュー・マネージャー名。

UserIdentifier

説明: レポート・メッセージを生成したアプリケーションのユーザー ID。
データ型: MQCHAR12.
値: 元のメッセージ記述子からコピーされます。

AccountingToken

説明: メッセージの結果として行われる作業についてアプリケーションによる課金を可能にするためのアカウントिंग・トークン。

データ型: MQBYTE32.
値: 元のメッセージ記述子からコピーされます。

ApplIdentityData

説明: IDに関連するアプリケーション・データ。
データ型: MQCHAR32.
値: 元のメッセージ記述子からコピーされます。

PutApplType

説明: レポート・メッセージを書き込んだアプリケーションのタイプ。
データ型: MQLONG
値: **MQAT_QMGR**
キュー・マネージャーによって生成されたメッセージ。

PutApplName

説明: レポート・メッセージを書き込んだアプリケーションの名前。
データ型: MQCHAR28.
値: キュー・マネージャー名の先頭 28 バイト、またはレポート・メッセージを生成した MCA の名前。

PutDate

説明: メッセージが書き込まれた日付。
データ型: MQCHAR8.
値: キュー・マネージャーによって生成された時点。

PutTime

説明: メッセージが書き込まれた時間。
データ型: MQCHAR8.
値: キュー・マネージャーによって生成された時点。

ApplOriginData

説明: 発生元に関するアプリケーション・データ。
データ型: MQCHAR4.
値: ブランク。

Version が MQMD_VERSION_2 である場合は、次の追加フィールドがあります。

GroupId

説明: 物理メッセージが属するメッセージ・グループ、または論理メッセージを識別します。
データ型: MQBYTE24.
値: 元のメッセージ記述子からコピーされます。

MsgSeqNumber

説明: グループ中の論理メッセージの順序番号。
データ型: MQLONG
値: 元のメッセージ記述子からコピーされます。

Offset

説明: 論理メッセージの先頭を起点とする、物理メッセージ中のデータのオフセット。
データ型: MQLONG
値: 元のメッセージ記述子からコピーされます。

MsgFlags

説明: メッセージの属性を指定したり、その処理を制御したりするメッセージ・フラグ。
データ型: MQLONG
値: 元のメッセージ記述子からコピーされます。

OriginalLength

説明: 元のメッセージの長さ。
データ型: MQLONG
値: 元のメッセージ記述子からコピーされます。

アクティビティ・レポート MQEPH (組み込み PCF ヘッダー)

このページを使用して、アクティビティ・レポートの MQEPH 構造に含まれた値を確認します。

MQEPH 構造には、アクティビティ・レポートのメッセージ・データに付随する PCF 情報と、これに続くアプリケーション・メッセージ・データの両方の説明が含まれています。

アクティビティ・レポートの場合、MQEPH 構造には以下の値が含まれます。

StrucId

説明: 構造体 ID
データ型: MQCHAR4.
値: MQEPH_STRUC_ID

Version

説明: 構造体のバージョン番号。
データ型: MQLONG
値: MQEPH_VERSION_1.

StrucLength

説明: 構造の長さ。
データ型: MQLONG
値: 後続の PCF パラメーター構造を含む、構造の合計長。

Encoding

説明: 最後の PCF パラメーター構造に続くメッセージ・データの数値エンコード。

データ型: MQLONG
値: 元のアプリケーション・メッセージ・データのデータがレポート・メッセージに含まれている場合、値は元のメッセージ記述子の *Encoding* フィールドからコピーされます。
そうでない場合は 0 です。

CodedCharSetId

説明: 最後の PCF パラメーター構造に続くメッセージ・データの文字セット ID。
データ型: MQLONG
値: 元のアプリケーション・メッセージ・データのデータがレポート・メッセージに含まれている場合、値は元のメッセージ記述子の *CodedCharSetId* フィールドからコピーされます。
そうでない場合は MQCCSI_UNDEFINED です。

Format

説明: 最後の PCF パラメーター構造に続くメッセージ・データの形式名。
データ型: MQCHAR8.
値: 元のアプリケーション・メッセージ・データのデータがレポート・メッセージに含まれている場合、値は元のメッセージ記述子の *Format* フィールドからコピーされます。
そうでない場合は MQFMT_NONE です。

Flags

説明: 構造の属性を指定する、またはその処理を制御するフラグ。
データ型: MQLONG
値: **MQEPH_CCSID_EMBEDDED**
文字データを含むパラメーターの文字セットが、それぞれの構造の *CodedCharSetId* フィールド内で個々に指定されることを指定します。

PCFHeader

説明: プログラマブル・コマンド・フォーマット・ヘッダー。
データ型: MQCFH
値: [107 ページの『アクティビティ・レポート MQCFH \(PCF ヘッダー\)』](#)を参照。

アクティビティ・レポート MQCFH (PCF ヘッダー)

このページを使用して、アクティビティ・レポートの MQCFH 構造に含まれた PCF 値を確認します。
アクティビティ・レポートの場合、MQCFH 構造には以下の値が含まれます。

Type

説明: レポート・メッセージの内容を識別する構造タイプ。
データ型: MQLONG
値: **MQCFT_REPORT**
メッセージはレポートです。

StrucLength

説明: 構造の長さ。
データ型: MQLONG
値: **MQCFH_STRUC_LENGTH**
MQCFH 構造のバイト単位の長さ。

Version

説明: 構造体のバージョン番号。
データ型: MQLONG
値: MQCFH_VERSION_3

Command

説明: コマンド ID。メッセージのカテゴリを識別します。
データ型: MQLONG
値: **MQCMD_ACTIVITY_MSG**
メッセージ・アクティビティ。

MsgSeqNumber

説明: メッセージ順序番号 関連メッセージのグループ内のメッセージの順序番号です。
データ型: MQLONG
値: 1 秒。

Control

説明: 制御オプション。
データ型: MQLONG
値: MQCFC_LAST

CompCode

説明: 完了コード。
データ型: MQLONG
値: MQCC_OK

Reason

説明: 完了コードを修飾する理由コード。
データ型: MQLONG
値: MQRC_NONE

ParameterCount

説明: パラメーター構造のカウント。これは、MQCFH 構造に続くパラメーター構造の数です。グループ構造 (MQCFGR) とそれに含まれるパラメーター構造は、1 つの構造としてカウントされます。
データ型: MQLONG
値: 1 以上。

アクティビティ・レポート・メッセージ・データ

このページを使用して、アクティビティ・レポート・メッセージ内の *Activity* PCF グループに含まれたパラメーターを確認します。一部のパラメーターは、特定の操作が実行された場合にのみ返されます。

アクティビティ・レポート・メッセージ・データは、*Activity* PCF グループから構成されます。ただし、経路トレース・メッセージについて生成された場合は、*TraceRoute* PCF グループから構成されます。

Activity PCF グループについては、このトピックで詳しく説明します。

一部のパラメーター (操作に固有のアクティビティ・レポート・メッセージ・データと記述している) は、特定の操作が実行された場合にのみ返されます。

アクティビティ・レポートの場合、アクティビティ・レポートのメッセージ・データには以下のパラメーターが含まれています。

Activity

説明:	アクティビティを記述するグループ化されたパラメーター。
ID:	MQGACF_ACTIVITY
データ型:	MQCFGR
組み込まれる PCF グループ:	なし。
PCF グループ内の パラメーター:	<i>ActivityApplName</i> <i>ActivityApplType</i> <i>ActivityDescription</i> <i>Operation</i> <i>TraceRoute</i>
戻り:	常時。

ActivityApplName

説明:	アクティビティを実行したアプリケーションの名前。
ID:	MQCACF_APPL_NAME
データ型:	MQCFST
組み込まれる PCF グループ:	<i>Activity</i>
最大長:	MQ_APPL_NAME_LENGTH
戻り:	常時。

ActivityApplType

説明:	アクティビティを実行したアプリケーションのタイプ。
ID:	MQIA_APPL_TYPE
データ型:	MQCFIN
組み込まれる PCF グループ:	<i>Activity</i>
戻り:	常時。

ActivityDescription

説明:	アプリケーションによって実行されたアクティビティの説明。
ID:	MQCACF_ACTIVITY_DESCRIPTION

データ型: MQCFST
組み込まれる PCF Activity
グループ:
最大長: 64
戻り: 常時。

Operation

説明: アクティビティの操作を記述するグループ化されたパラメーター。
ID: MQGACF_OPERATION
データ型: MQCFGR
組み込まれる PCF Activity
グループ:
PCF グループ内の
パラメーター: *OperationType*
OperationDate
OperationTime
Message
QMgrName
QSGName

注: 操作タイプに応じて、追加パラメーターがこのグループに返されます。これらの追加のパラメーターは、操作に固有のアクティビティ・レポート・メッセージ・データと記述しています。

戻り: アクティビティ内の 1 操作ごとに 1 つの *Operation* PCF グループ。

OperationType

説明: 実行された操作のタイプ。
ID: MQIACF_OPERATION_TYPE
データ型: MQCFIN
組み込まれる PCF *Operation*
グループ:
値: MQOPER_*
戻り: 常時。

OperationDate

説明: 操作が実行された日付。
ID: MQCACF_OPERATION_DATE
データ型: MQCFST
組み込まれる PCF *Operation*
グループ:
最大長: MQ_DATE_LENGTH
戻り: 常時。

OperationTime

説明: 操作が実行された時刻。

ID: MQCACF_OPERATION_TIME
データ型: MQCFST
組み込まれる PCF グループ: *Operation*
最大長: MQ_TIME_LENGTH
戻り: 常時。

Message

説明: アクティビティを引き起こしたメッセージを記述するグループ化されたパラメーター。

ID: MQGACF_MESSAGE
データ型: MQCFGR
組み込まれる PCF グループ: *Operation*
グループ内のパラメーター: *MsgLength*
MQMD
EmbeddedMQMD

戻り: 常時。ただし、パブリッシュ除外操作を除く。

MsgLength

説明: アクティビティが実行される前の、そのアクティビティを引き起こしたメッセージの長さ。

ID: MQIACF_MSG_LENGTH
データ型: MQCFIN
組み込まれる PCF グループ: *Message*
戻り: 常時。

MQMD

説明: アクティビティを引き起こしたメッセージのメッセージ記述子に関連するグループ化されたパラメーター。

ID: MQGACF_MQMD
データ型: MQCFGR
組み込まれる PCF グループ: *Message*

グループ内のパラ メーター:	<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>
-------------------	---

戻り: 常時。ただし、パブリッシュ除外操作を除く。

EmbeddedMQMD

説明:	伝送キュー上のメッセージ内に組み込まれたメッセージ記述子を記述する、グループ化されたパラメーター。
ID:	MQGACF_EMBEDDED_MQMD
データ型:	MQCFGR
組み込まれる PCF グループ:	<i>Message</i>

グループ内のパラ
メーター:

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

戻り: キューが伝送キューに解決される取得操作の場合。

StrucId

説明: 構造体 ID

ID: MQCACF_STRUC_ID

データ型: MQCFST

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

最大長: 4.

戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操
作およびパブリッシュ廃棄操作を除く。

Version

説明: 構造体のバージョン番号。

ID: MQIACF_VERSION

データ型: MQCFIN

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

Report

説明: レポート・メッセージのオプション。

ID: MQIACF_REPORT

データ型: MQCFIN

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

MsgType

説明: メッセージのタイプを示します。

ID: MQIACF_MSG_TYPE

データ型: MQCFIN

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

Expiry

説明: メッセージ存続期間。

ID: MQIACF_EXPIRY

データ型: MQCFIN

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

Feedback

説明: フィードバックまたは理由コード。

ID: MQIACF_FEEDBACK

データ型: MQCFIN

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

Encoding

説明: メッセージ・データの数値エンコード。

ID: MQIACF_ENCODING

データ型: MQCFIN

組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*

戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

CodedCharSetId

説明: メッセージ・データの文字セット ID。

ID: MQIA_CODED_CHAR_SET_ID

データ型: MQCFIN

組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*

戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

Format

説明: メッセージ・データの形式名。

ID: MQCACH_FORMAT_NAME

データ型: MQCFST

組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*

最大長: MQ_FORMAT_LENGTH

戻り: 常時。ただし、パブリッシュ除外操作を除く。

Priority

説明: メッセージの優先度。

ID: MQIACF_PRIORITY

データ型: MQCFIN

組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*

戻り: 常時。ただし、パブリッシュ除外操作を除く。

Persistence

説明: メッセージの持続性。

ID: MQIACF_PERSISTENCE

データ型: MQCFIN

組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*

戻り: 常時。ただし、パブリッシュ除外操作を除く。

MsgId

説明: メッセージ ID。

ID: MQBACF_MSG_ID

データ型: MQCFBS

組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*
最大長: MQ_MSG_ID_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。

CorrelId

説明: 相関 ID。
ID: MQBACF_CORREL_ID
データ型: MQCFBS
組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*
最大長: MQ_CORREL_ID_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。

BackoutCount

説明: バックアウトのカウンター。
ID: MQIACF_BACKOUT_COUNT
データ型: MQCFIN
組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*
戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

ReplyToQ

説明: 応答キューの名前。
ID: MQCACF_REPLY_TO_Q
データ型: MQCFST
組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*
最大長: MQ_Q_NAME_LENGTH。
戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

ReplyToQMgr

説明: 応答キュー・マネージャーの名前。
ID: MQCACF_REPLY_TO_Q_MGR
データ型: MQCFST
組み込まれる PCF グループ: *MQMD* または *EmbeddedMQMD*
最大長: MQ_Q_MGR_NAME_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

UserIdentifier

説明: メッセージを発信したアプリケーションのユーザー ID。
ID: MQCACF_USER_IDENTIFIER
データ型: MQCFST
組み込まれる PCF MQMD または *EmbeddedMQMD*
グループ:
最大長: MQ_USER_ID_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。

AccountingToken

説明: メッセージの結果として行われる作業についてアプリケーションによる課金を可能にするためのアカウントング・トークン。
ID: MQBACF_ACCOUNTING_TOKEN
データ型: MQCFBS
組み込まれる PCF MQMD または *EmbeddedMQMD*
グループ:
最大長: MQ_ACCOUNTING_TOKEN_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。

ApplIdentityData

説明: ID に関連するアプリケーション・データ。
ID: MQCACF_APPL_IDENTITY_DATA
データ型: MQCFST
組み込まれる PCF MQMD または *EmbeddedMQMD*
グループ:
最大長: MQ_APPL_IDENTITY_DATA_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。

PutApplType

説明: メッセージを書き込むアプリケーションのタイプ。
ID: MQIA_APPL_TYPE
データ型: MQCFIN
組み込まれる PCF MQMD または *EmbeddedMQMD*
グループ:
戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

PutApplName

説明: メッセージを書き込むアプリケーションの名前。
ID: MQCACF_APPL_NAME
データ型: MQCFST
組み込まれる PCF MQMD または *EmbeddedMQMD*
グループ:

最大長: MQ_APPL_NAME_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

PutDate

説明: メッセージが書き込まれた日付。
ID: MQCACF_PUT_DATE
データ型: MQCFST
組み込まれる PCF グループ: MQMD または *EmbeddedMQMD*
最大長: MQ_PUT_DATE_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

PutTime

説明: メッセージが書き込まれた時間。
ID: MQCACF_PUT_TIME
データ型: MQCFST
組み込まれる PCF グループ: MQMD または *EmbeddedMQMD*
最大長: MQ_PUT_TIME_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

ApplOriginData

説明: 発生元に関するアプリケーション・データ。
ID: MQCACF_APPL_ORIGIN_DATA
データ型: MQCFST
組み込まれる PCF グループ: MQMD または *EmbeddedMQMD*
最大長: MQ_APPL_ORIGIN_DATA_LENGTH
戻り: 常時。ただし、パブリッシュ除外操作を除く。MQMD の場合は、パブリッシュ操作およびパブリッシュ廃棄操作を除く。

GroupId

説明: 物理メッセージが属するメッセージ・グループ、または論理メッセージを識別します。
ID: MQBACF_GROUP_ID
データ型: MQCFBS
組み込まれる PCF グループ: MQMD または *EmbeddedMQMD*
最大長: MQ_GROUP_ID_LENGTH

戻り: *Version* に MQMD_VERSION_2 が指定されている場合。パブリッシュ除外操作では返されない。MQMD の場合、パブリッシュ操作およびパブリッシュ廃棄操作では返されない。

MsgSeqNumber

説明: グループ中の論理メッセージの順序番号。

ID: MQIACH_MSG_SEQUENCE_NUMBER

データ型: MQCFIN

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

戻り: *Version* に MQMD_VERSION_2 が指定されている場合。パブリッシュ除外操作では返されない。MQMD の場合、パブリッシュ操作およびパブリッシュ廃棄操作では返されない。

Offset

説明: 論理メッセージの先頭を起点とする、物理メッセージ中のデータのオフセット。

ID: MQIACF_OFFSET

データ型: MQCFIN

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

戻り: *Version* に MQMD_VERSION_2 が指定されている場合。パブリッシュ除外操作では返されない。MQMD の場合、パブリッシュ操作およびパブリッシュ廃棄操作では返されない。

MsgFlags

説明: メッセージの属性を指定したり、その処理を制御したりするメッセージ・フラグ。

ID: MQIACF_MSG_FLAGS

データ型: MQCFIN

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

戻り: *Version* に MQMD_VERSION_2 が指定されている場合。パブリッシュ除外操作では返されない。MQMD の場合、パブリッシュ操作およびパブリッシュ廃棄操作では返されない。

OriginalLength

説明: 元のメッセージの長さ。

ID: MQIACF_ORIGINAL_LENGTH

データ型: MQCFIN

組み込まれる PCF
グループ: MQMD または *EmbeddedMQMD*

戻り: *Version* に MQMD_VERSION_2 が指定されている場合。パブリッシュ除外操作では返されない。MQMD の場合、パブリッシュ操作およびパブリッシュ廃棄操作では返されない。

QMgrName

説明: アクティビティーが実行されたキュー・マネージャーの名前。
ID: MQCA_Q_MGR_NAME
データ型: MQCFST
組み込まれる PCF グループ: *Operation*
最大長: MQ_Q_MGR_NAME_LENGTH
戻り: 常時。

QSGName

説明: アクティビティーが実行されたキュー・マネージャーが属するキュー共有グループの名前。
ID: MQCA_QSG_NAME
データ型: MQCFST
組み込まれる PCF グループ: *Operation*
最大長: MQ_QSG_NAME_LENGTH
戻り: アクティビティーが IBM MQ for z/OS キュー・マネージャーで実行された場合。

TraceRoute

説明: 経路トレース・メッセージの属性を指定するグループ化されたパラメーター。
ID: MQGACF_TRACE_ROUTE
データ型: MQCFGR
含まれる PCF グループ: *Activity*
グループ内のパラメーター: *Detail*
RecordedActivities
UnrecordedActivities
DiscontinuityCount
MaxActivities
Accumulate
Forward
Deliver
戻り: 経路トレース・メッセージに対してアクティビティーが実行された場合。

TraceRoute PCF グループのパラメーターの値は、アクティビティー・レポートが生成された時点の経路トレース・メッセージからの値です。

操作に固有のアクティビティー・レポート・メッセージ・データ

このページを使用して、*OperationType* パラメーターの値に応じて、アクティビティー・レポート内の PCF グループ *Operation* に返される場合がある、追加の PCF パラメーターを確認します。

追加パラメーターは、以下の操作タイプによって異なります。

Get/Browse (MQOPER_GET/MQOPER_BROWSE)

Get/Browse (MQOPER_GET/MQOPER_BROWSE) 操作タイプ (キューのメッセージが取得または参照された) の場合に PCF グループ *Operation* に返される 追加のアクティビティ・レポート・メッセージ・データ・パラメーター。

QName

説明: オープンされたキューの名前。
ID: MQCA_Q_NAME
データ型: MQCFST
組み込まれる PCF グループ: *Operation*
最大長: MQ_Q_NAME_LENGTH
戻り: 常時。

ResolvedQName

説明: オープンされたキューが解決された名前。
ID: MQCACF_RESOLVED_Q_NAME
データ型: MQCFST
組み込まれる PCF グループ: *Operation*
最大長: MQ_Q_NAME_LENGTH
戻り: 常時。

Discard (MQOPER_DISCARD)

Discard (MQOPER_DISCARD) 操作タイプ (メッセージが廃棄された) の場合に PCF グループ *Operation* に返される 追加のアクティビティ・レポート・メッセージ・データ・パラメーター。

Feedback

説明: メッセージが廃棄された理由。
ID: MQIACF_FEEDBACK
データ型: MQCFIN
組み込まれる PCF グループ: *Operation*
戻り: 常時。

QName

説明: オープンされたキューの名前。
ID: MQCA_Q_NAME
データ型: MQCFST
最大長: MQ_Q_NAME_LENGTH
組み込まれる PCF グループ: *Operation*
戻り: メッセージが正常にキューに書き込まれなかったために、そのメッセージが廃棄された場合。

RemoteQMgrName

説明:	メッセージの宛先であったキュー・マネージャーの名前。
ID:	MQCA_REMOTE_Q_MGR_NAME
データ型:	MQCFST
最大長:	MQ_Q_MGR_NAME_LENGTH
組み込まれる PCF グループ:	<i>Operation</i>
戻り:	<i>Feedback</i> の値が MQFB_NOT_FORWARDED の場合。

Publish/Discarded Publish/Excluded Publish (MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/MQOPER_EXCLUDED_PUBLISH)

Publish/Discarded Publish/Excluded Publish (MQOPER_PUBLISH/MQOPER_DISCARDED_PUBLISH/MQOPER_EXCLUDED_PUBLISH) 操作タイプ (パブリッシュ/サブスクライブ・メッセージが送信、廃棄、または除外された) の場合に PCF グループ *Operation* に返される追加のアクティビティ・レポート・メッセージ・データ・パラメーター。

SubId

説明:	サブスクリプション ID。
ID:	MQBACF_SUB_ID
データ型:	MQCFBS
組み込まれる PCF グループ:	<i>Operation</i>
戻り:	常時。

SubLevel

説明:	サブスクリプション・レベル。
ID:	MQIACF_SUB_LEVEL
データ型:	MQCFIN
組み込まれる PCF グループ:	<i>Operation</i>
戻り:	常時。

Feedback

説明:	メッセージ廃棄の理由。
ID:	MQIACF_FEEDBACK
データ型:	MQCFIN
組み込まれる PCF グループ:	<i>Operation</i>
戻り:	メッセージがサブスクライバーに送信されなかったために、そのメッセージが廃棄された場合。または、サブスクライバーが除外されたために、メッセージが送信されなかった場合。

パブリッシュ操作 MQOPER_PUBLISH では、特定のサブスクライバーに送信されるメッセージに関する情報を得ることができます。この操作では、関連する書き込み操作で記述されたメッセージから変更された可能性のある、より新しいメッセージの要素が記述されます。書き込み操作の場合と同様に、この操作にはメッセージ・グループ MQGACF_MESSAGE が含まれ、その中に MQMD グループ MQGACF_MQMD が含

まれます。ただし、この MQMD グループに含まれるフィールドは、「Format」、「Priority」、「Persistence」、「MsgId」、「CorrelId」、「UserIdentifier」、「AccountingToken」、「ApplIdentityData」のみです。サブスクライバーは、これらのフィールドを指定変更できます。

サブスクライバーの *SubId* および *SubLevel* は、操作情報に組み込まれています。MQCMD_INQUIRE_SUBSCRIBER PCF コマンドで *SubID* を使用すれば、サブスクライバーに関する他のすべての属性を取得できます。

パブリッシュ廃棄操作 MQOPER_DISCARDED_PUBLISH は、point-to-point メッセージングでメッセージが送信されないときに使用される廃棄操作に類似しています。サブスクライバーがローカル宛先を指定していても、メッセージがローカル宛先に送信されないように明示的に要求されると、そのメッセージはそのサブスクライバーに送信されません。キューが満杯であるなどの原因により、宛先キューへのメッセージを取得するときに問題が発生した場合も、メッセージは送信されないと考えられます。

パブリッシュ廃棄操作の情報は、パブリッシュ操作の場合と同じ情報に、メッセージが送信されなかった理由が指定される「Feedback」フィールドが追加されます。この「Feedback」フィールドには、MQOPER_DISCARD 操作と共通する MQFB_* 値または MQRC_* 値が入ります。パブリッシュを (除外するのではなく) 廃棄する理由は、書き込みを廃棄する理由と同じです。

パブリッシュ除外操作 MQOPER_EXCLUDED_PUBLISH では、メッセージが送信されなかったサブスクライバーに関する情報が提供されます。このサブスクライバーは、サブスクライブしているトピックに関連する書き込み操作のトピックと一致するためメッセージ送信の対象とみなされましたが、他の選択基準が、トピックに書き込まれるメッセージに適合しないために、メッセージが送信されませんでした。パブリッシュ廃棄操作の場合と同様に、「Feedback」フィールドには、サブスクリプションが除外された理由に関する情報が指定されます。ただし、パブリッシュ廃棄操作と異なり、このサブスクライバーに対してはメッセージが生成されなかったためにメッセージ関連の情報は提供されません。

Put/Put Reply/Put Report (MQOPER_PUT/MQOPER_PUT_REPLY/MQOPER_PUT_REPORT)
Put/Put Reply/Put Report (MQOPER_PUT/MQOPER_PUT_REPLY/MQOPER_PUT_REPORT) 操作 タイプ (メッセージ、応答メッセージ、またはレポート・メッセージがキューに入れられた) の場合に PCF グループ *Operation* に返される追加のアクティビティ・レポート・メッセージ・データ・パラメーター。

QName

説明:	オープンされたキューの名前。
ID:	MQCA_Q_NAME
データ型:	MQCFST
組み込まれる PCF グループ:	<i>Operation</i>
最大長:	MQ_Q_NAME_LENGTH
戻り:	常時。ただし、書き込み操作がトピックに対しての操作である場合は返されず、パブリッシュ・アクティビティ内に含まれる。

ResolvedQName

説明:	オープンされたキューが解決された名前。
ID:	MQCACF_RESOLVED_Q_NAME
データ型:	MQCFST
組み込まれる PCF グループ:	<i>Operation</i>
最大長:	MQ_Q_NAME_LENGTH
戻り:	オープンされたキューが解決できた場合。書き込み操作がトピックに対しての操作である場合は返されず、パブリッシュ・アクティビティ内に含まれる。

RemoteQName

説明:	リモート・キュー・マネージャー上で分かっている、オープンされたキューの名前。
ID:	MQCA_REMOTE_Q_NAME
データ型:	MQCFST
組み込まれる PCF グループ:	Operation
最大長:	MQ_Q_NAME_LENGTH
戻り:	オープンされたキューがリモート・キューの場合。書き込み操作がトピックに対しての操作である場合は返されず、パブリッシュ・アクティビティー内に含まれる。

RemoteQMgrName

説明:	リモート・キューが定義されたリモート・キュー・マネージャーの名前。
ID:	MQCA_REMOTE_Q_MGR_NAME
データ型:	MQCFST
組み込まれる PCF グループ:	Operation
最大長:	MQ_Q_MGR_NAME_LENGTH
戻り:	オープンされたキューがリモート・キューの場合。書き込み操作がトピックに対しての操作である場合は返されず、パブリッシュ・アクティビティー内に含まれる。

TopicString

説明:	メッセージが書き込まれる完全トピック・ストリング。
ID:	MQCA_TOPIC_STRING
データ型:	MQCFST
組み込まれる PCF グループ:	Operation
戻り:	書き込み操作がトピックに対しての操作である場合は、パブリッシュ・アクティビティー内に含まれる。

Feedback

説明:	メッセージが送達不能キューに入れられている理由。
ID:	MQIACF_FEEDBACK
データ型:	MQCFIN
組み込まれる PCF グループ:	Operation
戻り:	メッセージが送達不能キューに入れられた場合。

Receive (MQOPER_RECEIVE)

Receive (MQOPER_RECEIVE) 操作タイプ (メッセージがチャンネルで受信された) の場合に PCF グループ *Operation* に返される 追加のアクティビティー・レポート・メッセージ・データ・パラメーター。

ChannelName

説明: メッセージが受信されたチャンネルの名前。
ID: MQCACH_CHANNEL_NAME
データ型: MQCFST
組み込まれる PCF *Operation*
グループ:
最大長: MQ_CHANNEL_NAME_LENGTH
戻り: 常時。

ChannelType

説明: メッセージが受信されたチャンネルのタイプ。
ID: MQIACH_CHANNEL_TYPE
データ型: MQCFIN
組み込まれる PCF *Operation*
グループ:
戻り: 常時。

RemoteQMgrName

説明: 受信されたメッセージの送信元キュー・マネージャーの名前。
ID: MQCA_REMOTE_Q_MGR_NAME
データ型: MQCFST
組み込まれる PCF *Operation*
グループ:
最大長: MQ_Q_MGR_NAME_LENGTH
戻り: 常時。

Send (MQOPER_SEND)

Send (MQOPER_SEND) 操作タイプ (メッセージがチャンネルで送信された) の場合に PCF グループ *Operation* に返される 追加のアクティビティ・レポート・メッセージ・データ・パラメーター。

ChannelName

説明: メッセージが送信されたチャンネルの名前。
ID: MQCACH_CHANNEL_NAME
データ型: MQCFST
組み込まれる PCF *Operation*
グループ:
最大長: MQ_CHANNEL_NAME_LENGTH
戻り: 常時。

ChannelType

説明: メッセージが送信されたチャンネルのタイプ。
ID: MQIACH_CHANNEL_TYPE
データ型: MQCFIN

組み込まれる PCF *Operation*
グループ:

戻り: 常時。

XmitQName

説明: メッセージが取り出された伝送キュー。

ID: MQCACH_XMIT_Q_NAME

データ型: MQCFST

組み込まれる PCF *Operation*
グループ:

最大長: MQ_Q_NAME_LENGTH。

戻り: 常時。

RemoteQMgrName

説明: メッセージが送信された送信先リモート・キュー・マネージャーの名前。

ID: MQCA_REMOTE_Q_MGR_NAME

データ型: MQCFST

組み込まれる PCF *Operation*
グループ:

最大長: MQ_Q_MGR_NAME_LENGTH

戻り: 常時。

経路トレース・メッセージ参照

このページを使用して、経路トレース・メッセージの形式についての概要を説明します。経路トレース・メッセージ・データには、経路トレース・メッセージによって引き起こされたアクティビティを記述したパラメーターが含まれています。

経路トレース・メッセージ形式

経路トレース・メッセージは、メッセージ記述子とメッセージ・データを含む標準 IBM MQ メッセージです。メッセージ・データには、経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信されたときに、このメッセージに対して実行されたアクティビティについての情報が含まれています。

経路トレース・メッセージには以下の情報が含まれています。

メッセージ記述子

Format フィールドが MQFMT_ADMIN または MQFMT_EMBEDDED_PCF に設定された MQMD 構造

メッセージ・データ

以下のいずれかで構成されます。

- PCF ヘッダー (MQCFH) および経路トレース・メッセージ・データ (*Format* が MQFMT_ADMIN に設定されている場合)
- 組み込み PCF ヘッダー (MQEPH)、経路トレース・メッセージ・データ、および追加のユーザー指定メッセージ・データ (*Format* が MQFMT_EMBEDDED_PCF に設定されている場合)

IBM MQ 経路表示アプリケーションを使用して経路トレース・メッセージを生成する場合、*Format* は MQFMT_ADMIN に設定されます。

経路トレース・メッセージ・データの内容は、以下のように、*TraceRoute* PCF グループの *Accumulate* パラメーターによって判別されます。

- *Accumulate* が MQROUTE_ACCUMULATE_NONE に設定されている場合、経路トレース・メッセージ・データには *TraceRoute* PCF グループが含まれます。
- *Accumulate* が MQROUTE_ACCUMULATE_IN_MSG または MQROUTE_ACCUMULATE_AND_REPLY のいずれかに設定されている場合、経路トレース・メッセージ・データには *TraceRoute* PCF グループとゼロ以上の *Activity* PCF グループが含まれます。

127 ページの表 20 に、経路トレース・メッセージの構造が示されています。

MQMD 構造	組み込み PCF ヘッダー MQEPH 構造体	経路トレース・メッセージ・データ
構造体 ID 構造バージョン レポート・オプション メッセージ・タイプ 満了時刻 Feedback Encoding コード化文字セット ID メッセージ形式 優先順位 Persistence メッセージ ID 関連 ID バックアウト・カウント 応答先キュー 応答先キュー・マネージャー ユーザー ID アカウンティング・トークン アプリケーション識別データ アプリケーション・タイプ アプリケーション名 書き込み日付 書き込み時刻 発生元アプリケーション・データ グループ ID メッセージ順序番号 オフセット メッセージ・フラグ 元の長さ	構造体 ID 構造バージョン 構造の長さ Encoding コード化文字セット ID メッセージ形式 フラグ PCF ヘッダー (MQCFH) 構造タイプ 構造の長さ 構造バージョン コマンド ID メッセージ順序番号 制御オプション 完了コード 理由コード パラメーター数	TraceRoute 詳細 記録されたアクティビティ 記録されていないアクティビティ 不連続カウント 最大アクティビティ数 累積 Deliver

経路トレース・メッセージ MQMD (メッセージ記述子)

このページを使用して、経路トレース・メッセージの MQMD 構造に含まれた値を確認します。

StrucId

説明: 構造体 ID
 データ型: MQCHAR4.
 値: MQMD_STRUC_ID

Version

説明: 構造体のバージョン番号。
 データ型: MQLONG

値: **MQMD_VERSION_1.**

Report

説明: レポート・メッセージのオプション。

データ型: MQLONG

値: 要件に従って設定します。一般的なレポート・オプションには以下があります。

MQRO_DISCARD_MSG

メッセージは、ローカル・キューに到着すると廃棄されます。

MQRO_PASS_DISCARD_AND_EXPIRY

各応答 (アクティビティ・レポートまたは経路トレース応答メッセージ) には、レポート・オプション **MQRO_DISCARD_MSG** が設定され、残りの有効期間が渡されています。このため、応答がキュー・マネージャー・ネットワークに無期限に残ることはありません。

MsgType

説明: メッセージのタイプ。

データ型: MQLONG

値: TraceRoute グループの *Accumulate* パラメーターが **MQROUTE_ACCUMULATE_AND_REPLY** に設定されている場合、メッセージ・タイプは **MQMT_REQUEST** です。

それ以外の場合:

MQMT_DATAGRAM

Expiry

説明: メッセージ存続期間。

データ型: MQLONG

値: 要件に従って設定します。このパラメーターは、経路トレース・メッセージがキュー・マネージャー・ネットワークに無期限に残るのを防ぐために使用できません。

Feedback

説明: フィードバックまたは理由コード。

データ型: MQLONG

値: **MQFB_NONE**

Encoding

説明: メッセージ・データの数値エンコード。

データ型: MQLONG

値: 必要に応じて設定します。

CodedCharSetId

説明: メッセージ・データの文字セット ID。

データ型: MQLONG

値: 必要に応じて設定します。

Format

説明:	メッセージ・データの形式名。
データ型:	MQCHAR8.
値:	MQFMT_ADMIN 管理メッセージ。 <i>TraceRoute</i> PCF グループの後にはユーザー・データが続きません。
	MQFMT_EMBEDDED_PCF 組み込み PCF メッセージ。 <i>TraceRoute</i> PCF グループの後にユーザー・データが続きます。

Priority

説明:	メッセージの優先度。
データ型:	MQLONG
値:	要件に従って設定します。

Persistence

説明:	メッセージの持続性。
データ型:	MQLONG
値:	要件に従って設定します。

MsgId

説明:	メッセージ ID。
データ型:	MQBYTE24.
値:	要件に従って設定します。

CorrelId

説明:	相関 ID。
データ型:	MQBYTE24.
値:	要件に従って設定します。

BackoutCount

説明:	バックアウトのカウンター。
データ型:	MQLONG
値:	0。

ReplyToQ

説明:	応答キューの名前。
データ型:	MQCHAR48.
値:	要件に従って設定します。

MsgType が MQMT_REQUEST に設定されている場合、または *Report* に何らかのレポート生成オプションが設定されている場合、このパラメーターは空白であってははいけません。

ReplyToQMgr

説明: 応答キュー・マネージャーの名前。
データ型: MQCHAR48.
値: 要件に従って設定します。

UserIdentifier

説明: メッセージを発信したアプリケーションのユーザー ID。
データ型: MQCHAR12.
値: 通常は設定します。

AccountingToken

説明: メッセージの結果として行われる作業についてアプリケーションによる課金を可能にするためのアカウントिंग・トークン。
データ型: MQBYTE32.
値: 通常は設定します。

ApplIdentityData

説明: IDに関連するアプリケーション・データ。
データ型: MQCHAR32.
値: 通常は設定します。

PutApplType

説明: メッセージを書き込むアプリケーションのタイプ。
データ型: MQLONG
値: 通常は設定します。

PutApplName

説明: メッセージを書き込むアプリケーションの名前。
データ型: MQCHAR28.
値: 通常は設定します。

PutDate

説明: メッセージが書き込まれた日付。
データ型: MQCHAR8.
値: 通常は設定します。

PutTime

説明: メッセージが書き込まれた時間。
データ型: MQCHAR8.
値: 通常は設定します。

ApplOriginData

説明: 発生元に関するアプリケーション・データ。

データ型: MQCHAR4.
値: 通常は設定します。

経路トレース・メッセージ MQEPH (組み込み PCF ヘッダー)

このページを使用して、経路トレース・メッセージの MQEPH 構造に含まれた値を確認します。

MQEPH 構造には、経路トレース・メッセージのメッセージ・データに付随する PCF 情報と、これに続くアプリケーション・メッセージ・データの両方の説明が含まれています。MQEPH 構造は、追加のユーザー・メッセージ・データが TraceRoute PCF グループに続く場合にのみ使用されます。

経路トレース・メッセージの場合、MQEPH 構造には以下の値が含まれます。

StrucId

説明: 構造体 ID
データ型: MQCHAR4.
値: MQEPH_STRUC_ID

Version

説明: 構造体のバージョン番号。
データ型: MQLONG
値: MQEPH_VERSION_1.

StrucLength

説明: 構造の長さ。
データ型: MQLONG
値: 後続の PCF パラメーター構造を含む、構造の合計長。

Encoding

説明: 最後の PCF パラメーター構造に続くメッセージ・データの数値エンコード。
データ型: MQLONG
値: メッセージ・データのエンコード方式。

CodedCharSetId

説明: 最後の PCF パラメーター構造に続くメッセージ・データの文字セット ID。
データ型: MQLONG
値: メッセージ・データの文字セット。

Format

説明: 最後の PCF パラメーター構造に続くメッセージ・データの形式名。
データ型: MQCHAR8.
値: メッセージ・データの形式名。

Flags

説明: 構造の属性を指定する、またはその処理を制御するフラグ。
データ型: MQLONG

値: **MQEPH_NONE**
フラグは指定されていません。

MQEPH_CCSID_EMBEDDED
文字データを含むパラメーターの文字セットが、それぞれの構造の *CodedCharSetId* フィールド内で個々に指定されることを指定します。

PCFHeader

説明: プログラマブル・コマンド・フォーマット・ヘッダー。

データ型: MQCFH

値: [132 ページの『経路トレース・メッセージ MQCFH \(PCF ヘッダー\)』を参照。](#)

経路トレース・メッセージ MQCFH (PCF ヘッダー)

このページを使用して、経路トレース・メッセージの MQCFH 構造に含まれた PCF 値を確認します。
経路トレース・メッセージの場合、MQCFH 構造には以下の値が含まれます。

Type

説明: メッセージの内容を識別する構造タイプ。

データ型: MQLONG

値: **MQCFT_TRACE_ROUTE**
メッセージは経路トレース・メッセージです。

StrucLength

説明: 構造の長さ。

データ型: MQLONG

値: **MQCFH_STRUC_LENGTH**
MQCFH 構造のバイト単位の長さ。

Version

説明: 構造体のバージョン番号。

データ型: MQLONG

値: MQCFH_VERSION_3

Command

説明: コマンド ID。メッセージのカテゴリを識別します。

データ型: MQLONG

値: **MQCMD_TRACE_ROUTE**
経路トレース・メッセージ。

MsgSeqNumber

説明: メッセージ順序番号 関連メッセージのグループ内のメッセージの順序番号です。

データ型: MQLONG

値: 1 秒.

Control

説明: 制御オプション。
データ型: MQLONG
値: MQCFC_LAST

CompCode

説明: 完了コード。
データ型: MQLONG
値: MQCC_OK

Reason

説明: 完了コードを修飾する理由コード。
データ型: MQLONG
値: MQRC_NONE

ParameterCount

説明: パラメーター構造のカウンタ。これは、MQCFH 構造に続くパラメーター構造の数です。グループ構造 (MQCFGR) とそれに含まれるパラメーター構造は、1つの構造としてカウンタされます。
データ型: MQLONG
値: 1 以上。

経路トレース・メッセージ・データ

このページを使用して、経路トレース・メッセージ・データの *TraceRoute* PCF グループの部分構成するパラメーターを確認します。

経路トレース・メッセージ・データの内容は、*TraceRoute* PCF グループの *Accumulate* パラメーターによって異なります。経路トレース・メッセージ・データは *TraceRoute* PCF グループと、0 個以上の *Activity* PCF グループで構成されます。*TraceRoute* PCF グループについては、このトピックで詳しく説明します。*Activity* PCF グループについては詳しくは、関連情報を参照してください。

経路トレース・メッセージ・データには以下のパラメーターが含まれています。

TraceRoute

説明: 経路トレース・メッセージの属性を指定するグループ化されたパラメーター。経路トレース・メッセージの場合、これらのパラメーターの一部を変更して、メッセージの処理方法を制御することができます。
ID: MQGACF_TRACE_ROUTE
データ型: MQCFGR
含まれる PCF グループ: なし。

グループ内のパラ
メーター: *Detail*
 RecordedActivities
 UnrecordedActivities
 DiscontinuityCount
 MaxActivities
 Accumulate
 Forward
 Deliver

Detail

説明: アクティビティーについて記録される詳細レベル。
ID: MQIACF_ROUTE_DETAIL
データ型: MQCFIN
含まれる PCF グループ: *TraceRoute.*
値: **MQROUTE_DETAIL_LOW**
 ユーザー作成アプリケーションで実行されたアクティビティーのみが記録されます。
 MQROUTE_DETAIL_MEDIUM
 MQROUTE_DETAIL_LOW で指定されたアクティビティーが記録されます。
 さらに、MCA によって実行されたアクティビティーも記録されます。
 MQROUTE_DETAIL_HIGH
 MQROUTE_DETAIL_LOW および MQROUTE_DETAIL_MEDIUM で指定された
 アクティビティーが記録されます。この詳細レベルでは、MCA はこれ以上の
 アクティビティー情報を記録しません。このオプションは、ユーザー作成ア
 プリケーションが追加のアクティビティー情報を記録する場合にのみ有効で
 す。

RecordedActivities

説明: 経路トレース・メッセージが引き起こした、情報が記録されたアクティビティー
 の数。
ID: MQIACF_RECORDED_ACTIVITIES
データ型: MQCFIN
含まれる PCF グループ: *TraceRoute.*

UnrecordedActivities

説明: 経路トレース・メッセージが引き起こした、情報が記録されなかったアクティビ
 ティーの数。
ID: MQIACF_UNRECORDED_ACTIVITIES
データ型: MQCFIN
含まれる PCF グループ: *TraceRoute.*

DiscontinuityCount

説明: 経路トレース・メッセージングをサポートしていないキュー・マネージャーから
 経路トレース・メッセージを受信した回数。

ID: MQIACF_DISCONTINUITY_COUNT
データ型: MQCFIN
含まれる PCF グループ: *TraceRoute.*

MaxActivities

説明: 処理が停止するまでに、経路トレース・メッセージが関与できるアクティビティの最大数。

ID: MQIACF_MAX_ACTIVITIES
データ型: MQCFIN
含まれる PCF グループ: *TraceRoute.*

値: **正整数**
アクティビティの最大数。

MQROUTE_UNLIMITED_ACTIVITIES
アクティビティ数に制限はありません。

Accumulate

説明: アクティビティ情報を経路トレース・メッセージ内に累積するかどうか、また経路トレース・メッセージが廃棄される前または伝送キュー以外のキューに入れられる前に、累積されたアクティビティ情報を含む応答メッセージを生成するかどうかを指定します。

ID: MQIACF_ROUTE_ACCUMULATION
データ型: MQCFIN
含まれる PCF グループ: *TraceRoute.*

値: **MQROUTE_ACCUMULATE_NONE**
経路トレース・メッセージのメッセージ・データにアクティビティ情報を累積しません。

MQROUTE_ACCUMULATE_IN_MSG
経路トレース・メッセージのメッセージ・データにアクティビティ情報を累積します。

MQROUTE_ACCUMULATE_AND_REPLY
アクティビティ情報を経路トレース・メッセージのメッセージ・データに累積し、経路トレース応答メッセージを生成します。

Forward

説明: 経路トレース・メッセージを転送可能な宛先キュー・マネージャーを指定します。メッセージをリモート・キュー・マネージャーに転送するかどうかを決定する際に、キュー・マネージャーでは [Forward](#) に記載されたアルゴリズムを使用します。

ID: MQIACF_ROUTE_FORWARDING
データ型: MQCFIN
含まれる PCF グループ: *TraceRoute.*

値: **MQROUTE_FORWARD_IF_SUPPORTED**
経路トレース・メッセージは、*TraceRoute* グループから *Deliver* パラメーターの値を受け取るキュー・マネージャーにのみ転送されます。

MQROUTE_FORWARD_ALL
経路トレース・メッセージは、*Deliver* パラメーターの値が渡されるかどうかにかかわらず、すべてのキュー・マネージャーに転送されます。

Deliver

説明: 経路トレース・メッセージが正常に宛先キューに到着した場合に実行するアクションを指定します。

ID: MQIACF_ROUTE_DELIVERY

データ型: MQCFIN

含まれる PCF グループ: *TraceRoute*.

値: **MQROUTE_DELIVER_YES**
到着すると、経路トレース・メッセージはターゲット・キューに入れられます。ターゲット・キューに対して破壊読み取りを実行するすべてのアプリケーションが、経路トレース・メッセージを受信できます。

MQROUTE_DELIVER_NO
経路トレース・メッセージは到着した時点で廃棄されます。

経路トレース応答メッセージ参照

このページを使用して、経路トレース応答メッセージの形式についての概要を説明します。経路トレース応答メッセージ・データは、そのデータが生成された経路トレース・メッセージの経路トレース・メッセージ・データの複写です。

経路トレース応答メッセージ形式

経路トレース応答メッセージは、メッセージ記述子とメッセージ・データを含む標準 IBM MQ メッセージです。メッセージ・データには、経路トレース・メッセージがキュー・マネージャー・ネットワーク経由で送信されたときに、このメッセージに対して実行されたアクティビティについての情報が含まれています。

経路トレース応答メッセージには以下の情報が含まれています。

メッセージ記述子

MQMD 構造

メッセージ・データ

PCF ヘッダー (MQCFH) および経路トレース応答メッセージ・データ

経路トレース応答メッセージ・データは 1 つ以上の *Activity* PCF グループで構成されます。

経路トレース・メッセージがそのターゲット・キューに到着したとき、経路トレース・メッセージからのアクティビティ情報のコピーを含む経路トレース応答メッセージを生成できます。経路トレース応答メッセージは、応答先キューまたはシステム・キューに送信されます。

[137 ページの表 21](#) は、一定の条件下でのみ返されるパラメーターを含む、経路トレース応答メッセージの構造を示しています。

表 21. 経路トレース応答メッセージ形式

MQMD 構造	PCF ヘッダー MQCFH 構造体	経路トレース応答メッセージ・データ
構造体 ID 構造バージョン レポート・オプション メッセージ・タイプ 満了時刻 Feedback Encoding コード化文字セット ID メッセージ形式 優先順位 Persistence メッセージ ID 関連 ID バックアウト・カウント 応答先キュー 応答先キュー・マネージャー ユーザー ID アカウンティング・トークン アプリケーション識別データ アプリケーション・タイプ アプリケーション名 書き込み日付 書き込み時刻 発生元アプリケーション・データ グループ ID メッセージ順序番号 オフセット メッセージ・フラグ 元の長さ	PCF ヘッダー (MQCFH) 構造タイプ 構造の長さ 構造バージョン コマンド ID メッセージ順序番号 制御オプション 完了コード 理由コード パラメーター数	アクティビティ アクティビティ・アプリケーション名 アクティビティ・アプリケーション・タイプ アクティビティの記述 Operation 操作タイプ 操作日付 操作時刻 メッセージ メッセージ長 MQMD EmbeddedMQMD キュー・マネージャー名 キュー共有グループ名 キュー名 ^{1 2 3} 解決されたキュー名 ^{1 3} リモート・キュー名 ³ リモート・キュー・マネージャー 2 3 4 5 フィードバック ² チャンネル名 ^{4 5} チャンネル・タイプ ^{4 5} 伝送キュー名 ⁵ TraceRoute 詳細 記録されたアクティビティ 記録されていないアクティビティ 不連続カウント 最大アクティビティ数 累積 Deliver

注:

1. Get 操作および Browse 操作に対して返されます。
2. Discard 操作に対して返されます。
3. Put 操作、Put Reply 操作、および Put Report 操作に対して返されます。
4. Receive 操作に対して返されます。
5. Send 操作に対して返されます。

経路トレース応答メッセージ MQMD (メッセージ記述子)

このページを使用して、経路トレース応答メッセージの MQMD 構造に含まれた値を確認します。

経路トレース応答メッセージの場合、MQMD 構造には『アクティビティ・レポート・メッセージ記述子』で説明されている各種パラメーターが含まれています。経路トレース応答メッセージ記述子の一部のパラメーター値は、以下のように、アクティビティ・レポート・メッセージ記述子のパラメーター値とは異なっています。

MsgType

説明: メッセージのタイプ。
データ型: MQLONG
値: **MQMT_REPLY**

Feedback

説明: フィードバックまたは理由コード。
データ型: MQLONG
値: **MQFB_NONE**

Encoding

説明: メッセージ・データの数値エンコード。
データ型: MQLONG
値: 経路トレース・メッセージ記述子からコピーされます。

CodedCharSetId

説明: メッセージ・データの文字セット ID。
データ型: MQLONG
値: 経路トレース・メッセージ記述子からコピーされます。

Format

説明: メッセージ・データの形式名。
データ型: MQCHAR8.
値: **MQFMT_ADMIN**
 管理メッセージ。

経路トレース応答メッセージ MQCFH (PCF ヘッダー)

このページを使用して、経路トレース応答メッセージの MQCFH 構造に含まれた PCF 値を確認します。

経路トレース応答メッセージの PCF ヘッダー (MQCFH) は、経路トレース・メッセージの PCF ヘッダーと同じです。

経路トレース応答メッセージ・データ

経路トレース応答メッセージ・データは、そのデータが生成された経路トレース・メッセージの経路トレース・メッセージ・データの複写です。

経路トレース応答メッセージ・データには 1 つ以上の *Activity* グループが含まれています。パラメーターについては、[109 ページの『アクティビティ・レポート・メッセージ・データ』](#)に説明があります。

アカウンティング・メッセージと統計メッセージ

キュー・マネージャーは、アカウンティング・メッセージと統計メッセージを生成して、IBM MQ アプリケーションによって実行された MQI 操作に関する情報を記録するか、または IBM MQ システムで実行されたアクティビティに関する情報を記録します。

アカウンティング・メッセージ

アカウンティング・メッセージは、IBM MQ アプリケーションによって実行される MQI 操作についての情報を記録するために使用されます。[139 ページの『アカウンティング・メッセージ』](#)を参照してください。

統計メッセージ

統計メッセージは、IBM MQ システムで発生するアクティビティーについての情報を記録するために使用されます。[142 ページの『統計メッセージ』](#)を参照してください。統計メッセージに記録されたアクティビティーの中には、内部キュー・マネージャー操作に関連するものがあります。

z/OS アカウンティング・メッセージと統計メッセージは、以下で説明されているように IBM MQ for z/OS では使用できませんが、同等の機能はシステム管理機能 (SMF) によって使用可能です。

アカウンティング・メッセージと統計メッセージは、2つのシステム・キューのいずれかに送られます。ユーザー・アプリケーションは、これらのシステム・キューからメッセージを取り出し、記録された情報を以下のようなさまざまな目的で使用できます。

- アプリケーション・リソースの使用を報告する。
- アプリケーション・アクティビティーを記録する。
- キャパシティー・プランニングを行う。
- キュー・マネージャー・ネットワークにある問題を検出する。
- キュー・マネージャー・ネットワークにある問題の原因の判別を支援する。
- キュー・マネージャー・ネットワークの効率を向上させる。
- キュー・マネージャー・ネットワークの実行方法を習得できるようにする。
- キュー・マネージャー・ネットワークが正しく実行されていることを確認する。

関連概念

[323 ページの『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.

アカウンティング・メッセージ

アカウンティング・メッセージは、IBM MQ アプリケーションによって実行される MQI 操作についての情報を記録します。アカウンティング・メッセージは、数多くの PCF 構造を含む PCF メッセージです。

アプリケーションがキュー・マネージャーから切断されると、アカウンティング・メッセージが生成され、システム・アカウンティング・キュー (SYSTEM.ADMIN.ACCOUNTING.QUEUE) に送信されます。長期間実行される IBM MQ アプリケーションでは、以下の場合に中間アカウンティング・メッセージが生成されます。

- 接続が確立されてから以降の時間が、構成された間隔を超えるとき。
- 最後の中間アカウンティング・メッセージ以降の時間が、構成された間隔を超えるとき。

アカウンティング・メッセージは、以下のカテゴリーに分類されます。

MQI アカウンティング・メッセージ

MQI アカウンティング・メッセージには、キュー・マネージャーへの接続を使用して行われた MQI 呼び出しの数に関する情報が含まれています。

キュー・アカウンティング・メッセージ

キュー・アカウンティング・メッセージには、キュー・マネージャーへの接続を使用して行われた MQI 呼び出しの数に関する情報が、キューごとにグループ化されて含まれています。

各キュー・アカウンティング・メッセージには最大 100 のレコードが含まれます。レコードはそれぞれ、特定のキューについてアプリケーションによって実行されたアクティビティーに関連しています。

アカウンティング・メッセージは、ローカル・キューについてのみ記録されます。アプリケーションが別名キューに対して MQI 呼び出しを行う場合、アカウンティング・データは基本キューに対して記録され、リモート・キューの場合、アカウンティング・データは伝送キューに対して記録されます。

注: IBM MQ アプリケーションによって実行される MQI 操作に関連する情報には、ストリーミング・キューに関連する操作は含まれません。ただし、その操作がアプリケーションによってそのキューに直接行われる場合を除きます。

関連資料

158 ページの『MQI アカウンティング・メッセージ・データ』

このページを使用して、MQI アカウンティング・メッセージの構造を確認します。

169 ページの『キュー・アカウンティング・メッセージ・データ』

このページを使用して、キュー・アカウンティング・メッセージの構造を確認します。

アカウンティング・メッセージの形式

アカウンティング・メッセージは、メッセージ記述子とメッセージ・データから成る一連の PCF フィールドで構成されています。

メッセージ記述子

- ・ アカウンティング・メッセージ MQMD (メッセージ記述子)

アカウンティング・メッセージ・データ

- ・ アカウンティング・メッセージ MQCFH (PCF ヘッダー)
- ・ 常に返されるアカウンティング・メッセージ・データ
- ・ 可能な場合に返されるアカウンティング・メッセージ・データ

アカウンティング・メッセージ MQCFH (PCF ヘッダー) には、アプリケーションについての情報と、アカウンティング・データが記録された間隔についての情報が含まれています。

アカウンティング・メッセージ・データは、アカウンティング情報を保管する PCF パラメーターから構成されます。アカウンティング・メッセージの内容は、メッセージのカテゴリによって以下のように異なります。

MQI アカウンティング・メッセージ

MQI アカウンティング・メッセージ・データは、いくつかの PCF パラメーターから構成されますが、PCF グループは含まれません。

キュー・アカウンティング・メッセージ

キュー・アカウンティング・メッセージ・データは、いくつかの PCF パラメーターと、1 から 100 個の範囲の *QAccountingData* PCF グループから構成されます。

アカウンティング・データが収集されたキューごとに、1 つの *QAccountingData* PCF グループがあります。アプリケーションが 100 を超えるキューにアクセスする場合は、複数のアカウンティング・メッセージが生成されます。各メッセージの MQCFH (PCF ヘッダー) にある *SeqNumber* は適宜更新され、シーケンスの最終メッセージの MQCFH には *MQCFH_LAST* と指定された *Control* パラメーターがあります。

アカウンティング情報の収集

キューとキュー・マネージャー属性を使用して、アカウンティング情報の収集を制御します。また、MQCONNX オプションを使用して、接続レベルで収集を制御することもできます。

MQI アカウンティング情報の収集の制御

キュー・マネージャー属性 ACCTMQI を使用して、MQI アカウンティング情報の収集を制御します。

この属性の値は、ACCTMQI パラメーターを指定した MQSC コマンド ALTER QMGR を使用して変更できます。アカウンティング・メッセージは、アカウンティングが使用可能になった後に開始された接続に対してのみ生成されます。ACCTMQI パラメーターには、以下の値を指定できます。

ON

キュー・マネージャーに対するすべての接続について MQI アカウンティング情報が収集されます。

OFF

MQI アカウンティング情報は収集されません。これがデフォルト値です。

例えば、MQI アカウンティング情報収集を有効にするには、次の MQSC コマンドを使用します。

```
ALTER QMGR ACCTMQI(ON)
```


キュー・アカウンティング情報

キュー属性 ACCTQ およびキュー・マネージャー属性 ACCTQ を使用して、キュー・アカウンティング情報の収集を制御します。

キュー属性の値を変更するには、MQSC コマンド ALTER QLOCAL を使用し、パラメーター ACCTQ を指定します。アカウンティング・メッセージは、アカウンティングが使用可能になった後に開始された接続に対してのみ生成されます。この値の変更は、属性を変更した後に発生するキュー・マネージャーへの接続でのみ有効であることに注意してください。

キュー属性 ACCTQ には、以下の値を指定できます。

ON

このキューをオープンするキュー・マネージャーに対するすべての接続について、キュー・アカウンティング情報が収集されます。

OFF

このキューのキュー・アカウンティング情報は収集されません。

QMGR

このキューのキュー・アカウンティング情報の収集は、キュー・マネージャー属性 ACCTQ の値に従って制御されます。これがデフォルト値です。

キュー・マネージャー属性の値は、ACCTQ パラメーターを指定した MQSC コマンド ALTER QMGR を使用して変更できます。キュー・マネージャー属性 ACCTQ には、以下の値を指定できます。

ON

キュー属性 ACCTQ が QMGR に設定されたキューのキュー・アカウンティング情報が収集されます。

OFF

キュー属性 ACCTQ が QMGR に設定されたキューのキュー・アカウンティング情報は収集されません。これがデフォルト値です。

NONE

キュー属性 ACCTQ とは無関係に、すべてのキューのキュー・アカウンティング情報の収集が無効になります。

キュー・マネージャー属性 ACCTQ が NONE に設定されている場合は、キュー属性 ACCTQ とは無関係に、すべてのキューのキュー・アカウンティング情報の収集が無効になります。

例えば、キュー Q1 のアカウンティング情報収集を有効にするには、次の MQSC コマンドを使用します。

```
ALTER QLOCAL(Q1) ACCTQ(ON)
```

キュー属性 ACCTQ に QMGR を指定したすべてのキューのアカウンティング情報収集を有効にするには、次の MQSC コマンドを使用します。

```
ALTER QMGR ACCTQ(ON)
```

アカウンティング情報収集の制御用の MQCONNX オプション

MQCONNX 呼び出しで **ConnectOpts** パラメーターを使用して、MQI アカウンティング情報とキュー・アカウンティング情報の収集の両方を接続レベルで変更します。これは、キュー・マネージャー属性の ACCTMQI と ACCTQ の有効値を指定変更することによって行います。

ConnectOpts パラメーターには、以下の値を指定できます。

MQCNO_ACCOUNTING_MQI_ENABLED

キュー・マネージャー属性 ACCTMQI の値に OFF が指定された場合、この接続の MQI アカウンティングは有効になります。これは、キュー・マネージャー属性 ACCTMQI に ON を指定するのと同様です。

キュー・マネージャー属性 ACCTMQI の値に OFF が指定されない場合、この属性には効果がありません。

MQCNO_ACCOUNTING_MQI_DISABLED

キュー・マネージャー属性 ACCTMQI の値に ON が指定された場合、この接続の MQI アカウンティングは無効になります。これは、キュー・マネージャー属性 ACCTMQI に OFF を指定するのと同様です。

キュー・マネージャー属性 ACCTMQI の値に ON が指定されない場合、この属性には効果がありません。

MQCNO_ACCOUNTING_Q_ENABLED

キュー・マネージャー属性 ACCTQ の値に OFF が指定された場合、この接続のキュー・アカウンティングは有効になります。ACCTQ に QMGR が指定されたキューはすべて、キュー・アカウンティングが有効になります。これは、キュー・マネージャー属性 ACCTQ に ON を指定するのと同様です。

キュー・マネージャー属性 ACCTQ の値に OFF が指定されない場合、この属性には効果がありません。

MQCNO_ACCOUNTING_Q_DISABLED

キュー・マネージャー属性 ACCTQ の値に ON が指定された場合、この接続のキュー・アカウンティングは無効になります。これは、キュー・マネージャー属性 ACCTQ に OFF を指定するのと同様です。

キュー・マネージャー属性 ACCTQ の値に ON が指定されない場合、この属性には効果がありません。

これらの指定変更は、デフォルトでは無効になっています。これらを有効にするには、キュー・マネージャー属性 ACCTCONO を ENABLED に設定します。個々の接続に対してアカウンティング・オーバーライドを使用可能にするには、以下の MQSC コマンドを使用します。

```
ALTER QMGR ACCTCONO(ENABLED)
```

アプリケーション切断時のアカウンティング・メッセージ生成

アプリケーションがキュー・マネージャーから切断されると、アカウンティング・メッセージが生成されます。また、長期間実行されている IBM MQ アプリケーションについても、中間アカウンティング・メッセージが書き込まれます。

アプリケーションが切断されると、アカウンティング・メッセージは以下のいずれかの方法で生成されます。

- アプリケーションは MQDISC 呼び出しを発行します。
- キュー・マネージャーは、アプリケーションが終了したことを認識します。

接続が確立された後、または最後に中間アカウンティング・メッセージが書き込まれた後で、間隔が構成された間隔を超えたとき、長期間実行されている IBM MQ アプリケーションについて中間アカウンティング・メッセージが書き込まれます。キュー・マネージャー属性 ACCTINT で、中間アカウンティング・メッセージの書き込みが可能になるまでの時間 (秒) を指定します。アカウンティング・メッセージは、アプリケーションがキュー・マネージャーと相互作用したときのみ生成されるため、MQI 要求を実行することなく長期間キュー・マネージャーに接続したままのアプリケーションは、アカウンティング間隔の完了後に最初の MQI 要求が実行されるまでアカウンティング・メッセージを生成しません。

デフォルトのアカウンティング間隔は 1800 秒 (30 分) です。例えば、アカウンティング間隔を 900 秒 (15 分) に変更するには、次の MQSC コマンドを使用します。

```
ALTER QMGR ACCTINT(900)
```

統計メッセージ

統計メッセージは、IBM MQ システムで発生するアクティビティーについての情報を記録します。統計メッセージは、数多くの PCF 構造を含む PCF メッセージです。

統計メッセージは、アクティビティーがある限り、構成された間隔でシステム・キュー (SYSTEM.ADMIN.STATISTICS.QUEUE) に送信されます。

統計メッセージは、以下のカテゴリーに分類されます。

MQI 統計メッセージ

MQI 統計メッセージには、構成された間隔中に行われた MQI 呼び出しの数に関する情報が入っています。この情報には、例えばキュー・マネージャーによって発行された MQI 呼び出しの数が含まれます。

キュー統計メッセージ

キュー統計メッセージには、構成された間隔中のキューのアクティビティーに関する情報が入っています。この情報には、キューに入れられたメッセージ数、キューから取り出されたメッセージ数、およびキューで処理された総バイト数が含まれます。

各キュー統計メッセージには最大 100 のレコードが含まれます。レコードはそれぞれ、統計が収集されたキューごとのアクティビティーに関連しています。

統計メッセージは、ローカル・キューについてのみ記録されます。アプリケーションが別名キューに対して MQI 呼び出しを行う場合、統計データは基本キューに対して記録され、リモート・キューの場合、統計データは伝送キューに対して記録されます。

チャンネル統計メッセージ

チャンネル統計メッセージには、構成された間隔中のチャンネルのアクティビティーに関する情報が入っています。この情報には、例えばチャンネルによって転送されたメッセージ数や、チャンネルによって転送されたバイト数があります。

各チャンネル統計メッセージには最大 100 のレコードが含まれます。レコードはそれぞれ、統計が収集されたチャンネルごとのアクティビティーに関連しています。

注: 情報は IBM MQ システムで発生するアクティビティーに関連するため、情報にはストリーミング・キューに関連する操作が含まれます。

関連資料

144 ページの『MQI 統計収集の制御』

キュー・マネージャー属性 STATMQI を使用して、MQI 統計情報の収集を制御します。

144 ページの『キュー統計収集の制御』

キュー属性 STATQ およびキュー・マネージャー属性 STATQ を使用して、キュー統計情報の収集を制御します。

145 ページの『チャンネル統計収集の制御』

チャンネル属性 STATCHL を使用して、チャンネル統計情報の収集を制御します。また、キュー・マネージャー属性を設定して情報収集を制御することもできます。

統計メッセージ形式

統計メッセージは、メッセージ記述子とメッセージ・データから成る一連の PCF フィールドで構成されています。

メッセージ記述子

- 統計メッセージ MQMD (メッセージ記述子)

アカウントिंग・メッセージ・データ

- 統計メッセージ MQCFH (PCF ヘッダー)
- 常に返される統計メッセージ・データ
- 可能な場合に返される統計メッセージ・データ

統計メッセージ MQCFH (PCF ヘッダー) には、統計データが記録された間隔についての情報が含まれています。

統計メッセージ・データは、統計情報を保管する PCF パラメーターから構成されます。統計メッセージの内容は、メッセージのカテゴリによって以下のように異なります。

MQI 統計メッセージ

MQI 統計メッセージ・データは、いくつかの PCF パラメーターから構成されますが、PCF グループは含まれません。

キュー統計メッセージ

キュー統計メッセージ・データは、いくつかの PCF パラメーターと、1 から 100 個の範囲の *QStatisticsData* PCF グループから構成されます。

間隔内でアクティブだったキューごとに、1 つの *QStatisticsData* PCF グループがあります。間隔内で 100 を超えるキューがアクティブだった場合は、複数の統計メッセージが生成されます。各メッセー

ジの MQCFH (PCF ヘッダー) にある *SeqNumber* は適宜更新され、シーケンスの最終メッセージの MQCFH には MQCFC_LAST と指定された *Control* パラメーターがあります。

チャンネル統計メッセージ

チャンネル統計メッセージ・データは、いくつかの PCF パラメーターと、1 から 100 個の範囲の *ChlStatisticsData* PCF グループから構成されます。

間隔内でアクティブだったチャンネルごとに、1 つの *ChlStatisticsData* PCF グループがあります。間隔内で 100 を超えるチャンネルがアクティブだった場合は、複数の統計メッセージが生成されます。各メッセージの MQCFH (PCF ヘッダー) にある *SeqNumber* は適宜更新され、シーケンスの最終メッセージの MQCFH には MQCFC_LAST と指定された *Control* パラメーターがあります。

統計情報収集

キュー属性、キュー・マネージャー属性、およびチャンネル属性を使用して、統計情報の収集を制御します。

MQI 統計収集の制御

キュー・マネージャー属性 *STATMQI* を使用して、MQI 統計情報の収集を制御します。

この属性の値は、MQSC コマンド ALTER QMGR にパラメーター **STATMQI** を指定して変更できます。統計メッセージは、統計収集が使用可能になった後でオープンされたキューについてのみ生成されます。

STATMQI パラメーターには、以下の値を指定できます。

ON

キュー・マネージャーに対するすべての接続について MQI 統計情報が収集されます。

OFF

MQI 統計情報は収集されません。これがデフォルト値です。

例えば、MQI 統計情報収集を有効にするには、次の MQSC コマンドを使用します。

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

キュー統計収集の制御

キュー属性 *STATQ* およびキュー・マネージャー属性 *STATQ* を使用して、キュー統計情報の収集を制御します。

キュー統計情報の収集は、個々のキューについて、または複数のキューについて有効または無効にすることができます。個々のキューを制御するには、キュー属性 *STATQ* を設定します。キュー・マネージャー属性 *STATQ* を使用して、キュー・マネージャー・レベルでキュー統計情報収集を有効または無効にします。キュー属性 *STATQ* に値 QMGR が指定されているどのキューについても、キュー統計情報収集はキュー・マネージャー・レベルで制御されます。

キュー統計は、統計収集が使用可能になった後で開かれた IBM MQMQI オブジェクト・ハンドルを使用する操作についてのみ増分されます。

キュー統計メッセージは、以前の時間枠で収集された統計データのキューについてのみ生成されます。

同じキューに、複数のオブジェクト・ハンドルが関係する複数の PUT 操作および GET 操作が存在する可能性があります。一部のオブジェクト・ハンドルは統計収集が使用可能になる前に開かれ、他のものは使用可能になった後に開かれたかもしれません。そのため、キュー統計では、すべてではなく、一部の PUT 操作および GET 操作のアクティビティーが記録される可能性があります。

キュー統計がすべてのアプリケーションのアクティビティーを確実に記録するようにするには、モニター対象キューのオブジェクト・ハンドルを閉じてから新しく再オープンする必要があります。これを実行する最善の方法は、統計収集を使用可能にした後に、すべてのアプリケーションを終了してから再始動する方法です。

キュー属性 *STATQ* の値は、*STATQ* パラメーターを指定した MQSC コマンド ALTER QLOCAL を使用して変更できます。キュー属性 *STATQ* には、以下の値を指定できます。

ON

キューをオープンするキュー・マネージャーに対するすべての接続について、キュー統計情報が収集されます。

OFF

このキューのキュー統計情報は収集されません。

QMGR

このキューのキュー統計情報の収集は、キュー・マネージャー属性 STATQ の値に従って制御されます。これがデフォルト値です。

キュー・マネージャー属性 STATQ の値は、MQSC コマンド ALTER QMGR にパラメーター STATQ を指定して変更できます。キュー・マネージャー属性 STATQ には、以下の値を指定できます。

ON

キュー属性 STATQ が QMGR に設定されたキューのキュー統計情報が収集されます。

OFF

キュー属性 STATQ が QMGR に設定されたキューのキュー統計情報は収集されません。これがデフォルト値です。

NONE

キュー属性 STATQ とは無関係に、すべてのキューのキュー統計情報の収集が無効になります。

キュー・マネージャー属性 STATQ が NONE に設定されている場合は、キュー属性 STATQ とは無関係に、すべてのキューのキュー統計情報の収集が無効になります。

例えば、キュー Q1 の統計情報収集を有効にするには、次の MQSC コマンドを使用します。

```
ALTER QLOCAL(Q1) STATQ(ON)
```

キュー属性 STATQ に QMGR を指定したすべてのキューの統計情報収集を有効にするには、次の MQSC コマンドを使用します。

```
ALTER QMGR STATQ(ON)
```

チャンネル統計収集の制御

チャンネル属性 STATCHL を使用して、チャンネル統計情報の収集を制御します。また、キュー・マネージャー属性を設定して情報収集を制御することもできます。

チャンネル統計情報の収集は、個々のチャンネルについて、または複数のチャンネルについて有効または無効にすることができます。個々のチャンネルを制御するには、チャンネル属性 STATCHL を設定して、チャンネル統計情報収集を有効または無効にする必要があります。複数のチャンネルを同時に制御するには、キュー・マネージャー属性 STATCHL を使用すれば、キュー・マネージャー・レベルでチャンネル統計情報収集を有効または無効にできます。チャンネル属性 STATCHL に値 QMGR が指定されているどのチャンネルについても、チャンネル統計情報収集はキュー・マネージャー・レベルで制御されます。

自動定義されたクラスター送信側チャンネルは IBM MQ オブジェクトではないため、チャンネル・オブジェクトが持っているような属性は持っていません。自動定義されたクラスター送信側チャンネルを制御するには、キュー・マネージャー属性 STATACLS を使用します。この属性により、キュー・マネージャー内で自動定義されたクラスター送信側チャンネルのチャンネル統計情報収集を有効にするか無効にするかが決まります。

チャンネル統計情報収集は、低、中、または高の 3 つのモニター・レベルのいずれかに設定できます。モニター・レベルは、オブジェクト・レベルまたはキュー・マネージャー・レベルのいずれかに設定できます。どのレベルを使用するかは、ご使用のシステムによって異なります。統計情報データ収集には、計算に相対的に負荷の高い命令がいくつか必要になる場合があるため、チャンネル統計情報収集の影響を低減するために、中および低のモニター・オプションでは、常にデータを収集するのではなく、一定の間隔でデータのサンプルを測定します。[146 ページの表 22](#) に、チャンネル統計情報収集で使用可能なレベルの要約を示します。

表 22. チャネル統計情報収集の詳細レベル

レベル	説明	使用法
低	少数のデータ・サンプルを通常の間隔で測定します。	大量のメッセージを処理するオブジェクトで使用されます。
中	データ・サンプルを通常の間隔で測定します。	ほとんどのオブジェクトで使用されます。
高	すべてのデータを通常の間隔で測定します。	最新情報が重要である場合に、秒あたりに少数のメッセージのみを処理するオブジェクトで使用されます。

チャネル属性 `STATCHL` の値は、`STATCHL` パラメーターを指定した MQSC コマンド `ALTER CHANNEL` を使用して変更できます。

キュー・マネージャー属性 `STATCHL` の値は、MQSC コマンド `ALTER QMGR` にパラメーター `STATCHL` を指定して変更できます。

キュー・マネージャー属性 `STATACLS` の値は、MQSC コマンド `ALTER QMGR` にパラメーター `STATACLS` を指定して変更できます。

チャネル属性 `STATCHL` には、以下の値を指定できます。

LOW

低い詳細レベルでチャネル統計情報が収集されます。

MEDIUM

中程度の詳細レベルでチャネル統計情報が収集されます。

HIGH

高い詳細レベルでチャネル統計情報が収集されます。

OFF

このチャネルのチャネル統計情報は収集されません。

QMGR

チャネル属性は `QMGR` に設定されます。このチャネルの統計情報の収集は、キュー・マネージャー属性 `STATCHL` の値によって制御されます。

これがデフォルト値です。

z/OS の上 z/OS システムでは、このパラメータを有効にすると、選択した値に関係なく、統計データの収集がオンになるだけです。LOW、MEDIUM、または HIGH のどれを指定しても、結果に違いはありません。チャネル・アカウント・レコードを収集するには、このパラメータを有効にしなければなりません。

キュー・マネージャー属性 `STATCHL` には、以下の値を指定できます。

LOW

チャネル属性 `STATCHL` が `QMGR` に設定されたすべてのチャネルについて、チャネル統計情報が低い詳細レベルで収集されます。

MEDIUM

チャネル属性 `STATCHL` が `QMGR` に設定されたすべてのチャネルについて、チャネル統計情報が中程度の詳細レベルで収集されます。

HIGH

チャネル属性 `STATCHL` が `QMGR` に設定されたすべてのチャネルについて、チャネル統計情報が高い詳細レベルで収集されます。

OFF

チャネル属性 `STATCHL` が `QMGR` に設定されたどのチャネルについても、チャネル統計情報は収集されません。

これがデフォルト値です。

NONE

チャンネル属性 **STATCHL** とは無関係に、すべてのチャンネルのチャンネル統計情報の収集が無効になります。

z/OS の上 z/OS システムでは、このパラメータを有効にすると、選択した値に関係なく、統計データの収集がオンになるだけです。 **LOW**、**MEDIUM**、または **HIGH** のどれを指定しても、結果に違いはありません。チャンネル・アカウンティング・レコードを収集するには、このパラメータを有効にしなければなりません。

キュー・マネージャー属性 **STATACLS** には、以下の値を指定できます。

LOW

自動定義されたクラスター送信側チャンネルの統計情報が、低い詳細レベルで収集されます。

MEDIUM

自動定義されたクラスター送信側チャンネルの統計情報が、中程度の詳細レベルで収集されます。

HIGH

自動定義されたクラスター送信側チャンネルの統計情報が、高い詳細レベルで収集されます。

OFF

自動定義されたクラスター送信側チャンネルの統計情報は収集されません。

QMGR

自動定義されたクラスター送信側チャンネルの統計情報の収集は、キュー・マネージャー属性 **STATCHL** の値によって制御されます。

これがデフォルト値です。

z/OS の上 z/OS システムでは、このパラメータを有効にすると、選択した値に関係なく、統計データの収集がオンになるだけです。 **LOW**、**MEDIUM**、または **HIGH** のどれを指定しても、結果に違いはありません。チャンネル・アカウンティング・レコードを収集するには、このパラメータを有効にしなければなりません。

例えば、送信側チャンネル **QM1.TO.QM2** の統計情報収集を中程度の詳細レベルで有効にするには、次の **MQSC** コマンドを使用します。

```
ALTER CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) STATCHL(MEDIUM)
```

チャンネル属性 **STATCHL** に **QMGR** を指定したすべてのチャンネルの統計情報収集を中程度の詳細レベルで有効にするには、次の **MQSC** コマンドを使用します。

```
ALTER QMGR STATCHL(MEDIUM)
```

すべての自動定義されたクラスター送信側チャンネルの統計情報収集を中程度の詳細レベルで有効にするには、次の **MQSC** コマンドを使用します。

```
ALTER QMGR STATACLS(MEDIUM)
```

統計メッセージ生成

統計メッセージは、構成された間隔で、またキュー・マネージャーが制御された方法でシャットダウンされるときに生成されます。

構成された間隔は、**STATINT** キュー・マネージャー属性によって制御されます。この属性では、統計メッセージ生成の間隔を秒単位で指定します。デフォルトの統計間隔は 1800 秒 (30 分) です。統計間隔を変更するには、**MQSC** コマンド **ALTER QMGR** を使用し、**STATINT** パラメータを指定します。例えば、統計間隔を 900 秒 (15 分) に変更するには、次の **MQSC** コマンドを使用します。

```
ALTER QMGR STATINT(900)
```

現在収集されている統計データを、統計収集間隔が終了する前に統計キューに書き込むには、MQSC コマンド RESET QMGR TYPE(STATISTICS) を使用できます。このコマンドを発行すると、収集された統計データが統計キューに書き込まれ、新たな統計データ収集間隔が開始されます。

アカウント情報と統計情報の表示

アカウント情報・メッセージと統計メッセージに記録された情報を使用するには、**amqsmon** サンプル・プログラムなどのアプリケーションを実行して、記録された情報を適切な形式に変換します。

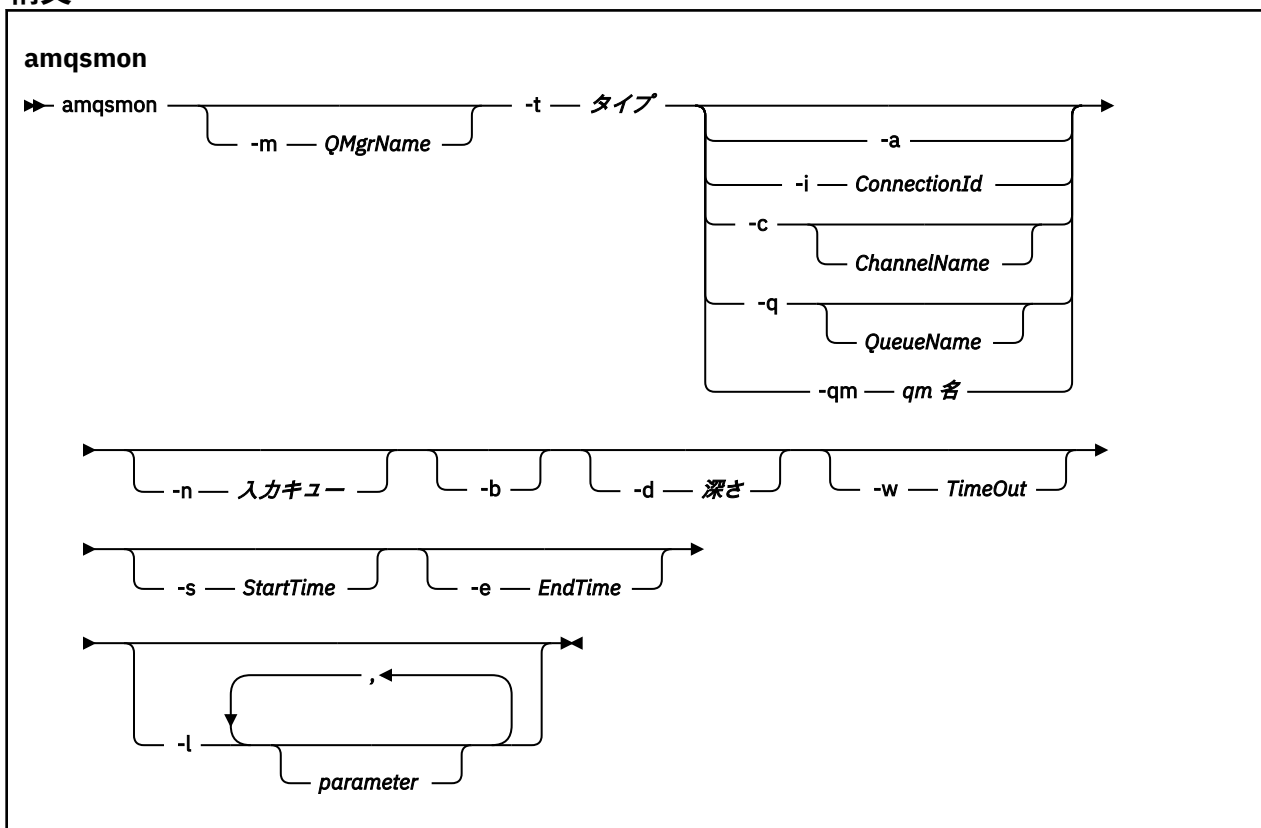
アカウント情報・メッセージと統計メッセージは、システムのアカウント情報・キューと統計キューに書き込まれます。**amqsmon** は、IBM MQ に用意されたサンプル・プログラムです。このサンプル・プログラムは、アカウント情報・キューおよび統計キューのメッセージを処理し、読み取り可能な形式で情報を画面に表示します。

amqsmon はサンプル・プログラムであるため、提供されたソース・コードを、アカウント情報・メッセージまたは統計メッセージを処理する独自のアプリケーションを作成するためのテンプレートとして使用できます。あるいは、**amqsmon** のソース・コードを、特定の要件に合わせて変更することもできます。

amqsmon (フォーマット済みモニター情報の表示)

amqsmon サンプル・プログラムを使用して、アカウント情報・メッセージと統計メッセージ内に含まれた情報を読み取り可能なフォーマットで表示します。**amqsmon** プログラムでは、アカウント情報・メッセージをアカウント情報・キュー SYSTEM.ADMIN.ACCOUNTING.QUEUE から読み取ります。また、統計メッセージを統計キュー SYSTEM.ADMIN.STATISTICS.QUEUE から読み取ります。

構文



必要なパラメーター

-t Type

処理するメッセージのタイプ。Type には以下のいずれかを指定します。

アカウントティング

アカウントティング・レコードが処理されます。メッセージは、システム・キュー SYSTEM.ADMIN.ACCOUNTING.QUEUE (読み取り元の特定のキューを選択するために `-n` パラメーターを使用した場合を除く)。

統計

統計レコードが処理されます。メッセージは、システム・キュー

SYSTEM.ADMIN.STATISTICS.QUEUE **V 9.4.0** (`-n` パラメーターを使用して読み取り元の特定のキューを選択した場合を除く)。

オプション・パラメーター

-m QMgrName

アカウントティング・メッセージまたは統計メッセージが処理されるキュー・マネージャーの名前。

このパラメーターを指定しない場合は、デフォルトのキュー・マネージャーが使用されます。

-a

MQI レコードを含むメッセージのみが処理されます。

MQI レコードのみが表示されます。MQI レコードを含まないメッセージは常に、そのメッセージが読み取られたキューに残されます。

-q QueueName

QueueName はオプション・パラメーターです。

QueueName を指定しない場合: キュー・アカウントティング・レコードとキュー統計レコードのみが表示されます。

QueueName を指定した場合: *QueueName* で指定されたキューのキュー・アカウントティング・レコードとキュー統計レコードのみが表示されます。

`-b` を指定しない場合、レコードが入っていたアカウントティング・メッセージと統計メッセージは廃棄されます。アカウントティング・メッセージと統計メッセージには他のキューからのレコードが入ることがあるため、`-b` を指定しないと、まだ表示されていないレコードが廃棄される場合があります。

-c ChannelName

ChannelName はオプション・パラメーターです。

ChannelName を指定しない場合: チャンネル統計レコードのみが表示されます。

ChannelName を指定した場合: *ChannelName* で指定されたチャンネルのチャンネル統計レコードのみが表示されます。

`-b` を指定しない場合、レコードが入っていた統計メッセージは廃棄されます。統計メッセージには他のチャンネルからのレコードが入ることがあるため、`-b` を指定しないと、まだ表示されていないレコードが廃棄される場合があります。

このパラメーターは、統計メッセージのみを表示するときに使用できます (`-t statistics`)。

V 9.4.0

-qm qm 名

このパラメーターはオプションですが、指定する場合はキュー・マネージャー名を指定する必要があります

指定されたキュー・マネージャーによって作成されたアカウントリング・メッセージまたは統計メッセージのみをキューから読み取ることを指定します。通常、複数のキュー・マネージャーからのメッセージが中央キュー・マネージャーに転送された場合にのみ使用されます。

-i ConnectionId

ConnectionId で指定された接続 ID に関連するレコードのみが表示されます。

このパラメーターは、アカウントリング・メッセージのみを表示するときに使用できます (*-t accounting*)。

-b を指定しない場合、レコードが入っていた統計メッセージは廃棄されます。統計メッセージには他のチャンネルからのレコードが入ることがあるため、*-b* を指定しないと、まだ表示されていないレコードが廃棄される場合があります。

V 9.4.0

-n 入力キュー

アカウントリング・メッセージまたは統計メッセージの読み取り元のキュー。これが指定されていない場合は、選択された *-t <type>* のデフォルト・キューが使用されます。

-b

メッセージをブラウズします。

メッセージは非破壊的に取り出されます。

-d Depth

処理可能な最大メッセージ数。

このパラメーターを指定しない場合、処理可能なメッセージ数に制限はありません。

-w Timeout

メッセージが使用可能になるのを待機する最大秒数。

このパラメーターを指定しない場合、*amqsmon* は一旦終了するため、これ以上メッセージは処理されません。

-s StartTime

指定された *StartTime* の後に書き込まれたメッセージのみが処理されます。

StartTime は *yyyy-mm-dd hh.mm.ss* の形式で指定します。時刻なしで日付を指定した場合、時刻は指定した日の *00.00.00* にデフォルト設定されます。時刻は GMT 形式です。

このパラメーターを指定しない場合の効果については、[注 1](#) を参照してください。

-e EndTime

指定された *EndTime* の前に書き込まれたメッセージのみが処理されます。

EndTime は *yyyy-mm-dd hh.mm.ss* の形式で指定します。時刻なしで日付を指定した場合、時刻は指定した日の *23.59.59* にデフォルト設定されます。時刻は GMT 形式です。

このパラメーターを指定しない場合の効果については、[注 1](#) を参照してください。

-l Parameter

処理対象のレコードから選択したフィールドのみが表示されます。 *Parameter* は整数値のコンマ区切りリストで、整数値はそれぞれフィールドの数値定数にマッピングされます。[amqsmon の例 5](#) を参照してください。

このパラメーターを指定しない場合は、使用可能なすべてのフィールドが表示されます。

注:

1. *-s StartTime* または *-e EndTime* を指定しない場合、処理可能なメッセージは書き込み時刻による制限は受けません。

amqsmon (フォーマット済みモニター情報の表示) の例

このページを使用して、*amqsmon* (フォーマット済みモニター情報の表示) サンプル・プログラムの実行例を確認します。

1. 属性の説明については、[192 ページ](#)の『[キュー統計メッセージ・データ](#)』を参照。

次のコマンドは、キュー・マネージャー saturn.queue.manager からのすべての MQI 統計メッセージを表示します。

```
amqsmon -m saturn.queue.manager -t statistics -a
```

このコマンドの出力は以下のとおりです。

```
RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 73, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqCount: [4, 2102, 0, 0, 0, 46, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [26, 1]
PutFailCount: 0
Put1Count: [40, 0]
Put1FailCount: 0
PutBytes: [57064, 12320]
GetCount: [18, 1]
GetBytes: [52, 12320]
GetFailCount: 2254
BrowseCount: [18, 60]
BrowseBytes: [23784, 30760]
BrowseFailCount: 9
CommitCount: 0
CommitFailCount: 0
BackCount: 0
ExpiredMsgCount: 0
PurgeCount: 0
```

2. 次のコマンドは、キュー・マネージャー saturn.queue.manager 上のキュー LOCALQ に関するすべてのキュー統計メッセージを表示します。

```
amqsmon -m saturn.queue.manager -t statistics -q LOCALQ
```

このコマンドの出力は以下のとおりです。

```
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. 次のコマンドは、キュー・マネージャー `saturn.queue.manager` から 2005 年 4 月 30 日の 15:30 以降に記録されたすべての統計メッセージを表示します。

```
amqsmon -m saturn.queue.manager -t statistics -s "2005-04-30 15.30.00"
```

このコマンドの出力は以下のとおりです。

```
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. 属性の説明については、[169 ページの『キュー・アカウンティング・メッセージ・データ』](#)を参照。

次のコマンドは、キュー・マネージャー `saturn.queue.manager` から 2005 年 4 月 30 日に記録されたすべてのアカウンティング・メッセージを表示します。

```
amqsmon -m saturn.queue.manager -t accounting -s "2005-04-30" -e "2005-04-30"
```

このコマンドの出力は以下のとおりです。

```
RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.29'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.09.30'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'
SeqNumber: 0
ApplicationName: 'amqsput'
ApplicationPid: 8572
ApplicationTid: 1
UserId: 'admin'
ConnDate: '2005-03-16'
```

```

ConnTime: '15.09.29'
DiscDate: '2005-03-16'
DiscTime: '15.09.30'
DiscType: Normal
OpenCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [1, 0]
PutFailCount: 0
PutBytes: [4, 0]
GetCount: [0, 0]
GetFailCount: 0
GetBytes: [0, 0]
BrowseCount: [0, 0]
BrowseFailCount: 0
BrowseBytes: [0, 0]
CommitCount: 0
CommitFailCount: 0
BackCount: 0
InqCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-03-16'
IntervalStartTime: '15.16.22'
IntervalEndDate: '2005-03-16'
IntervalEndTime: '15.16.22'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010c0020'
SeqNumber: 0
ApplicationName: 'runmqsc'
ApplicationPid: 8615
ApplicationTid: 1
...

```

5. 次のコマンドは、アカウントिंग・キューを参照し、MQI アカウントング情報が使用可能なすべてのアプリケーションのアプリケーション名と接続 ID を表示します。

```
amqsmon -m saturn.queue.manager -t accounting -b -l 7006,3024
```

このコマンドの出力は以下のとおりです。

```

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

```


2. ステップ 154 ページの『1』で見つけた ID を検索します。

定数セクションに移動し、MQIAMO * (コマンド形式の整数モニター・パラメーター・タイプ) リストが見つかるまでスクロールダウンします。MQIAMO_Q_MAX_DEPTH を見つけると、値 739 が表示されます。

同様に、MQIAMO64 * (コマンド形式の 64 ビット整数モニター・パラメーター・タイプ) リストを探します。MQIAMO64_AVG_Q_TIME を見つけたら、値 703 を確認します。

アカウントिंग・メッセージ参照と統計メッセージ参照

このページを使用して、アカウントिंग・メッセージと統計メッセージの形式、およびこれらのメッセージに返される情報についての概要を説明します。

アカウントING・メッセージと統計メッセージは、メッセージ記述子とメッセージ・データを含む標準の IBM MQ メッセージです。メッセージ・データには、IBM MQ アプリケーションによって実行された MQI 操作に関する情報、または IBM MQ システムで発生したアクティビティーに関する情報が含まれます。

メッセージ記述子

- MQMD 構造

メッセージ・データ

- PCF ヘッダー (MQCFH)
- 常に返されるアカウントING・メッセージ・データまたは統計メッセージ・データ
- 可能な場合に返されるアカウントING・メッセージ・データまたは統計メッセージ・データ

アカウントING・メッセージと統計メッセージの形式

このページを MQI アカウントING・メッセージの構造の例として使用します。

表 23. MQI アカウンティング・メッセージ構造

MQMD 構造	アカウンティング・メッセージ・ヘッダー MQCFH 構造体	MQI アカウンティング・メッセージ・データ ¹
<p>構造体 ID 構造バージョン レポート・オプション メッセージ・タイプ 満了時刻 フィードバック・コード Encoding コード化文字セット ID メッセージ形式 メッセージ優先順位 Persistence メッセージ ID 相関 ID バックアウト・カウント 応答先キュー 応答先キュー・マネージャー ユーザー ID アカウンティング・トークン アプリケーション識別データ アプリケーション・タイプ アプリケーション名 書き込み日付 書き込み時刻 発生元アプリケーション・データ グループ ID メッセージ順序番号 オフセット メッセージ・フラグ 元の長さ</p>	<p>構造タイプ 構造の長さ 構造バージョン コマンド ID メッセージ順序番号 制御オプション 完了コード 理由コード パラメーター数</p>	<p>キュー・マネージャー 間隔開始日付 間隔開始時刻 間隔終了日付 間隔終了時刻 コマンド・レベル 接続 ID シーケンス番号 アプリケーション名 アプリケーション・プロセス ID アプリケーション・スレッド ID ユーザー ID 接続日付 接続時刻 接続名 チャンネル名 切断日付 切断時刻 切断タイプ オープン・カウント オープン失敗カウント クローズ・カウント クローズ失敗カウント 書き込みカウント 書き込み失敗カウント 書き込み 1 カウント 書き込み 1 失敗カウント 書き込みバイト数 読み取りカウント 読み取り失敗カウント 読み取りバイト数 ブラウズ・カウント ブラウズ失敗カウント ブラウズ・バイト数 コミット・カウント コミット失敗カウント バックアウト・カウント 照会カウント 照会失敗カウント 設定カウント 設定失敗カウント</p>
<p>注： 1. 示されたパラメーターは、MQI アカウンティング・メッセージについて返されるパラメーターです。実際の アカウンティング・メッセージまたは統計メッセージのデータは、メッセージ・カテゴリーによって異なり ます。</p>		

アカウントティング・メッセージと統計メッセージ MQMD (メッセージ記述子)

このページを使用して、アカウントティング・メッセージと統計メッセージのメッセージ記述子と、イベント・メッセージのメッセージ記述子との間の違いを理解します。

アカウントティング・メッセージと統計メッセージのメッセージ記述子に含まれるパラメーターと値は、イベント・メッセージのメッセージ記述子にあるものと同じですが、以下の例外があります。

Format

説明:	メッセージ・データの形式名。
データ型:	MQCHAR8.
値:	MQFMT_ADMIN 管理メッセージ。

アカウントティング・メッセージと統計メッセージのメッセージ記述子に含まれるパラメーターの一部には、そのメッセージを生成したキュー・マネージャーが提供する固定データが設定されています。

MQMD はまた、メッセージを書き込んだキュー・マネージャーの名前 (28 文字に切り捨てられます)、およびメッセージがアカウントティング・キューまたは統計キューに書き込まれた日時を指定します。

アカウントティング・メッセージと統計メッセージのメッセージ・データ

アカウントティング・メッセージと統計メッセージのメッセージ・データは、PCF コマンドの照会と応答に使用される プログラマブル・コマンド・フォーマット (PCF) に基づいています。アカウントティング・メッセージと統計メッセージのメッセージ・データは、PCF ヘッダー (MQCFH)、およびアカウントティング・レポートまたは統計レポートから構成されます。

アカウントティング・メッセージと統計メッセージ MQCFH (PCF ヘッダー)

アカウントティング・メッセージと統計メッセージのメッセージ・ヘッダーは MQCFH 構造です。アカウントティング・メッセージと統計メッセージのメッセージ・ヘッダーに含まれるパラメーターと値は、イベント・メッセージのメッセージ・ヘッダーにあるものと同じですが、以下の例外があります。

Command

説明:	コマンド ID。アカウントティング・メッセージまたは統計メッセージのカテゴリを識別します。
データ型:	MQLONG
値:	MQCMD_ACCOUNTING_MQI MQI アカウントティング・メッセージ。 MQCMD_ACCOUNTING_Q キュー・アカウントティング・メッセージ。 MQCMD_STATISTICS_MQI MQI 統計メッセージ。 MQCMD_STATISTICS_Q キュー統計メッセージ。 MQCMD_STATISTICS_CHANNEL チャンネル統計メッセージ。

Version

説明:	構造体のバージョン番号。
データ型:	MQLONG
値:	MQCFH_VERSION_3 Version-3 (アカウントティング・メッセージおよび統計メッセージの場合)。

アカウントティング・メッセージ・データと統計メッセージ・データ

アカウントティング・メッセージ・データと統計メッセージ・データの内容は、以下のように、アカウントティング・メッセージまたは統計メッセージのカテゴリによって異なります。

MQI アカウントティング・メッセージ

MQI アカウントティング・メッセージ・データは、いくつかの PCF パラメーターから構成されますが、PCF グループは含まれません。

キュー・アカウントティング・メッセージ

キュー・アカウントティング・メッセージ・データは、いくつかの PCF パラメーターと、1 から 100 個の範囲の *QAccountingData* PCF グループから構成されます。

MQI 統計メッセージ

MQI 統計メッセージ・データは、いくつかの PCF パラメーターから構成されますが、PCF グループは含まれません。

キュー統計メッセージ

キュー統計メッセージ・データは、いくつかの PCF パラメーターと、1 から 100 個の範囲の *QStatisticsData* PCF グループから構成されます。

チャンネル統計メッセージ

チャンネル統計メッセージ・データは、いくつかの PCF パラメーターと、1 から 100 個の範囲の *ChIStatisticsData* PCF グループから構成されます。

MQI アカウントティング・メッセージ・データ

このページを使用して、MQI アカウントティング・メッセージの構造を確認します。

メッセージ名:	MQI アカウントティング・メッセージ。
プラットフォーム:	すべて。ただし IBM MQ for z/OS を除く。
システム・キュー:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

説明:	キュー・マネージャーの名前
ID:	MQCA_Q_MGR_NAME
データ型:	MQCFST
最大長:	MQ_Q_MGR_NAME_LENGTH
戻り:	常時

IntervalStartDate

説明:	モニター期間の開始日。
ID:	MQCAMO_START_DATE
データ型:	MQCFST
最大長:	MQ_DATE_LENGTH
戻り:	常時

IntervalStartTime

説明:	モニター期間の開始時刻。
ID:	MQCAMO_START_TIME
データ型:	MQCFST
最大長:	MQ_TIME_LENGTH
戻り:	常時

IntervalEndDate

説明: モニター期間の終了日。
ID: MQCAMO_END_DATE
データ型: MQCFST
最大長: MQ_DATE_LENGTH
戻り: 常時

IntervalEndTime

説明: モニター期間の終了時刻。
ID: MQCAMO_END_TIME
データ型: MQCFST
最大長: MQ_TIME_LENGTH
戻り: 常時

CommandLevel

説明: キュー・マネージャーのコマンド・レベル。
ID: MQIA_COMMAND_LEVEL
データ型: MQCFIN
戻り: 常時

ConnectionId

説明: IBM MQ 接続の接続 ID
ID: MQBACF_CONNECTION_ID
データ型: MQCFBS
最大長: MQ_CONNECTION_ID_LENGTH
戻り: 常時

SeqNumber

説明: 順序番号。この値は、長期間実行される接続の後続のレコードごとに増加します。
ID: MQIACF_SEQUENCE_NUMBER
データ型: MQCFIN
戻り: 常時

ApplicationName

説明: アプリケーションの名前。このフィールドの内容は、メッセージ記述子の *PutApplName* フィールドの内容と同じです。
ID: MQCACF_APPL_NAME
データ型: MQCFST
最大長: MQ_APPL_NAME_LENGTH
戻り: 常時

ApplicationPid

説明: オペレーティング・システムのアプリケーション・プロセス ID。
ID: MQIACF_PROCESS_ID
データ型: MQCFIN
戻り: 常時

ApplicationTid

説明: アプリケーションにおける接続の IBM MQ スレッド ID
ID: MQIACF_THREAD_ID
データ型: MQCFIN
戻り: 常時

UserId

説明: アプリケーションのユーザー ID コンテキスト。
ID: MQCACF_USER_IDENTIFIER
データ型: MQCFST
最大長: MQ_USER_ID_LENGTH
戻り: 常時

ConnDate

説明: MQCONN 操作の日付。
ID: MQCAMO_CONN_DATE
データ型: MQCFST
最大長: MQ_TIME_LENGTH
戻り: 可能な場合。

ConnTime

説明: MQCONN 操作の時刻。
ID: MQCAMO_CONN_TIME
データ型: MQCFST
最大長: MQ_TIME_LENGTH
戻り: 可能な場合。

ConnName

説明: クライアント接続の接続名。
ID: MQCACH_CONNECTION_NAME
データ型: MQCFST
最大長: MQ_CONN_NAME_LENGTH
戻り: 可能な場合。

ChannelName

説明: クライアント接続のチャンネル名
ID: MQCACH_CHANNEL_NAME
データ型: MQCFST
最大長: MQ_CHANNEL_NAME_LENGTH
戻り: 可能な場合。

RemoteProduct

説明: DISPLAY CHSATUS の **RPRODUCT** フィールドに表示される、クライアント接続のリモート製品 ID
ID: MQCACH_REMOTE_PRODUCT
データ型: MQCFST
最大長: MQ_REMOTE_PRODUCT_LENGTH
戻り: 可能な場合。

RemoteVersion

説明: DISPLAY CHSTATUS の **RVERSION** フィールドに表示される、クライアント接続のリモート製品バージョン
ID: MQCACH_REMOTE_VERSION
データ型: MQCFST
最大長: MQ_REMOTE_VERSION_LENGTH
戻り: 可能な場合。

DiscDate

説明: MQDISC 操作の日付。
ID: MQCAMO_DISC_DATE
データ型: MQCFST
最大長: MQ_DATE_LENGTH
戻り: 可能な場合。

DiscTime

説明: MQDISC 操作の時刻。
ID: MQCAMO_DISC_TIME
データ型: MQCFST
最大長: MQ_TIME_LENGTH
戻り: 可能な場合。

DiscType

説明: 切断のタイプ。
ID: MQIAMO_DISC_TYPE
データ型: MQCFIN

値: 指定できる値は以下のとおりです。
MQDISCONNECT_NORMAL
アプリケーションにより要求
MQDISCONNECT_IMPLICIT
アプリケーションの異常終了
MQDISCONNECT_Q_MGR
キュー・マネージャーによって切断された接続

戻り: 可能な場合。

OpenCount

説明: MQOPEN への呼び出しを直接発行する方法、または MQPUT1 verb を使用する
方法のいずれかによって、正常にオープンされたオブジェクトの数。このパラメ
ーターはオブジェクト・タイプで索引付けされた整数リストです。 [リファレンス・
ノート 1](#) を参照してください。

ID: MQIAMO_OPENS

データ型: MQCFIL

戻り: 可能な場合。

OpenFailCount

説明: オブジェクトをオープンしようとして失敗した回数。このパラメーターはオブ
ジェクト・タイプで索引付けされた整数リストです。 [リファレンス・ノート 1](#) を
参照してください。

ID: MQIAMO_OPENS_FAILED

データ型: MQCFIL

戻り: 可能な場合。

CloseCount

説明: クローズされたオブジェクトの数。このパラメーターはオブジェクト・タイプで
索引付けされた整数リストです。 [リファレンス・ノート 1](#) を参照してください。

ID: MQIAMO_CLOSES

データ型: MQCFIL

戻り: 可能な場合。

CloseFailCount

説明: オブジェクトをクローズしようとして失敗した回数。このパラメーターはオブ
ジェクト・タイプで索引付けされた整数リストです。 [リファレンス・ノート 1](#) を
参照してください。

ID: MQIAMO_CLOSES_FAILED

データ型: MQCFIL

戻り: 可能な場合。

PutCount

説明: 正常にキューに書き込まれた持続メッセージと非持続メッセージの数。
MQPUT1 呼び出しを使用して書き込まれたメッセージを除きます。このパラメ
ーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を
参照してください。

ID: MQIAMO_PUTS
データ型: MQCFIL
戻り: 可能な場合。

PutFailCount

説明: メッセージを書き込もうとして失敗した回数。

ID: MQIAMO_PUTS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

Put1Count

説明: MQPUT1 呼び出しを使用して正常にキューに書き込まれた持続メッセージと非持続メッセージの数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO_PUT1S
データ型: MQCFIL
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

Put1FailCount

説明: MQPUT1 呼び出しを使用してメッセージを書き込もうとして失敗した回数。

ID: MQIAMO_PUT1S_FAILED
データ型: MQCFIN
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

PutBytes

説明: 持続メッセージと非持続メッセージで、書き込み呼び出しを使用して書き込まれたバイト数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO64_PUT_BYTES
データ型: MQCFIL64
戻り: 可能な場合。

GetCount

説明: 持続メッセージと非持続メッセージで正常に行われた破壊 MQGET 呼び出しの回数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO_GETS
データ型: MQCFIL
戻り: 可能な場合。

GetFailCount

説明: 失敗した破壊 MQGET 呼び出しの回数。

ID: MQIAMO_GETS_FAILED

データ型: MQCFIN

戻り: 可能な場合。

GetBytes

説明: 持続メッセージと非持続メッセージで取り出された総バイト数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_GET_BYTES

データ型: MQCFIL64

戻り: 可能な場合。

BrowseCount

説明: 持続メッセージと非持続メッセージで正常に行われた非破壊 MQGET 呼び出しの回数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_BROWSES

データ型: MQCFIL

戻り: 可能な場合。

BrowseFailCount

説明: 失敗した非破壊 MQGET 呼び出しの回数。

ID: MQIAMO_BROWSES_FAILED

データ型: MQCFIN

戻り: 可能な場合。

BrowseBytes

説明: 持続メッセージと非持続メッセージで参照された総バイト数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_BROWSE_BYTES

データ型: MQCFIL64

戻り: 可能な場合。

CommitCount

説明: 正常に実行されたトランザクションの数。この数には、接続されたアプリケーションによって暗黙的にコミットされたトランザクションが含まれます。未処理の作業がないコミット要求は、この数に含まれます。

ID: MQIAMO_COMMITS

データ型: MQCFIN

戻り: 可能な場合。

CommitFailCount

説明: トランザクションを完了しようとして失敗した回数。
ID: MQIAMO_COMMITS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

BackCount

説明: 処理されたバックアウトの数。異常切断による暗黙バックアウトも含まれます。
ID: MQIAMO_BACKOUTS
データ型: MQCFIN
戻り: 可能な場合。

InqCount

説明: 正常に行われたオブジェクト照会の数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。[リファレンス・ノート 1](#)を参照してください。
ID: MQIAMO_INQS
データ型: MQCFIL
戻り: 可能な場合。

InqFailCount

説明: 失敗したオブジェクト照会の数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。[リファレンス・ノート 1](#)を参照してください。
ID: MQIAMO_INQS_FAILED
データ型: MQCFIL
戻り: 可能な場合。

SetCount

説明: 正常に行われた MQSET 呼び出しの回数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。[リファレンス・ノート 1](#)を参照してください。
ID: MQIAMO_SETS
データ型: MQCFIL
戻り: 可能な場合。

SetFailCount

説明: 失敗した MQSET 呼び出しの回数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。[リファレンス・ノート 1](#)を参照してください。
ID: MQIAMO_SETS_FAILED
データ型: MQCFIL
戻り: 可能な場合。

SubCountDur

説明: 正常実行され、永続サブスクリプションを作成、変更、または再開したサブスクライブ要求の数。これは、操作のタイプで索引付けされた値の配列です。

0 = 作成されたサブスクリプションの数
1 = 変更されたサブスクリプションの数
2 = 再開されたサブスクリプションの数

ID: MQIAMO_SUBS_DUR
データ型: MQCFIL
戻り: 可能な場合。

SubCountNDur

説明: 正常実行され、非永続サブスクリプションを作成、変更、または再開したサブスクライブ要求の数。これは、操作のタイプで索引付けされた値の配列です。

0 = 作成されたサブスクリプションの数
1 = 変更されたサブスクリプションの数
2 = 再開されたサブスクリプションの数

ID: MQIAMO_SUBS_NDUR
データ型: MQCFIL
戻り: 可能な場合。

SubFailCount

説明: 失敗したサブスクライブ要求の数。

ID: MQIAMO_SUBS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

UnsubCountDur

説明: 正常に行われた、永続サブスクリプションに対するアンサブスクライブ要求の数。これは、操作のタイプで索引付けされた値の配列です。

0 - サブスクリプションは閉じられましたが、除去はされていません。
1 - サブスクリプションが閉じられ、除去されました。

ID: MQIAMO_UNSUBS_DUR
データ型: MQCFIL
戻り: 可能な場合。

UnsubCountNDur

説明: 正常に行われた、永続サブスクリプションに対するアンサブスクライブ要求の数。これは、操作のタイプで索引付けされた値の配列です。

0 - サブスクリプションは閉じられましたが、除去はされていません。
1 - サブスクリプションが閉じられ、除去されました。

ID: MQIAMO_UNSUBS_NDUR

データ型: MQCFIL
戻り: 可能な場合。

UnsubFailCount

説明: 失敗したアンサブスクライブ要求の数。
ID: MQIAMO_UNSUBS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

SubRqCount

説明: 正常に行われた MQSUBRQ 要求の数。
ID: MQIAMO_SUBRQS
データ型: MQCFIN
戻り: 可能な場合。

SubRqFailCount

説明: 失敗した MQSUB 要求の数。
ID: MQIAMO_SUBRQS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

CBCount

説明: 正常に行われた MQCB 要求の数。これは、操作のタイプで索引付けされた値の配列です。
0 - コールバックが作成または変更されました。
1 - コールバックが除去されました。
2 - コールバックが再開されました。
3 - コールバックが中断されました。
ID: MQIAMO_CBS
データ型: MQCFIN
戻り: 可能な場合。

CBFailCount

説明: 失敗した MQCB 要求の数。
ID: MQIAMO_CBS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

CtlCount

説明: 正常に行われた MQCTL 要求の数。これは、操作のタイプで索引付けされた値の配列です。

- 0 - 接続が開始されました。
- 1 - 接続が停止されました。
- 2 - 接続が再開されました。
- 3 - 接続が中断されました。

ID: MQIAMO_CTL5

データ型: MQCFIL

戻り: 可能な場合。

CtlFailCount

説明: 失敗した MQCTL 要求の数。

ID: MQIAMO_CTL5_FAILED

データ型: MQCFIN

戻り: 可能な場合。

StatCount

説明: 正常に行われた MQSTAT 要求の数。

ID: MQIAMO_STATS

データ型: MQCFIN

戻り: 可能な場合。

StatFailCount

説明: 失敗した MQSTAT 要求の数。

ID: MQIAMO_STATS_FAILED

データ型: MQCFIN

戻り: 可能な場合。

PutTopicCount

説明: 正常にトピックに書き込まれた持続メッセージと非持続メッセージの数。MQPUT1 呼び出しを使用して書き込まれたメッセージを除きます。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

注: 解決するとトピックになるキュー別名を使用して書き込まれたメッセージは、この値に含まれます。

ID: MQIAMO_TOPIC_PUTS

データ型: MQCFIL

戻り: 可能な場合。

PutTopicFailCount

説明: メッセージをトピックに書き込もうとして失敗した回数。

ID: MQIAMO_TOPIC_PUTS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

Put1TopicCount

説明: MQPUT1 呼び出しを使用して正常にトピックに書き込まれた持続メッセージと非持続メッセージの数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

注: 解決するとトピックになるキュー別名を使用して書き込まれたメッセージは、この値に含まれます。

ID: MQIAMO_TOPIC_PUT1S
データ型: MQCFIL
戻り: 可能な場合。

Put1TopicFailCount

説明: MQPUT1 呼び出しを使用してメッセージをトピックに書き込もうとして失敗した回数。

ID: MQIAMO_TOPIC_PUT1S_FAILED
データ型: MQCFIN
戻り: 可能な場合。

PutTopicBytes

説明: 解決するとパブリッシュ操作になる持続メッセージと非持続メッセージに対する書き込み呼び出しを使用して書き込まれたバイト数。これはアプリケーションによって書き込まれたバイト数であり、サブスクライバーに送信された最終結果バイト数ではありません。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_TOPIC_PUT_BYTES
データ型: MQCFIL64
戻り: 可能な場合。

キュー・アカウントिंग・メッセージ・データ

このページを使用して、キュー・アカウントिंग・メッセージの構造を確認します。

メッセージ名:	キュー・アカウントिंग・メッセージ。
プラットフォーム:	すべて。ただし IBM MQ for z/OS を除く。
システム・キュー:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

QueueManager

説明: キュー・マネージャーの名前
ID: MQCA_Q_MGR_NAME
データ型: MQCFST
最大長: MQ_Q_MGR_NAME_LENGTH
戻り: 常時

IntervalStartDate

説明: モニター期間の開始日。
ID: MQCAMO_START_DATE
データ型: MQCFST
最大長: MQ_DATE_LENGTH
戻り: 常時

IntervalStartTime

説明: モニター期間の開始時刻。
ID: MQCAMO_START_TIME
データ型: MQCFST
最大長: MQ_TIME_LENGTH
戻り: 常時

IntervalEndDate

説明: モニター期間の終了日。
ID: MQCAMO_END_DATE
データ型: MQCFST
最大長: MQ_DATE_LENGTH
戻り: 常時

IntervalEndTime

説明: モニター期間の終了時刻。
ID: MQCAMO_END_TIME
データ型: MQCFST
最大長: MQ_TIME_LENGTH
戻り: 常時

CommandLevel

説明: キュー・マネージャーのコマンド・レベル。
ID: MQIA_COMMAND_LEVEL
データ型: MQCFIN
戻り: 常時

ConnectionId

説明: IBM MQ 接続の接続 ID
ID: MQBACF_CONNECTION_ID
データ型: MQCFBS
最大長: MQ_CONNECTION_ID_LENGTH
戻り: 常時

SeqNumber

説明:	順序番号。この値は、長期間実行される接続の後続のレコードごとに増加します。
ID:	MQIACF_SEQUENCE_NUMBER
データ型:	MQCFIN
戻り:	常時

ApplicationName

説明:	アプリケーションの名前。このフィールドの内容は、メッセージ記述子の PutApplName フィールドの内容と同じです。
ID:	MQCACF_APPL_NAME
データ型:	MQCFST
最大長:	MQ_APPL_NAME_LENGTH
戻り:	常時

ApplicationPid

説明:	オペレーティング・システムのアプリケーション・プロセス ID。
ID:	MQIACF_PROCESS_ID
データ型:	MQCFIN
戻り:	常時

ApplicationTid

説明:	アプリケーションにおける接続の IBM MQ スレッド ID
ID:	MQIACF_THREAD_ID
データ型:	MQCFIN
戻り:	常時

UserId

説明:	アプリケーションのユーザー ID コンテキスト。
ID:	MQCACF_USER_IDENTIFIER
データ型:	MQCFST
最大長:	MQ_USER_ID_LENGTH
戻り:	常時

ChannelName

説明:	クライアント接続のチャンネル名
ID:	MQCACH_CHANNEL_NAME
データ型:	MQCFST
最大長:	MQ_CHANNEL_NAME_LENGTH
戻り:	可能な場合。

ConnName

説明: クライアント接続の接続名。
ID: MQCACH_CONNECTION_NAME
データ型: MQCFST
最大長: MQ_CONN_NAME_LENGTH
戻り: 可能な場合。

ObjectCount

説明: アカウンティング・データが記録された、間隔中にアクセスされたキューの数。
この値は、メッセージに含まれる *QAccountingData* PCF グループの数に設定されます。
ID: MQIAMO_OBJECT_COUNT
データ型: MQCFIN
戻り: 常時

QAccountingData

説明: キューのアカウンティング詳細を指定するグループ化されたパラメーター。
ID: MQGACF_Q_ACCOUNTING_DATA
データ型: MQCFGR

グループ内のパラ メーター:	<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>
-------------------	---

戻り:	常時
-----	----

QName

説明:	キューの名前。
ID:	MQCA_Q_NAME
データ型:	MQCFST
組み込まれる PCF グループ:	<i>QAccountingData</i>
最大長:	MQ_Q_NAME_LENGTH
戻り:	可能な場合。

CreateDate

説明:	キューが作成された日付。
ID:	MQCA_CREATION_DATE
データ型:	MQCFST
組み込まれる PCF グループ:	<i>QAccountingData</i>

最大長: MQ_DATE_LENGTH
戻り: 可能な場合。

CreateTime

説明: キューが作成された時刻。
ID: MQCA_CREATION_TIME
データ型: MQCFST
組み込まれる PCF グループ: QAccountingData
最大長: MQ_TIME_LENGTH
戻り: 可能な場合。

QType

説明: キューのタイプ。
ID: MQIA_Q_TYPE
データ型: MQCFIN
組み込まれる PCF グループ: QAccountingData
値: MQQT_LOCAL
戻り: 可能な場合。

QDefinitionType

説明: キュー定義タイプ。
ID: MQIA_DEFINITION_TYPE
データ型: MQCFIN
組み込まれる PCF グループ: QAccountingData
値: 指定可能な値は以下のとおりです。
MQQDT_PREDEFINED
MQQDT_PERMANENT_DYNAMIC
MQQDT_TEMPORARY_DYNAMIC
戻り: 可能な場合。

OpenCount

説明: MQOPEN への呼び出しを直接発行する方法、または MQPUT1 verb を使用する方
法のいずれかによって、このキューがこの間隔内でアプリケーションによってオ
ープンされた回数。
ID: MQIAMO_OPENS
データ型: MQCFIL
組み込まれる PCF グループ: QAccountingData
戻り: 可能な場合。

OpenDate

説明: キューがこの記録間隔内で最初にオープンされた日付。この間隔の開始時にキューが既にオープンしていた場合、この値はキューが最初にオープンされた日付を反映します。

ID: MQCAMO_OPEN_DATE

データ型: MQCFST

組み込まれる PCF グループ: *QAccountingData*

戻り: 可能な場合。

OpenTime

説明: キューがこの記録間隔内で最初にオープンされた時刻。この間隔の開始時にキューが既にオープンしていた場合、この値はキューが最初にオープンされた時刻を反映します。

ID: MQCAMO_OPEN_TIME

データ型: MQCFST

組み込まれる PCF グループ: *QAccountingData*

戻り: 可能な場合。

CloseDate

説明: この記録間隔内でキューが最後にクローズされた日付。キューがまだオープンしている場合、この値は返されません。

ID: MQCAMO_CLOSE_DATE

データ型: MQCFST

組み込まれる PCF グループ: *QAccountingData*

戻り: 可能な場合。

CloseTime

説明: この記録間隔内でキューが最後にクローズされた時刻。キューがまだオープンしている場合、この値は返されません。

ID: MQCAMO_CLOSE_TIME

データ型: MQCFST

組み込まれる PCF グループ: *QAccountingData*

戻り: 可能な場合。

PutCount

説明: MQPUT1 呼び出しを除く、正常にキューに書き込まれた持続メッセージと非持続メッセージの数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_PUTS

データ型: MQCFIL

組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

PutFailCount

説明: MQPUT1 呼び出しを除く、メッセージを書き込もうとして失敗した回数。
ID: MQIAMO_PUTS_FAILED
データ型: MQCFIN
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

Put1Count

説明: MQPUT1 呼び出しを使用して正常にキューに書き込まれた持続メッセージと非持続メッセージの数。このパラメータは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。
ID: MQIAMO_PUT1S
データ型: MQCFIL
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

Put1FailCount

説明: MQPUT1 呼び出しを使用してメッセージを書き込もうとして失敗した回数。
ID: MQIAMO_PUT1S_FAILED
データ型: MQCFIN
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

PutBytes

説明: 持続メッセージと非持続メッセージで書き込まれた総バイト数。このパラメータは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。
ID: MQIAMO64_PUT_BYTES
データ型: MQCFIL64
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

PutMinBytes

説明: キューに入れられた持続メッセージと非持続メッセージの最小サイズ。このパラメータは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_PUT_MIN_BYTES
データ型: MQCFIL
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

PutMaxBytes

説明: キューに入れられた持続メッセージと非持続メッセージの最大サイズ。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_PUT_MAX_BYTES
データ型: MQCFIL
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

GeneratedMsgCount

説明: 生成されたメッセージの数。以下のメッセージが生成されます。

- キュー・サイズ上限イベント
- キュー・サイズ下限イベント

ID: MQIAMO_GENERATED_MSGS
データ型: MQCFIN
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

GetCount

説明: 持続メッセージと非持続メッセージで正常に行われた破壊 MQGET 呼び出しの回数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_GETS
データ型: MQCFIL
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

GetFailCount

説明: 失敗した破壊 MQGET 呼び出しの回数。

ID: MQIAMO_GETS_FAILED
データ型: MQCFIN
組み込まれる PCF グループ: *QAccountingData*
戻り: 可能な場合。

GetBytes

説明: 持続メッセージと非持続メッセージの破壊 MQGET 呼び出しで読み取られたバイト数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_GET_BYTES

データ型: MQCFIL64

組み込まれる PCF グループ: QAccountingData

戻り: 可能な場合。

GetMinBytes

説明: キューから取り出された持続メッセージと非持続メッセージの最小サイズ。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_GET_MIN_BYTES

データ型: MQCFIL

組み込まれる PCF グループ: QAccountingData

戻り: 可能な場合。

GetMaxBytes

説明: キューから取り出された持続メッセージと非持続メッセージの最大サイズ。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_GET_MAX_BYTES

データ型: MQCFIL

組み込まれる PCF グループ: QAccountingData

戻り: 可能な場合。

BrowseCount

説明: 持続メッセージと非持続メッセージで正常に行われた非破壊 MQGET 呼び出しの回数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_BROWSES

データ型: MQCFIL

組み込まれる PCF グループ: QAccountingData

戻り: 可能な場合。

BrowseFailCount

説明: 失敗した非破壊 MQGET 呼び出しの回数。

ID: MQIAMO_BROWSES_FAILED

データ型: MQCFIN

組み込まれる PCF グループ: *QAccountingData*

戻り: 可能な場合。

BrowseBytes

説明: 持続メッセージを返した非破壊 MQGET 呼び出しで読み取られたバイト数。

ID: MQIAMO64_BROWSE_BYTES

データ型: MQCFIL64

組み込まれる PCF グループ: *QAccountingData*

戻り: 可能な場合。

BrowseMinBytes

説明: キューから参照された持続メッセージと非持続メッセージの最小サイズ。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_BROWSE_MIN_BYTES

データ型: MQCFIL

組み込まれる PCF グループ: *QAccountingData*

戻り: 可能な場合。

BrowseMaxBytes

説明: キューから参照された持続メッセージと非持続メッセージの最大サイズ。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_BROWSE_MAX_BYTES

データ型: MQCFIL

組み込まれる PCF グループ: *QAccountingData*

戻り: 可能な場合。

TimeOnQMin

説明: 破壊的に取り出されるまでに、持続メッセージと非持続メッセージがキューに残る最小時間 (マイクロ秒)。同期点で取り出されたメッセージの場合、この値には取得操作がコミットされる前の時間は含まれていません。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_Q_TIME_MIN

データ型: MQCFIL64

組み込まれる PCF グループ: *QAccountingData*

戻り: 可能な場合。

TimeOnQAvg

説明:	破壊的に取り出されるまでに、持続メッセージと非持続メッセージがキューに残る平均時間 (マイクロ秒)。同期点で取り出されたメッセージの場合、この値には取得操作がコミットされる前の時間は含まれていません。このパラメーターは持続性値で索引付けされた整数リストです。 リファレンス・ノート 2 を参照してください。
ID:	MQIAMO64_Q_TIME_AVG
データ型:	MQCFIL64
組み込まれる PCF グループ:	QAccountingData
戻り:	可能な場合。

TimeOnQMax

説明:	破壊的に取り出されるまでに、持続メッセージと非持続メッセージがキューに残る最大時間 (マイクロ秒)。同期点で取り出されたメッセージの場合、この値には取得操作がコミットされる前の時間は含まれていません。このパラメーターは持続性値で索引付けされた整数リストです。 リファレンス・ノート 2 を参照してください。
ID:	MQIAMO64_Q_TIME_MAX
データ型:	MQCFIL64
組み込まれる PCF グループ:	QAccountingData
戻り:	可能な場合。

CBCCount

説明:	正常に行われた MQCB 要求の数。これは、操作のタイプで索引付けされた値の配列です。 0 - コールバックが作成または変更されました。 1 - コールバックが除去されました。 2 - コールバックが再開されました。 3 - コールバックが中断されました。
ID:	MQIAMO_CBS
データ型:	MQCFIN
戻り:	可能な場合。

CBFailCount

説明:	失敗した MQCB 要求の数。
ID:	MQIAMO_CBS_FAILED
データ型:	MQCFIN
戻り:	可能な場合。

MQI 統計メッセージ・データ

このページを使用して、MQI 統計メッセージの構造を確認します。

メッセージ名: MQI 統計メッセージ。

プラットフォーム:	すべて。ただし IBM MQ for z/OS を除く。
システム・キュー:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

説明:	キュー・マネージャーの名前。
ID:	MQCA_Q_MGR_NAME
データ型:	MQCFST
最大長:	MQ_Q_MGR_NAME_LENGTH
戻り:	常時。

IntervalStartDate

説明:	モニター期間の開始日。
ID:	MQCAMO_START_DATE
データ型:	MQCFST
最大長:	MQ_DATE_LENGTH
戻り:	常時。

IntervalStartTime

説明:	モニター期間の開始時刻。
ID:	MQCAMO_START_TIME
データ型:	MQCFST
最大長:	MQ_TIME_LENGTH
戻り:	常時。

IntervalEndDate

説明:	モニター期間の終了日。
ID:	MQCAMO_END_DATE
データ型:	MQCFST
最大長:	MQ_DATE_LENGTH
戻り:	常時。

IntervalEndTime

説明:	モニター期間の終了時刻。
ID:	MQCAMO_END_TIME
データ型:	MQCFST
最大長:	MQ_TIME_LENGTH
戻り:	常時。

CommandLevel

説明:	キュー・マネージャーのコマンド・レベル。
ID:	MQIA_COMMAND_LEVEL

データ型: MQCFIN
戻り: 常時。

ConnCount

説明: 正常に行われたキュー・マネージャーへの接続の回数。
ID: MQIAMO_CONNS
データ型: MQCFIN
戻り: 可能な場合。

ConnFailCount

説明: 失敗した接続の試行回数。
ID: MQIAMO_CONNS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

ConnsMax

説明: 記録間隔内の同時接続の最大数。
ID: MQIAMO_CONNS_MAX
データ型: MQCFIN
戻り: 可能な場合。

DiscCount

説明: キュー・マネージャーからの切断の回数。これは以下の定数によって索引付けされた整数配列です。

- MQDISCONNECT_NORMAL
- MQDISCONNECT_IMPLICIT
- MQDISCONNECT_Q_MGR

ID: MQIAMO_DISCS
データ型: MQCFIL
戻り: 可能な場合。

OpenCount

説明: MQOPEN への呼び出しを直接発行する方法、または MQPUT1 verb を使用する方法のいずれかによって、正常にオープンされたオブジェクトの数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。[リファレンス・ノート 1](#) を参照してください。
ID: MQIAMO_OPENS
データ型: MQCFIL
戻り: 可能な場合。

OpenFailCount

説明: 失敗したオブジェクトのオープンの試行回数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。 [リファレンス・ノート 1](#) を参照してください。

ID: MQIAMO_OPENS_FAILED

データ型: MQCFIL

戻り: 可能な場合。

CloseCount

説明: 正常にクローズされたオブジェクトの数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。 [リファレンス・ノート 1](#) を参照してください。

ID: MQIAMO_CLOSES

データ型: MQCFIL

戻り: 可能な場合。

CloseFailCount

説明: 失敗したオブジェクトのクローズの試行回数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。 [リファレンス・ノート 1](#) を参照してください。

ID: MQIAMO_CLOSES_FAILED

データ型: MQCFIL

戻り: 可能な場合。

InqCount

説明: 正常に照会されたオブジェクトの数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。 [リファレンス・ノート 1](#) を参照してください。

ID: MQIAMO_INQS

データ型: MQCFIL

戻り: 可能な場合。

InqFailCount

説明: 失敗したオブジェクト照会の数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。 [リファレンス・ノート 1](#) を参照してください。

ID: MQIAMO_INQS_FAILED

データ型: MQCFIL

戻り: 可能な場合。

SetCount

説明: 正常に更新 (SET) されたオブジェクトの数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。 [リファレンス・ノート 1](#) を参照してください。

ID: MQIAMO_SETS

データ型: MQCFIL
戻り: 可能な場合。

SetFailCount

説明: 失敗した SET の試行回数。このパラメーターはオブジェクト・タイプで索引付けされた整数リストです。 [リファレンス・ノート 1](#) を参照してください。

ID: MQIAMO_SETS_FAILED

データ型: MQCFIL

戻り: 可能な場合。

PutCount

説明: MQPUT1 要求を除く、正常にキューに書き込まれた持続メッセージと非持続メッセージの数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_PUTS

データ型: MQCFIL

戻り: 可能な場合。

PutFailCount

説明: 失敗したメッセージ書き込みの試行回数。

ID: MQIAMO_PUTS_FAILED

データ型: MQCFIN

戻り: 可能な場合。

Put1Count

説明: MQPUT1 要求を使用して正常にキューに書き込まれた持続メッセージと非持続メッセージの数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_PUT1S.

データ型: MQCFIL

戻り: 可能な場合。

Put1FailCount

説明: MQPUT1 要求を使用して持続メッセージと非持続メッセージをキューに書き込もうとして失敗した回数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_PUT1S_FAILED.

データ型: MQCFIL

戻り: 可能な場合。

PutBytes

説明: 書き込み要求を使用して書き込まれた持続メッセージと非持続メッセージのバイト数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_PUT_BYTES.
データ型: MQCFIL64.
戻り: 可能な場合。

GetCount

説明: 持続メッセージと非持続メッセージで正常に行われた破壊読み取り要求の回数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO_GETS
データ型: MQCFIL
戻り: 可能な場合。

GetFailCount

説明: 失敗した破壊取得要求の数。
ID: MQIAMO_GETS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

GetBytes

説明: 持続メッセージと非持続メッセージの破壊取得要求で読み取られたバイト数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO64_GET_BYTES.
データ型: MQCFIL64.
戻り: 可能な場合。

BrowseCount

説明: 持続メッセージと非持続メッセージで正常に行われた非破壊読み取り要求の回数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO_BROWSES
データ型: MQCFIL
戻り: 可能な場合。

BrowseFailCount

説明: 失敗した非破壊取得要求の数。
ID: MQIAMO_BROWSES_FAILED
データ型: MQCFIN
戻り: 可能な場合。

BrowseBytes

説明: 持続メッセージと非持続メッセージの非破壊取得要求で読み取られたバイト数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO64_BROWSE_BYTES.
データ型: MQCFIL64.
戻り: 可能な場合。

CommitCount

説明: 正常に完了したトランザクションの数。この数には、アプリケーションの切断によって暗黙にコミットされたトランザクション、および未処理の作業がないコミット要求が含まれます。

ID: MQIAMO_COMMITS
データ型: MQCFIN
戻り: 可能な場合。

CommitFailCount

説明: トランザクションを完了しようとして失敗した回数。

ID: MQIAMO_COMMITS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

BackCount

説明: 処理されたバックアウトの数。異常切断による暗黙のバックアウトも含まれません。

ID: MQIAMO_BACKOUTS
データ型: MQCFIN
戻り: 可能な場合。

ExpiredMsgCount

説明: 有効期限が切れたために、取り出す前に廃棄された持続メッセージと非持続メッセージの数。

ID: MQIAMO_MSGS_EXPIRED
データ型: MQCFIN
戻り: 可能な場合。

PurgeCount

説明: キューがクリアされた回数。

ID: MQIAMO_MSGS_PURGED
データ型: MQCFIN
戻り: 可能な場合。

SubCountDur

説明: 正常実行され、永続サブスクリプションを作成、変更、または再開したサブスクライブ要求の数。これは、操作のタイプで索引付けされた値の配列です。

0 = 作成されたサブスクリプションの数
1 = 変更されたサブスクリプションの数
2 = 再開されたサブスクリプションの数

ID: MQIAMO_SUBS_DUR
データ型: MQCFIL
戻り: 可能な場合。

SubCountNDur

説明: 正常実行され、非永続サブスクリプションを作成、変更、または再開したサブスクライブ要求の数。これは、操作のタイプで索引付けされた値の配列です。

0 = 作成されたサブスクリプションの数
1 = 変更されたサブスクリプションの数
2 = 再開されたサブスクリプションの数

ID: MQIAMO_SUBS_NDUR
データ型: MQCFIL
戻り: 可能な場合。

SubFailCount

説明: 失敗したサブスクライブ要求の数。

ID: MQIAMO_SUBS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

UnsubCountDur

説明: 正常に行われた、永続サブスクリプションに対するアンサブスクライブ要求の数。これは、操作のタイプで索引付けされた値の配列です。

0 - サブスクリプションは閉じられましたが、除去はされていません。
1 - サブスクリプションが閉じられ、除去されました。

ID: MQIAMO_UNSUBS_DUR
データ型: MQCFIL
戻り: 可能な場合。

UnsubCountNDur

説明: 正常に行われた、非永続サブスクリプションに対するアンサブスクライブ要求の数。これは、操作のタイプで索引付けされた値の配列です。

0 - サブスクリプションは閉じられましたが、除去はされていません。
1 - サブスクリプションが閉じられ、除去されました。

ID: MQIAMO_UNSUBS_NDUR

データ型: MQCFIL
戻り: 可能な場合。

UnsubFailCount

説明: 失敗したアンサブスクライブ要求の数。
ID: MQIAMO_UNSUBS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

SubRqCount

説明: 正常に行われた MQSUBRQ 要求の数。
ID: MQIAMO_SUBRQS
データ型: MQCFIN
戻り: 可能な場合。

SubRqFailCount

説明: 失敗した MQSUBRQ 要求の数。
ID: MQIAMO_SUBRQS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

CBCount

説明: 正常に行われた MQCB 要求の数。これは、操作のタイプで索引付けされた値の配列です。
0 - コールバックが作成または変更されました。
1 - コールバックが除去されました。
2 - コールバックが再開されました。
3 - コールバックが中断されました。
ID: MQIAMO_CBS
データ型: MQCFIL
戻り: 可能な場合。

CBFailCount

説明: 失敗した MQCB 要求の数。
ID: MQIAMO_CBS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

CtlCount

説明: 正常に行われた MQCTL 要求の数。これは、操作のタイプを示す添字が付いた値の配列です。

- 0 - 接続が開始されました。
- 1 - 接続が停止されました。
- 2 - 接続が再開されました。
- 3 - 接続が中断されました。

ID: MQIAMO_CTL5

データ型: MQCFIL

戻り: 可能な場合。

CtlFailCount

説明: 失敗した MQCTL 要求の数。

ID: MQIAMO_CTL5_FAILED

データ型: MQCFIN

戻り: 可能な場合。

StatCount

説明: 正常に行われた MQSTAT 要求の数。

ID: MQIAMO_STATS

データ型: MQCFIN

戻り: 可能な場合。

StatFailCount

説明: 失敗した MQSTAT 要求の数。

ID: MQIAMO_STATS_FAILED

データ型: MQCFIN

戻り: 可能な場合。

SubCountDurHighWater

説明: ある時間間隔内での永続サブスクリプション数の最高水準点。これは、SUBTYPE で索引付けされた値の配列です。

- 0 - システム内のすべての永続サブスクリプションの最高水準点
- 1 - 永続アプリケーション・サブスクリプションの最高水準点 (MQSUBTYPE_API)
- 2 - 永続管理サブスクリプションの最高水準点 (MQSUBTYPE_ADMIN)
- 3 - 永続プロキシ・サブスクリプションの最高水準点 (MQSUBTYPE_PROXY)

ID: MQIAMO_SUB_DUR_HIGHWATER

データ型: MQCFIL

戻り: 可能な場合。

SubCountDurLowWater

説明: ある時間間隔内での永続サブスクリプション数の最低水準点。これは、SUBTYPE で索引付けされた値の配列です。

- 0 - システム内のすべての永続サブスクリプションの最低水準点
- 1 - 永続アプリケーション・サブスクリプションの最低水準点 (MQSUBTYPE_API)
- 2 - 永続管理サブスクリプションの最低水準点 (MQSUBTYPE_ADMIN)
- 3 - 永続プロキシ・サブスクリプションの最低水準点 (MQSUBTYPE_PROXY)

ID: MQIAMO_SUB_DUR_LOWWATER

データ型: MQCFIL

戻り: 可能な場合。

SubCountNDurHighWater

説明: ある時間間隔内での非永続サブスクリプション数の最高水準点。これは、SUBTYPE で索引付けされた値の配列です。

- 0 - システム内のすべての非永続サブスクリプションの最高水準点
- 1 - 非永続アプリケーション・サブスクリプションの最高水準点 (MQSUBTYPE_API)
- 2 - 非永続管理サブスクリプションの最高水準点 (MQSUBTYPE_ADMIN)
- 3 - 非永続プロキシ・サブスクリプションの最高水準点 (MQSUBTYPE_PROXY)

ID: MQIAMO_SUB_NDUR_HIGHWATER

データ型: MQCFIL

戻り: 可能な場合。

SubCountNDurLowWater

説明: ある時間間隔内での非永続サブスクリプション数の最低水準点。これは、SUBTYPE で索引付けされた値の配列です。

- 0 - システム内のすべての非永続サブスクリプションの最低水準点
- 1 - 非永続アプリケーション・サブスクリプションの最低水準点 (MQSUBTYPE_API)
- 2 - 非永続管理サブスクリプションの最低水準点 (MQSUBTYPE_ADMIN)
- 3 - 非永続プロキシ・サブスクリプションの最低水準点 (MQSUBTYPE_PROXY)

ID: MQIAMO_SUB_NDUR_LOWWATER

データ型: MQCFIL

戻り: 可能な場合。

PutTopicCount

説明: 正常にトピックに書き込まれた持続メッセージと非持続メッセージの数。MQPUT1 呼び出しを使用して書き込まれたメッセージを除きます。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

注: 解決するとトピックになるキュー別名を使用して書き込まれたメッセージは、この値に含まれます。

ID: MQIAMO_TOPIC_PUTS
データ型: MQCFIL
戻り: 可能な場合。

PutTopicFailCount

説明: メッセージをトピックに書き込もうとして失敗した回数。
ID: MQIAMO_TOPIC_PUTS_FAILED
データ型: MQCFIN
戻り: 可能な場合。

Put1TopicCount

説明: MQPUT1 呼び出しを使用して正常にトピックに書き込まれた持続メッセージと非持続メッセージの数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

注: 解決するとトピックになるキュー別名を使用して書き込まれたメッセージは、この値に含まれます。

ID: MQIAMO_TOPIC_PUT1S.
データ型: MQCFIL
戻り: 可能な場合。

Put1TopicFailCount

説明: MQPUT1 呼び出しを使用してメッセージをトピックに書き込もうとして失敗した回数。
ID: MQIAMO_TOPIC_PUT1S_FAILED.
データ型: MQCFIN
戻り: 可能な場合。

PutTopicBytes

説明: 解決するとパブリッシュ操作になる持続メッセージと非持続メッセージに対する書き込み呼び出しを使用して書き込まれたバイト数。これはアプリケーションによって書き込まれたバイト数であり、サブスクライバーに送信された最終結果バイト数ではありません。後者の値については、[PublishMsgBytes](#) を参照してください。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_TOPIC_PUT_BYTES.
データ型: MQCFIL64.
戻り: 可能な場合。

PublishMsgCount

説明: ある時間間隔内にサブスクリプションに送信されたメッセージの数。このパラメーターは持続性値で索引付けされた整数リストです。 [リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_PUBLISH_MSG_COUNT
データ型: MQCFIL

戻り: 可能な場合。

PublishMsgBytes

説明: ある時間間隔内にサブスクリプションに送信されたバイト数。このパラメータは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_PUBLISH_MSG_BYTES

データ型: MQCFIL64.

戻り: 可能な場合。

キュー統計メッセージ・データ

このページを使用して、キュー統計メッセージの構造を確認します。

メッセージ名:	キュー統計メッセージ。
プラットフォーム:	すべて。ただし IBM MQ for z/OS を除く。
システム・キュー:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

説明: キュー・マネージャーの名前。

ID: MQCA_Q_MGR_NAME

データ型: MQCFST

最大長: MQ_Q_MGR_NAME_LENGTH

戻り: 常時

IntervalStartDate

説明: モニター期間の開始日。

ID: MQCAMO_START_DATE

データ型: MQCFST

最大長: MQ_DATE_LENGTH

戻り: 常時

IntervalStartTime

説明: モニター期間の開始時刻。

ID: MQCAMO_START_TIME

データ型: MQCFST

最大長: MQ_TIME_LENGTH

戻り: 常時

IntervalEndDate

説明: モニター期間の終了日。

ID: MQCAMO_END_DATE

データ型: MQCFST

最大長: MQ_DATE_LENGTH

戻り: 常時

IntervalEndTime

説明: モニター期間の終了時刻。

ID: MQCAMO_END_TIME

データ型: MQCFST

最大長: MQ_TIME_LENGTH

戻り: 常時

CommandLevel

説明: キュー・マネージャーのコマンド・レベル。

ID: MQIA_COMMAND_LEVEL

データ型: MQCFIN

戻り: 常時

ObjectCount

説明: 統計データの記録間隔中にアクセスされたキュー・オブジェクトの数。この値は、メッセージに含まれる QStatisticsData PCF グループの数に設定されます。

ID: MQIAMO_OBJECT_COUNT

データ型: MQCFIN

戻り: 常時

QStatisticsData

説明: キューの統計詳細を指定するグループ化されたパラメーター。

ID: MQGACF_Q_STATISTICS_DATA

データ型: MQCFGR

グループ内のパラ メーター:	<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>
-------------------	--

戻り:	常時
-----	----

QName

説明:	キューの名前。
ID:	MQCA_Q_NAME
データ型:	MQCFST
最大長:	MQ_Q_NAME_LENGTH
戻り:	常時

CreateDate

説明:	キューが作成された日付。
ID:	MQCA_CREATION_DATE
データ型:	MQCFST
最大長:	MQ_DATE_LENGTH
戻り:	常時

CreateTime

説明:	キューが作成された時刻。
ID:	MQCA_CREATION_TIME
データ型:	MQCFST
最大長:	MQ_TIME_LENGTH
戻り:	常時

QType

説明: キューのタイプ。
ID: MQIA_Q_TYPE
データ型: MQCFIN
値: MQOT_LOCAL
戻り: 常時

QDefinitionType

説明: キュー定義タイプ。
ID: MQIA_DEFINITION_TYPE
データ型: MQCFIN
値: 可能な値は以下のとおりです。

- MQQDT_PREDEFINED
- MQQDT_PERMANENT_DYNAMIC
- MQQDT_TEMPORARY_DYNAMIC

戻り: 可能な場合。

QMinDepth

説明: モニター期間中の最小キュー・サイズ。
ID: MQIAMO_Q_MIN_DEPTH
データ型: MQCFIN
組み込まれる PCF グループ: *QStatisticsData*
戻り: 可能な場合。

QMaxDepth

説明: モニター期間中の最大キュー・サイズ。
ID: MQIAMO_Q_MAX_DEPTH
データ型: MQCFIN
組み込まれる PCF グループ: *QStatisticsData*
戻り: 可能な場合。

AvgTimeOnQ

説明: 監視期間中にキューから破壊的に取り出したメッセージの平均待ち時間 (マイクロ秒)。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#) を参照してください。
ID: MQIAMO64_AVG_Q_TIME
データ型: MQCFIL64
組み込まれる PCF グループ: *QStatisticsData*
戻り: 可能な場合。

PutCount

説明: MQPUT1 要求を除く、正常にキューに書き込まれた持続メッセージと非持続メッセージの数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_PUTS

データ型: MQCFIL

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

PutFailCount

説明: メッセージをキューに書き込もうとして失敗した回数。

ID: MQIAMO_PUTS_FAILED

データ型: MQCFIN

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

Put1Count

説明: MQPUT1 呼び出しを使用して正常にキューに書き込まれた持続メッセージと非持続メッセージの数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO_PUT1S

データ型: MQCFIL

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

Put1FailCount

説明: MQPUT1 呼び出しを使用してメッセージを書き込もうとして失敗した回数。

ID: MQIAMO_PUT1S_FAILED

データ型: MQCFIN

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

PutBytes

説明: 書き込み要求でキューに書き込まれたバイト数。

ID: MQIAMO64_PUT_BYTES

データ型: MQCFIL64

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

GetCount

説明: 持続メッセージと非持続メッセージで正常に行われた破壊読み取り要求の回数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO_GETS

データ型: MQCFIL

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

GetFailCount

説明: 失敗した破壊取得要求の数。

ID: MQIAMO_GETS_FAILED

データ型: MQCFIN

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

GetBytes

説明: 持続メッセージと非持続メッセージの破壊書き込み要求で読み取られたバイト数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO64_GET_BYTES

データ型: MQCFIL64

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

BrowseCount

説明: 持続メッセージと非持続メッセージで正常に行われた非破壊読み取り要求の回数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#)を参照してください。

ID: MQIAMO_BROWSES

データ型: MQCFIL

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

BrowseFailCount

説明: 失敗した非破壊取得要求の数。

ID: MQIAMO_BROWSES_FAILED

データ型: MQCFIN

組み込まれる PCF グループ: *QStatisticsData*

戻り: 可能な場合。

BrowseBytes

説明: 持続メッセージと非持続メッセージの非破壊取得要求で読み取られたバイト数。このパラメーターは持続性値で索引付けされた整数リストです。[リファレンス・ノート 2](#) を参照してください。

ID: MQIAMO64_BROWSE_BYTES

データ型: MQCFIL64

組み込まれる PCF
グループ: QStatisticsData

戻り: 可能な場合。

NonQueuedMsgCount

説明: キューを迂回して、待機中のアプリケーションに直接転送されたメッセージの数。

キューの迂回は特定の状況でのみ行うことができます。この数は、アプリケーションが待機していた回数ではなく、IBM MQ がキューを迂回できた回数を表します。

ID: MQIAMO_MSGS_NOT_QUEUED

データ型: MQCFIN

組み込まれる PCF
グループ: QStatisticsData

戻り: 可能な場合。

ExpiredMsgCount

説明: 有効期限が切れたために、取り出す前に廃棄された持続メッセージと非持続メッセージの数。

ID: MQIAMO_MSGS_EXPIRED

データ型: MQCFIN

組み込まれる PCF
グループ: QStatisticsData

戻り: 可能な場合。

PurgeCount

説明: 消去されたメッセージの数。

ID: MQIAMO_MSGS_PURGED

データ型: MQCFIN

組み込まれる PCF
グループ: QStatisticsData

戻り: 可能な場合。

チャンネル統計メッセージ・データ

このページを使用して、チャンネル統計メッセージの構造を確認します。

メッセージ名: チャンネル統計メッセージ。

プラットフォーム:	すべて。ただし IBM MQ for z/OS を除く。
システム・キュー:	SYSTEM.ADMIN.STATISTICS.QUEUE.

QueueManager

説明:	キュー・マネージャーの名前。
ID:	MQCA_Q_MGR_NAME
データ型:	MQCFST
最大長:	MQ_Q_MGR_NAME_LENGTH
戻り:	常時。

IntervalStartDate

説明:	モニター期間の開始日。
ID:	MQCAMO_START_DATE
データ型:	MQCFST
最大長:	MQ_DATE_LENGTH
戻り:	常時。

IntervalStartTime

説明:	モニター期間の開始時刻。
ID:	MQCAMO_START_TIME
データ型:	MQCFST
最大長:	MQ_TIME_LENGTH
戻り:	常時。

IntervalEndDate

説明:	モニター期間の終了日。
ID:	MQCAMO_END_DATE
データ型:	MQCFST
最大長:	MQ_DATE_LENGTH
戻り:	常時。

IntervalEndTime

説明:	モニター期間の終了時刻。
ID:	MQCAMO_END_TIME
データ型:	MQCFST
最大長:	MQ_TIME_LENGTH
戻り:	常時。

CommandLevel

説明:	キュー・マネージャーのコマンド・レベル。
ID:	MQIA_COMMAND_LEVEL

データ型: MQCFIN
戻り: 常時。

ObjectCount

説明: 統計データの記録間隔中にアクセスされたチャンネル・オブジェクトの数。この値は、メッセージに含まれる `ChlStatisticsData PCF` グループの数に設定されます。

ID: MQIAMO_OBJECT_COUNT

データ型: MQCFIN

戻り: 常時。

ChlStatisticsData

説明: チャンネルの統計詳細を指定するグループ化されたパラメーター。

ID: MQGACF_CHL_STATISTICS_DATA

データ型: MQCFGR

グループ内のパラメーター:

- ChannelName*
- ChannelType*
- RemoteQmgr*
- ConnectionName*
- MsgCount*
- TotalBytes*
- NetTimeMin*
- NetTimeAvg*
- NetTimeMax*
- ExitTimeMin*
- ExitTimeAvg*
- ExitTimeMax*
- FullBatchCount*
- IncplBatchCount*
- AverageBatchSize*
- PutRetryCount*

戻り: 常時。

ChannelName

説明: チャンネルの名前。

ID: MQCACH_CHANNEL_NAME

データ型: MQCFST

最大長: MQ_CHANNEL_NAME_LENGTH

戻り: 常時。

ChannelType

説明: チャンネル・タイプ。

ID: MQIACH_CHANNEL_TYPE

データ型: MQCFIN

値: 指定可能な値は以下のとおりです。

MQCHT_SENDER

送信側チャネル。

MQCHT_SERVER

サーバー・チャネル。

MQCHT_RECEIVER

受信側チャネル。

MQCHT_REQUESTER

要求側チャネル。

MQCHT_CLUSRCVR

クラスター受信側チャネル。

MQCHT_CLUSSDR

クラスター送信側チャネル。

戻り: 常時。

RemoteQmgr

説明: リモート・キュー・マネージャーの名前。

ID: MQCA_REMOTE_Q_MGR_NAME

データ型: MQCFST

最大長: MQ_Q_MGR_NAME_LENGTH

戻り: 可能な場合。

ConnectionName

説明: リモート・キュー・マネージャーの接続名。

ID: MQCACH_CONNECTION_NAME

データ型: MQCFST

最大長: MQ_CONN_NAME_LENGTH

戻り: 可能な場合。

MsgCount

説明: 送受信された持続メッセージと非持続メッセージの数。

ID: MQIAMO_MSGS

データ型: MQCFIN

戻り: 可能な場合。

TotalBytes

説明: 持続メッセージと非持続メッセージで送受信されたバイト数。

ID: MQIAMO64_BYTES.

データ型: MQCFIN64.

戻り: 可能な場合。

NetTimeMin

説明: 記録間隔中に測定された最短チャネル往復時間 (マイクロ秒)。

ID: MQIAMO_NET_TIME_MIN
データ型: MQCFIN
戻り: 可能な場合。

NetTimeAvg

説明: 記録間隔中に測定された平均チャンネル往復時間 (マイクロ秒)。
ID: MQIAMO_NET_TIME_AVG
データ型: MQCFIN
戻り: 可能な場合。

NetTimeMax

説明: 記録間隔中に測定された最長チャンネル往復時間 (マイクロ秒)。
ID: MQIAMO_NET_TIME_MAX
データ型: MQCFIN
戻り: 可能な場合。

ExitTimeMin

説明: 記録間隔中にユーザー出口の実行に費やされた最短時間 (マイクロ秒)。
ID: MQIAMO_EXIT_TIME_MIN
データ型: MQCFIN
戻り: 可能な場合。

ExitTimeAvg

説明: 記録間隔中にユーザー出口の実行に費やされた平均時間 (マイクロ秒)。マイクロ秒で測定されます。
ID: MQIAMO_EXIT_TIME_AVG
データ型: MQCFIN
戻り: 可能な場合。

ExitTimeMax

説明: 記録間隔中にユーザー出口の実行に費やされた最長時間 (マイクロ秒)。マイクロ秒で測定されます。
ID: MQIAMO_EXIT_TIME_MAX
データ型: MQCFIN
戻り: 可能な場合。

FullBatchCount

説明: チャンネル属性 BATCHSZ または BATCHLIM の値に達したために送信された、チャンネルによって処理されたバッチの数。
ID: MQIAMO_FULL_BATCHES
データ型: MQCFIN
戻り: 可能な場合。

IncomplBatchCount

説明: チャンネル属性 BATCHSZ または BATCHLIM の値に到達せずに送信された、チャンネルによって処理されたバッチの数。

ID: MQIAMO_INCOMPLETE_BATCHES

データ型: MQCFIN

戻り: 可能な場合。

AverageBatchSize

説明: チャンネルで処理されたバッチの平均バッチ・サイズ。

ID: MQIAMO_AVG_BATCH_SIZE

データ型: MQCFIN

戻り: 可能な場合。

PutRetryCount

説明: 時間間隔中にメッセージの書き込みに失敗し、再試行ループに入った回数。

ID: MQIAMO_PUT_RETRIES

データ型: MQCFIN

戻り: 可能な場合。

リファレンス・ノート

このページを使用して、アカウントティング・メッセージと統計メッセージの構造の説明の参照先のノートを表示します。

以下のメッセージ・データ記述は、それぞれ各ノートを参照しています。

- [158 ページの『MQI アカウントティング・メッセージ・データ』](#)
- [169 ページの『キュー・アカウントティング・メッセージ・データ』](#)
- [180 ページの『MQI 統計メッセージ・データ』](#)
- [192 ページの『キュー統計メッセージ・データ』](#)
- [198 ページの『チャンネル統計メッセージ・データ』](#)

1. 次のパラメーターは、IBM MQ オブジェクトに関連するものです。このパラメーターは、以下の定数によって索引付けされる値の配列 (MQCFIL または MQCFIL64) です。

オブジェクト・タイプ	値コンテキスト
MQOT_Q (1)	キュー・オブジェクトに関連する値が含まれます。
MQOT_NAMELIST (2)	名前リスト・オブジェクトに関連する値が含まれます。
MQOT_PROCESS (3)	プロセス・オブジェクトに関連する値が含まれます。
MQOT_Q_MGR (5)	キュー・マネージャー・オブジェクトに関連する値が含まれます。
MQOT_CHANNEL (6)	チャンネル・オブジェクトに関連する値が含まれます。
MQOT_AUTH_INFO (7)	認証情報オブジェクトに関連する値が含まれます。

表 24. オブジェクト・タイプによって索引付けされる配列 (続き)	
オブジェクト・タイプ	値コンテキスト
MQOT_TOPIC (8)	トピック・オブジェクトに関連する値が含まれます。

注: MQCFIL または MQCFIL64 の 13 の値の配列が返されますが、そのうち意味のある値を上記にリストしています。

2. 次のパラメーターは、IBM MQ メッセージに関連するものです。このパラメーターは、以下の定数によって索引付けされる値の配列 (MQCFIL または MQCFIL64) です。

表 25. 持続性値によって索引付けされる配列	
定数	値
1	非持続メッセージの値が含まれます。
2	持続メッセージの値が含まれます。

注: これらの各配列の索引はゼロで始まるため、索引 1 は配列の第 2 行を指します。この表に記載されていない配列要素には、アカウント情報も統計情報も含まれていません。

アプリケーション・アクティビティ・トレース

アプリケーション・アクティビティ・トレースは、キュー・マネージャーに接続されたアプリケーションの動作に関する詳細な情報を生成します。これは、IBM MQ リソースと相互作用する際に、アプリケーションの動作をトレースし、アプリケーションによって使用されるパラメーターの詳細なビューを提供します。さらに、アプリケーションによって発行される MQI 呼び出しの順序も表示します。

イベント・モニター、メッセージ・モニター、アカウント情報および統計メッセージ、およびリアルタイム・モニターで提供される情報よりも多くの情報が必要な場合、アプリケーション・アクティビティ・トレースを使用します。

注: アクティビティ・トレースは、各アプリケーションの IBM MQ 接続で生成されます。したがって、アカウント情報・メッセージが使用可能になっている場合、アクティビティ・トレース操作は、各アプリケーションの MQI アカウント情報にカウントされます。

IBM MQ は、アプリケーション・アクティビティ・トレースのデータ収集方式として 2 つの方式をサポートしています。

- アプリケーション・アクティビティ・トレース情報の一元収集。アプリケーション・アクティビティ・トレースは、システム・キュー SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE にアクティビティ・トレース PCF メッセージを書き込むことによって収集され、読み取られます。
- アクティビティ・トレース・データのサブスクリプション。特別な IBM MQ システム・トピックに書き込まれます。

IBM MQ for z/OS はアクティビティ・トレースをサポートしていません。

ALW IBM MQ でサポートされているほとんどのプログラミング言語でアプリケーション名を指定できるようになったので、詳しい情報を確認するために、[サポートされているプログラミング言語でのアプリケーション名の指定を参照してください。](#)

アプリケーション・アクティビティ・トレース情報の一元収集の構成

アプリケーションのアクティビティ・トレース・メッセージは PCF メッセージです。アクティビティ・トレースは、構成ファイルを使用して構成します。アプリケーション・アクティビティ・トレース情報の一元収集を構成するには、ACTVTRC キュー・マネージャー属性を設定します。この設定は、MQCONNX オプションを使用して接続レベルで、またはアクティビティ・トレース構成ファイルを使用してアプリケーション・スタンザ・レベルで指定変更できます。

このタスクについて

アクティビティ・トレース・メッセージは、MQMD 構造 (PCF (MQCFH) ヘッダー構造にいくつかの PCF パラメーターが続く) で構成されます。一連の ApplicationTraceData PCF グループが PCF パラメーターに続きます。これらの PCF グループによって、アプリケーションがキュー・マネージャーへの接続中に実行する MQI 操作に関する情報が収集されます。mqat.ini という構成ファイルを使用してアクティビティ・トレースを構成します。

アプリケーションのアクティビティ・トレース情報を収集するかどうかを制御するには、以下の設定のうち 1 つ以上を構成します。

1. ACTVTRC キュー・マネージャー属性。
2. ACTVCONO 設定 (MQCONNX に渡される MQCNO 構造内)。
3. アクティビティ・トレース構成ファイル mqat.ini 内のアプリケーションに対応するスタンザ。

前述の項目の順序は重要です。ACTVTRC 属性は、ACTVCONO 設定によってオーバーライドされます。ACTVCONO 設定は、mqat.ini ファイルの設定によってオーバーライドされます。

特に指定のない限り、トレース項目は各操作の完了後に書き込まれます。これらの項目は、まずシステム・キュー SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE に書き込まれ、次に、アプリケーションがキュー・マネージャーから切断されたときにアプリケーション・アクティビティ・トレース・メッセージに書き込まれます。長期間実行されるアプリケーションでは、以下のいずれかのイベントが発生した場合に中間メッセージが書き込まれます。

- 接続の存続期間が定義済みのタイムアウト値に達した。
- 操作数が指定の数に達した。
- メモリー内に収集されたデータ量が、キューで許容される最大メッセージ長に達した。

タイムアウト値は、**ActivityInterval** パラメーターを使用して設定します。操作数は、**ActivityCount** パラメーターを使用して設定します。両方のパラメーターが、アクティビティ・トレース構成ファイル mqat.ini に指定されています。

アプリケーション・アクティビティ・トレースを有効にすると、パフォーマンスに影響する可能性があります。オーバーヘッドは、**ActivityCount** 設定と **ActivityInterval** 設定をチューニングして減らすことができます。212 ページの『アプリケーション・アクティビティ・トレースのパフォーマンス上の影響のチューニング』を参照してください。

アプリケーションのアクティビティ・トレース・メッセージの内容を見るための一番簡単な方法は、213 ページの『amqsact サンプル・プログラム』を使用する方法です。

手順

1. [205 ページの『アクティビティ・トレース情報の収集を制御するための ACTVTRC の設定』](#)。
2. [206 ページの『アクティビティ・トレース情報の収集を制御するための MQCONNX オプションの設定』](#)。
3. [207 ページの『mqat.ini を使用したアクティビティ・トレース動作の構成』](#)。
4. [212 ページの『アプリケーション・アクティビティ・トレースのパフォーマンス上の影響のチューニング』](#)。

アクティビティ・トレース情報の収集を制御するための ACTVTRC の設定

キュー・マネージャー属性 ACTVTRC を使用して、MQI アプリケーションのアクティビティ・トレース情報の収集を制御します。

このタスクについて

アプリケーションのアクティビティ・メッセージは、アプリケーションのアクティビティ・トレースが使用可能になった後に開始された接続に対してのみ生成されます。**ACTVTRC** パラメーターには、以下の値を指定できます。

ON

API アクティビティ・トレースの収集が有効になります。

OFF

API アクティビティ・トレースの収集が無効になります。

注: **ACTVTRC** 設定は、キュー・マネージャーの **ACTVCONO** パラメーターで指定変更することができます。**ACTVCONO** パラメーターを **ENABLED** に設定すると、MQCNO 構造の **Options** フィールドを使用して、特定の接続の **ACTVTRC** 設定をオーバーライドできます。「[206 ページの『アクティビティ・トレース情報の収集を制御するための MQCONNX オプションの設定』](#)」を参照してください。

例

ACTVTRC パラメーターの値を変更するには、MQSC コマンド **ALTER QMGR** を使用します。例えば、MQI アプリケーションのアクティビティ・トレース情報収集を有効にするには、次の MQSC コマンドを使用します。

```
ALTER QMGR ACTVTRC(ON)
```

次のタスク

アプリケーションのアクティビティ・トレース・メッセージの内容を見るための一番簡単な方法は、[213 ページの『amqsact サンプル・プログラム』](#)を使用する方法です。

アプリケーション・アクティビティ・トレースを有効にすると、パフォーマンスに影響する可能性があります。オーバーヘッドは、**ActivityCount** 設定と **ActivityInterval** 設定をチューニングして減らすことができます。[212 ページの『アプリケーション・アクティビティ・トレースのパフォーマンス上の影響のチューニング』](#)を参照してください。

アクティビティ・トレース情報の収集を制御するための MQCONNX オプションの設定

キュー・マネージャー属性 **ACTVCONO** を **ENABLED** に設定すると、MQCONNX 呼び出しで **ConnectOpts** パラメーターを使用して、接続ごとにアプリケーション・アクティビティ・レポートを使用可能または使用不可にすることができます。これらのオプションは、キュー・マネージャー属性 **ACTVTRC** で定義されたアクティビティ・トレース動作を指定変更します。また、これらのオプションは、アクティビティ・トレース構成ファイル `mqat.ini` の設定によって指定変更することができます。

手順

1. キュー・マネージャー属性 **ACTVCONO** を **ENABLED** に設定します。

注: アプリケーションが **ConnectOpts** パラメーターを使用してアプリケーションのアカウントリング動作を変更しようとしたときに、QMGR 属性 **ACTVCONO** が **DISABLED** に設定されていると、アプリケーションにエラーは返されず、アクティビティ・トレースの収集は、キュー・マネージャー属性またはアクティビティ・トレース構成ファイル `mqat.ini` で定義されます。

2. MQCONNX 呼び出しで、**ConnectOpts** パラメーターを `MQCNO_ACTIVITY_TRACE_ENABLED` に設定します。

MQCONNX 呼び出しの **ConnectOpts** パラメーターには、以下の値を指定できます。

MQCNO_ACTIVITY_TRACE_DISABLED

接続のアクティビティ・トレースを無効にします。

MQCNO_ACTIVITY_TRACE_ENABLED

接続のアクティビティ・トレースを有効にします。

注: アプリケーションが MQCONNX に対して `MQCNO_ACTIVITY_TRACE_ENABLED` と `MQCNO_ACTIVITY_TRACE_DISABLED` の両方を選択した場合、その呼び出しは理由コード `MQRC_OPTIONS_ERROR` で失敗します。

3. これらのアクティビティ・トレース設定が、アクティビティ・トレース構成ファイル `mqat.ini` の設定によってオーバーライドされていないことを確認してください。

207 ページの『[mqat.ini を使用したアクティビティ・トレース動作の構成](#)』を参照してください。

次のタスク

アプリケーションのアクティビティ・トレース・メッセージの内容を見るための一番簡単な方法は、213 ページの『[amqsact サンプル・プログラム](#)』を使用する方法です。

アプリケーション・アクティビティ・トレースを有効にすると、パフォーマンスに影響する可能性があります。オーバーヘッドは、**ActivityCount** 設定と **ActivityInterval** 設定をチューニングして減らすことができます。212 ページの『[アプリケーション・アクティビティ・トレースのパフォーマンス上の影響のチューニング](#)』を参照してください。

mqat.ini を使用したアクティビティ・トレース動作の構成

アクティビティ・トレース動作は、`mqat.ini` という構成ファイルを使用して構成します。このファイルは、アクティビティ・トレース・データの報告のレベルと頻度の定義に使用します。また、このファイルでは、アプリケーションの名前に基づいてアクティビティ・トレースを有効化および無効化するルールを定義できます。

このタスクについて

Linux

AIX

AIX and Linux システムでは、`mqat.ini` はキュー・マネージャーのデータ・ディレクトリーに配置されています。qm.ini ファイルと同じ場所です。

Windows

Windows システムでは、`mqat.ini` はキュー・マネージャーのデータ・ディレクトリー `C:\Program Files\IBM\WebSphere MQ\qmgrs\queue_manager_name` に配置されています。トレースされるアプリケーションを実行するユーザーには、このファイルの読み取りアクセス権が必要です。

注：IBM WebSphere MQ 7.1 またはそれ以前にマイグレーションされたキュー・マネージャーには、`mqat.ini` ファイルがありません。このような場合は、`mqat.ini` ファイルを手動で作成する必要があります。ファイルには 660 のアクセス権を設定する必要があります。

`mqat.ini` ファイルが変更された場合、新しく作成された IBM MQ 接続は、変更されたバージョンに従って処理される必要があります。既存の接続は、**ALTER QMGR** コマンドなどに続いてキュー・マネージャー・パラメーターが変更される場合を除き、引き続き以前のバージョンを使用します。

このファイルは、`mqs.ini` ファイルおよび `qm.ini` ファイルと同じスタンザ鍵とパラメーター値のペアのフォーマットに従います。

このファイルにはスタンザ **AllActivityTrace** が 1 つあり、すべてのアクティビティ・トレースにおけるアクティビティ・トレース・データのレポート作成のレベルと頻度がデフォルトで構成されます。

このファイルには、複数の **ApplicationTrace** スタンザを含めることもできます。それぞれのスタンザは、接続のアプリケーション名とルールの対応に基づいて、1 つ以上の接続のトレース動作のルールを定義します。

AllActivityTrace スタンザ

単一の **AllActivityTrace** スタンザによって、すべての IBM MQ 接続に適用されるアクティビティ・トレースの設定が定義されます (指定変更された場合を除く)。

AllActivityTrace スタンザの個々の値は、**ApplicationTrace stanza** スタンザのさらに特定された情報によりオーバーライドされる可能性があります。

複数の **AllActivityTrace** スタンザが指定された場合、最後のスタンザにある値が使用されます。選択された **AllActivityTrace** に欠落しているパラメーターは、デフォルト値を取ります。以前の **AllActivityTrace** スタンザのパラメーターと値は無視されます。

AllActivityTrace スタンザの下には以下のパラメーターを指定できます。

表 26. アクティビティー・トレース構成ファイルで使用可能なパラメーター/値のペア

名前	値 (太字がデフォルト値)	説明
ActivityInterval	0-99999999 (1)	1つのトレース・メッセージに含めるおおよその時間間隔 (秒単位)。この間隔の間に1つの接続で実行されるすべてのアクティビティーは、単一のメッセージに書き込まれます。この値が0の場合、トレース・メッセージは、接続が切断されたとき (またはアクティビティー・カウンタに達したとき) に書き込まれます。
ActivityCount	0-99999999 (100)	トレース・メッセージ間のMQI操作またはXA操作の数。この値が0の場合、トレース・メッセージは、接続が切断されたとき (またはアクティビティー間隔が経過したとき) に書き込まれます。
TraceLevel	LOW/ MEDIUM /HIGH	各操作でトレースされたパラメーター詳細の量。個々の操作の説明では、どのパラメーターが各トレース・レベルに含まれているかを詳しく示しています。
TraceMessageData	0 から 104 857 600 (最大 100 MB)	MQGET、MQPUT、MQPUT1、およびコールバック操作に関してトレースされたメッセージ・データの量 (バイト単位)。
StopOnGetTraceMsg	ON / OFF	アプリケーションをトレースするとき、そのアプリケーションがアクティビティー・トレース・メッセージも処理する場合には、ループが発生する可能性があるため、アクティビティー・トレースの使用は推奨されません。
SubscriptionDelivery	BATCHED /即時	1つ以上のアクティビティー・トレース・サブスクリプションが存在するときに、ActivityInterval および ActivityCount パラメーターを使用するかどうかを決定します。このパラメーターを IMMEDIATE に設定すると、マッチングするサブスクリプションがトレース・データにある場合、ActivityInterval および ActivityCount の値は有効値の1でオーバーライドされます。すべてのアクティビティー・トレース・レコードが、同じ接続に由来する他のレコードとまとめられることなく、遅延なしで即時にサブスクリプションに配信されます。

ApplicationTrace スタンザ

ApplicationTrace スタンザには、アプリケーション名に基づいてトレース対象または対象外の IBM MQ 接続を定義するルールが含まれています。オプションで、グローバル・トレース・レベルと頻度の設定を指定変更する Allsettings の下でデフォルトの動作を定義できます。

このスタンザには、ApplName、ApplFunction、および ApplClass の各パラメーターを含めることができます。これらのパラメーターは、スタンザが特定の接続に適用されるかどうかを決定する接続マッチング・ルールで定義されたマッチング・ルールに従って使用されます。

このスタンザには、対応する接続においてこのルールがアクティビティー・トレースをオンとオフのどちらにするかを決定するための Trace パラメーターが含まれていなければなりません。

オフのルールを使用して、より具体的なアプリケーション名のトレースを明示的に無効にしたり、キュー・マネージャーの ACTVTRC 設定またはアクティビティー・トレース接続オプションを指定変更したりできます。

ApplicationTrace スタンザの下には以下のパラメーターを指定できます。

表 27. アプリケーション・トレース構成ファイルで使用可能なパラメーター/値のペア		
名前	値 (太字がデフォルト値)	説明
トレース	ON / OFF (必須パラメーター - デフォルト値なし)	アクティビティー・トレース・スイッチ。このスイッチをアプリケーション固有のスタンザで使用して、現在のアプリケーション・スタンザの有効範囲でアクティビティー・トレースがアクティブであるかどうかを決定します。この値はキュー・マネージャーの ACTVTRC および ACTVCONO 設定を指定変更することに注意してください。
ApplName	文字ストリング (必須パラメーター - デフォルトなし)	<p>この値を使用して、ApplicationTrace スタンザが適用されるアプリケーションを決定します。これは、API 出口コンテキスト構造の ApplName 値 (MQMD.PutApplName に相当) にマッチングされます。ApplName 値の内容はアプリケーション環境によって異なります。</p> <p>z/OS 以外のプラットフォームの場合、MQAXC.ApplName のファイル名の部分のみが、スタンザの値とマッチングされます。比較が行われるときに、右端のパス・セパレーターの左側の文字は無視されます。</p> <p> z/OS z/OS アプリケーションの場合、MQAXC.ApplName 全体が、スタンザの値とマッチングされます。</p> <p>単一のワイルドカード文字 (*) を ApplName 値の最後に使用すると、その箇所以降の任意の数の文字に一致するように指定できます。ApplName 値が単一のワイルドカード文字 (*) に設定されている場合、ApplName 値はすべてのアプリケーションに一致します。</p>
  ApplFunction	文字ストリング (デフォルト値 *)	<p>この値を使用して、どのアプリケーション・プログラムに ApplicationTrace スタンザおよび ApplName 値が適用されるかを限定します。</p> <p>このスタンザはオプションですが、IBM i キュー・マネージャーに対してのみ有効です。単一のワイルドカード文字 (*) を ApplName 値の最後に使用すると、任意の数の文字に一致することができます。</p> <p>例えば、ApplName = * および ApplFunction = AMQSPUTO を指定している ApplicationTrace スタンザは、任意のジョブの AMQSPUTO プログラムのすべての呼び出しに適用されます。</p>
ApplClass	USER / MCA / ALL	アプリケーションのクラス。AppType 値が IBM MQ 接続とどのように対応するかについては、次の表を参照してください。

次の表に、AppClass 値が、接続 API 出口コンテキスト構造内の **APICallerType** フィールドと **APIEnvironment** フィールドにどのように対応しているかを示します。

表 28. applclass 値、および APICallerType フィールドと APIEnvironment フィールドへの対応関係

APPLCLASS	API 呼び出し元タイプ:	API 環境:	説明
USER	MQXACT_EXTERNAL	MQXE_OTHER	ユーザー・アプリケーションのみがトレースされます。
MCA	(任意の値)	MQXE_MCA MQXE_MCA_CLNTCONN MQXE_MCA_SVRCONN	クライアントおよびチャンネル (amqrmppa)
ALL	(任意の値)	(任意の値)	すべての接続がトレースされます。



重要: *USER* クラスはこれらと一致しないため、*MCA* の **APPLCLASS** をクライアント・ユーザー・アプリケーション用に使用する必要があります。

例えば、**amqsputc** サンプル・アプリケーションをトレースするには、以下のコードを使用できます。

```

ApplicationTrace:
ApplClass=MCA                                # Application type
                                             # Values: (USER | MCA | INTERNAL | ALL)
                                             # Default: USER
ApplName=amqsputc    # Application name (may be wildcarded)
                                             # (matched to app name without path)
                                             # Default: *
Trace=ON                                           # Activity trace switch for application
                                             # Values: ( ON | OFF )
                                             # Default: OFF
ActivityInterval=30                               # Time interval between trace messages
                                             # Values: 0-99999999 (0=off)
                                             # Default: 0
ActivityCount=1                                   # Number of operations between trace msgs
                                             # Values: 0-99999999 (0=off)
                                             # Default: 0
TraceLevel=MEDIUM                               # Amount of data traced for each operation
                                             # Values: LOW | MEDIUM | HIGH
                                             # Default: MEDIUM
TraceMessageData=1000                            # Amount of message data traced
                                             # Values: 0-1000000000
                                             # Default: 0
    
```

キュー・マネージャーの作成時に生成されるデフォルトの `mqat.ini` には、提供されたアクティビティー・トレース・サンプル **amqsact** のアクティビティー・トレースを明示的に無効化するための単一のルールが含まれています。

接続マッチング・ルール

キュー・マネージャーは、以下のルールを適用して、どのスタンザの設定を接続で使用するかを決定します。

1. **AllActivityTrace** スタンザで指定されている値は、接続に使用されます。ただし、その値が **ApplicationTrace** スタンザでも出現し、そのスタンザがポイント 2、3、および 4 で記述されている接続のためのマッチング基準を満たしている場合を除きます。
2. **ApplClass** は、IBM MQ 接続のタイプに対して突き合わされます。 **ApplClass** が接続タイプに一致しない場合、この接続ではスタンザは無視されます。
3. スタンザの **ApplName** 値が、接続の API 出口コンテキスト構造 (MQAXC) からの **ApplName** フィールドのファイル名部分と一致している。

ファイル名部分は、最後のパス区切り文字 (「/」または「¥」) 文字の右側の文字から取得されます。スタンザ **ApplName** にワイルドカード (*) が含まれている場合、ワイルドカードの左側の文字だけが、接続の **ApplName** にある同等の数の文字と比較されます。

例えば、「FRE*」というスタンザ値が指定された場合、最初の 3 文字だけが比較に使用されるので、「path/FREEDOM」と「path¥FREDDY」は一致しますが、「path/FRIEND」は一致しません。スタンザの **ApplName** 値が接続の **ApplName** に一致しない場合、この接続ではスタンザは無視されます。

4. 複数のスタンザが接続の **AppName** および **AppClass** に一致する場合は、最も具体的な **AppName** を持つスタンザが使用されます。

最も具体的な **AppName** とは、接続の **AppName** との突き合わせに使用される文字数が最も多いものと定義されています。

例えば、ini ファイルに **AppName = "FRE*"** というスタンザと **AppName = "FREE*"** という別のスタンザが含まれる場合、**AppName = "path/FREEDOM"** という接続に最も良く一致しているものとして選択されるのは、一致している文字が 4 文字の **AppName = "FREE*"** になります (一方 **AppName = "FRE*"** で一致するのは 3 文字のみです)。

5. ポイント 2、3、および 4 のルールを適用した後でも、接続の **AppName** およびその接続の **AppClass** と一致する複数のスタンザがある場合、最後に一致したものの値が使用され、他のすべてのスタンザは無視されます。

各ルールのデフォルト設定の指定変更

オプションで、**AllActivityTrace** スタンザの下のグローバル・トレース・レベルと頻度の設定は、**ApplicationTrace** スタンザに一致するこれらの接続において指定変更できます。

ApplicationTrace スタンザの下には以下のパラメーターを設定できます。設定しない場合、値は **AllActivityTrace** スタンザ設定から継承されます。

- **ActivityInterval**
- **ActivityCount**
- **TraceLevel**
- **TraceMessageData**
- **StopOnTraceMsg**

mqat.ini 構文

mqat.ini ファイルのフォーマットの構文規則には、以下のものがあります。

- ハッシュまたはセミコロンで始まるテキストは、行末までコメントと見なされます。
- 最初の意味のある (コメントでない) 行は、スタンザ・キーでなければなりません。
- スタンザ・キーは、スタンザの名前にコロンを続けたものから構成されます。
- パラメーター値ペアは、パラメーターの名前に等号を続け、その後に値を置いたものから構成されます。
- 単一のパラメーター値ペアだけを、1 つの行に置くことができます。(パラメーター値を、次の行に折り返すことはできません)。
- 先頭と末尾の空白文字は無視されます。スタンザ名、パラメーター名と値、パラメーター/値ペアでは、空白文字数に制限はありません。改行には意味があり、無視されません。
- どの行も最大長は 2048 文字です
- スタンザ・キー、パラメーター名、および定数パラメーター値には大/小文字の区別がありませんが、変数パラメーター値 (**AppName** と **DebugPath**) には大/小文字の区別があります。

アプリケーション・アクティビティ・トレース・ファイルの例

次の例は、アクティビティ・トレース ini ファイル中で、構成データがどのように指定されるかを示したものです。


```
AllActivityTrace:
ActivityInterval=1
ActivityCount=100
TraceLevel=MEDIUM
TraceMessageData=0
StopOnGetTraceMsg=ON

ApplicationTrace:
AppIName=amqs*
Trace=ON
TraceLevel=HIGH
TraceMessageData=1000

ApplicationTrace:
AppIName=amqsact*
Trace=OFF
```

上記の **AllActivityTrace** スタンザは、アクティビティ・トレースが、**ApplicationTrace** ルールまたはキュー・マネージャーの **ACTVTRC** 属性によって有効にされた場合、あるいはアプリケーションによってプログラマチックに有効にされた場合に、デフォルトでどのように実行されるのかを定義しています。

最初の **ApplicationTrace** スタンザは、名前が「amqs」で始まるアプリケーションによる MQI アクティビティがトレースされるというルールを定義しています。これらのアプリケーションに対して生成されるトレースは、詳細度が高く、最大 1000 バイトのメッセージ・データを格納します。アクティビティの間隔およびカウント・パラメーターは継承されます。

2つ目の **ApplicationTrace** スタンザは、名前が「amqsact」で始まるアプリケーション (アクティビティ・トレースのサンプル) のトレースをオフにするルールを定義しています。このルールは **amqsact** アプリケーションで以前に設定されていた「オン」のルールを指定変更するので、そのアプリケーションはトレースされません。

サンプルはまた、C サンプル・ディレクトリー (**amqsact.c** ファイルと同じディレクトリー) の **mqat.ini** というサンプルとして出荷されます。IBM MQ の以前のリリースから移行されたキュー・マネージャー用に、このファイルをキュー・マネージャー・データ・ディレクトリーにコピーできます。

次のタスク

アプリケーション・アクティビティ・トレースを有効にすると、パフォーマンスに影響する可能性があります。オーバーヘッドは、**ActivityCount** 設定と **ActivityInterval** 設定をチューニングして減らすことができます。212 ページの『[アプリケーション・アクティビティ・トレースのパフォーマンス上の影響のチューニング](#)』を参照してください。

アプリケーション・アクティビティ・トレースのパフォーマンス上の影響のチューニング

アプリケーション・アクティビティ・トレースを有効にすると、パフォーマンスに悪影響を及ぼす可能性があります。これは、必要なアプリケーションだけをトレースすることで軽減される場合があります。これを行うには、キューを排出するアプリケーションの数を増やしたり、**mqat.ini** 内の **ActivityInterval**、**ActivityCount**、および **TraceLevel** をチューニングしたりします。

このタスクについて

選択したアプリケーションまたはすべてのキュー・マネージャー・アプリケーションのアプリケーション・アクティビティ・トレースを有効にすると、追加のメッセージング・アクティビティが発生したり、キュー・マネージャーで追加のストレージ・スペースが必要になったりする場合があります。メッセージングのパフォーマンスが重要となる環境、例えば、作業負荷が高いアプリケーションや、サービス・レベル契約 (SLA) でメッセージング・プロバイダーからの必須の最小応答時間が規定されている場合は、アプリケーション・アクティビティ・トレースの収集が適切でなかったり、生成されるトレース・アクティビティ・メッセージの詳細度や頻度を調整する必要が生じたりする可能性があります。**mqat.ini** ファイル内の **ActivityInterval**、**ActivityCount**、および **TraceLevel** の事前設定値は、詳細度とパフォーマンスのバランスを考慮したデフォルト値です。しかし、ご使用のシステムの機能およびパフォーマンスの詳細な要件を満たすために、これらの値をチューニングすることができます。

手順

- 必要なアプリケーションのみトレースします。

そのためには、mqat.ini で ApplicationTrace アプリケーション固有スタンザを作成するか、MQCONNX 呼び出しの **MQCNO** 構造のオプション・フィールドで **MQCNO_ACTIVITY_TRACE_ENABLED** を指定するようにアプリケーションを変更します。207 ページの『mqat.ini を使用したアクティビティ・トレース動作の構成』および 206 ページの『アクティビティ・トレース情報の収集を制御するための MQCONNX オプションの設定』を参照してください。

- トレースを開始する前に、少なくとも 1 つのアプリケーションが実行されていて、それが **SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE** からアクティビティ・トレース・メッセージ・データを取り出す準備ができていないか確認します。
- キューを排出するアプリケーションの数を増やすことで、キュー項目数をできるだけ少なくする。
- mqat.ini ファイルの **TraceLevel** 値を、必要最小限のデータを収集するように設定します。

TraceLevel=LOW の場合に、メッセージング・パフォーマンスへの影響が最小である。『207 ページの『mqat.ini を使用したアクティビティ・トレース動作の構成』』を参照してください。

- mqat.ini にある **ActivityCount** および **ActivityInterval** の値をチューニングして、アクティビティ・トレース・メッセージを生成する頻度を調整します。

複数のアプリケーションをトレースする場合、アクティビティ・トレース・メッセージが生成される頻度は、**SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE** から除去される頻度より速くなる場合があります。ただし、アクティビティ・トレース・メッセージの生成頻度を減らすと、キュー・マネージャーが必要とするストレージ・スペースが増え、キューに書き込まれるときのメッセージのサイズも増えます。




次のタスク

アプリケーションのアクティビティ・トレース・メッセージの内容を見るための一番簡単な方法は、213 ページの『amqsact サンプル・プログラム』を使用する方法です。

amqsact サンプル・プログラム

アプリケーション・アクティビティ・トレース・メッセージを自動的にフォーマット設定する **amqsact** が、IBM MQ で提供されています。

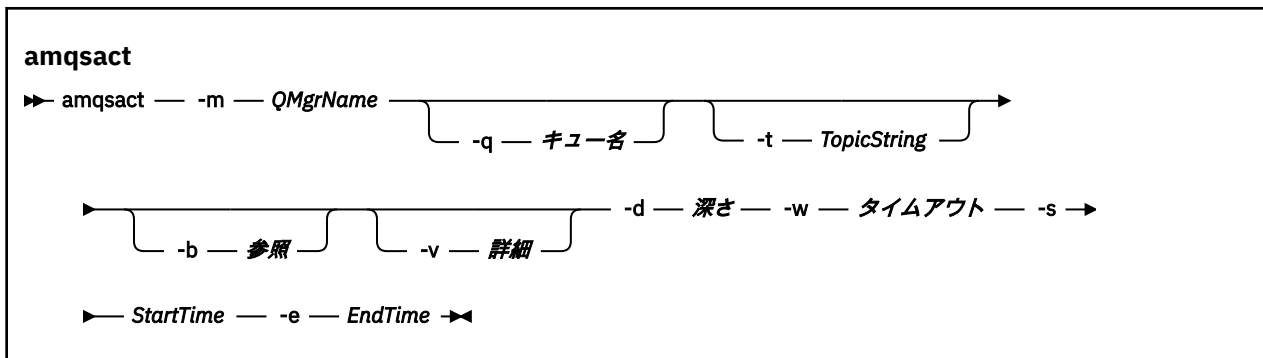
コンパイル済みプログラムが、以下のサンプル・ディレクトリーにあります。

-   AIX and Linux MQ_INSTALLATION_PATH/samp/bin 上
-  Windows MQ_INSTALLATION_PATH\tools\c\Samples\Bin 上

表示モード

デフォルトでは、表示モードの **amqsact** はメッセージを **SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE** で処理します。この動作は、キュー名またはトピック・ストリングを指定することで指定変更できます。

また、表示されるトレース期間を制御したり、表示後にアクティビティ・トレース・メッセージを除去するの保持するのかを指定したりできます。



表示モードの必須パラメーター

-m *QMgrName*

キュー・マネージャーの名前。

-d *Depth*

表示するレコード数。

-w *Timeout*

待機時間 (秒)。指定された期間にトレース・メッセージが出現しない場合、**amqsact** は終了します。

-s *StartTime*

レコードの処理の開始時刻。

-e *EndTime*

レコードの処理の終了時刻。

表示モードのオプション・パラメーター

-q *QName*

デフォルトのキュー名を指定変更する特定のキューを指定します。

-t *TopicString*

イベント・トピックにサブスクライブします。

-b

レコードのみ参照します。

-v

冗長出力

表示モードの出力例

MQCONN API 呼び出しにおいて、キュー・マネージャー *TESTQM* に対して冗長出力で **amqsact** を使用する場合は、次のようにします。

```
amqsact -m TESTQM -v
```

上記のコマンドを実行すると、以下の例のような出力が表示されます。

```
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
```

```

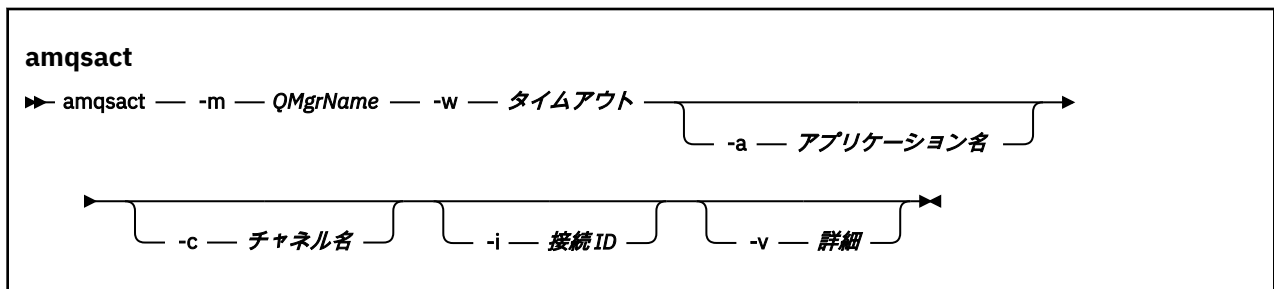
UserId: 'Emma_Bushby'
API Caller Type: MQXACT_EXTERNAL
API Environment: MQXE_OTHER
Application Function: ''
Appl Function Type: MQFUN_TYPE_UNKNOWN
Trace Detail Level: 2
Trace Data Length: 0
Pointer size: 4
Platform: MQPL_WINDOWS_7
MQI Operation: 0
Operation Id: MQXF_CONN
ApplicationId: 1
OperationDate: '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

```

動的モード

amqsact の引数としてアプリケーション名、チャンネル名、または接続 ID を指定することで、動的モードを有効にすることができます。名前にワイルドカード文字を使用することができます。

動的モードでは、システム・トピックへの非永続サブスクリプションを使用することで、サンプルの始めでアクティビティ・トレース・データが有効になります。アクティビティ・トレース・データの収集は、**amqsact** が停止した時に停止します。動的モードでは、**amqsact** のタイムアウトを指定する必要があります。**amqsact** の複数のコピーを同時に実行し、各インスタンスで任意のアクティビティ・トレース・データのコピーを受信することができます。



動的モードの必須パラメーター

-m QMgrName

キュー・マネージャーの名前。

-w Timeout

待機時間 (秒)。指定された期間にトレース・メッセージが出現しない場合、**amqsact** は終了します。

動的モードのオプション・パラメーター

-a Application name

メッセージの収集対象となるアプリケーション名を指定します。

-c Channel name

メッセージの収集対象となるチャンネルを指定します。

-i Connection ID

メッセージの収集対象となる接続を指定します。

-v

冗長出力

動的モードの出力例

以下のコマンドは、テキスト "amqs" で始まるアプリケーションによって行われたすべての接続に関するアクティビティ・トレース・メッセージを生成し、表示します。30 秒間アクティビティがないと、**amqsact** プログラムは終了し、新しいアクティビティ・トレース・データは生成されません。

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

以下のコマンドは、QMGR1.TO.QMGR2 チャンネルでのアクティビティに関するアクティビティ・トレース・メッセージを生成して表示します。10 秒間アクティビティがないと、**amqsact** プログラムは終了し、新しいアクティビティ・トレース・データは生成されません。

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

以下のコマンドは、CONN が「6B576B5420000701」で EXTCNN が「414D5143514D47523120202020202020」である既存の IBM MQ 接続でのアクティビティに関する冗長アクティビティ・トレース・メッセージを生成して表示します。1 分間アクティビティがないと、**amqsact** プログラムは終了し、新しいアクティビティ・トレース・データは生成されません。

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

アプリケーション・アクティビティ・トレース情報のサブスクライブ

キュー・マネージャー・レベルの構成によって情報を収集する代わりに、アプリケーション・アクティビティ・トレース情報を動的にサブスクライブすることができます。

このタスクについて

アプリケーション・アクティビティ・トレースは、IBM MQ リソースと対話するアプリケーションの動作をトレースし、アプリケーションで使用されたパラメーターの詳細を示します。さらに、アプリケーションによって発行される MQI 呼び出しの順序も表示します。

製品は、トレース・データをシステム・キューに書き込むだけでなく、キュー・マネージャー・レベルの構成によって情報を収集するのではなく、特殊な IBM MQ システム・トピックに書き込まれるアクティビティ・トレース・データを動的にサブスクライブする機能も備えています。

IBM MQ 9.0 以降、本製品ではこの目的のために出口を使用しないことに注意してください。以前に出口を使用してアプリケーション・アクティビティをトレースしたことがある場合は、アプリケーション・アクティビティ・トレースを収集するための代替方式を使用するように切り替える必要があります。

サブスクリプションを作成してアクティビティ・トレースを有効にします。トレース・データを一元収集する場合のようにキュー・マネージャーやアプリケーションの属性を設定する必要はありません。ただし、キュー・マネージャーやアプリケーションのレベルでトレースを無効にしてアクティビティ・トレースを明示的にブロックすると、マッチングするサブスクリプションへのアクティビティ・トレースの配信もブロックされます。

手順

- [216 ページの『アプリケーション・アクティビティ・トレースのサブスクリプション』](#)
- [217 ページの『アプリケーション・アクティビティ・トレースに対するサブスクリプションの作成』](#)
- [219 ページの『amqsact を使用したトレース・メッセージの表示』](#)
- [221 ページの『mqat.ini を使用したトレース・レベル』](#)

アプリケーション・アクティビティ・トレースのサブスクリプション

IBM MQ システム・トピックをサブスクライブして、アプリケーション・アクティビティ・トレース情報を収集できます。

トレースするアクティビティーを表す特殊な IBM MQ システム・トピック・ストリングをサブスクライブします。サブスクライブを行うと、アクティビティー・トレース・データ・メッセージが自動的に生成されて、サブスクリプション宛先キューにパブリッシュされます。サブスクリプションを削除すると、そのサブスクリプションのアクティビティー・トレース・データの生成は停止します。

サブスクリプションは、以下のいずれかのリソースのアクティビティーをトレースできます。

- 指定したアプリケーション
- 指定した IBM MQ チャンネル
- 既存の IBM MQ 接続

異なるトピック・ストリングや同じトピック・ストリングを使用して、複数のサブスクリプションを作成できます。同じシステム・アクティビティー・トレースのトピック・ストリングを使用して複数のサブスクリプションを作成する場合、各サブスクリプションがアクティビティー・トレース・データのコピーを受け取るため、パフォーマンスに不利な影響を与える可能性があります。

あらゆるレベルのアクティビティー・トレースを有効にすると、パフォーマンスに悪影響を与える可能性があります。サブスクリプションの数が多いほど、またはサブスクライブするリソースが多いほど、潜在的なパフォーマンスのオーバーヘッドも大きくなります。アクティビティー・トレースを収集するためのオーバーヘッドを最小化するために、データがメッセージに書き込まれてサブスクリプションに配信されるのは、そのデータが指すアプリケーション・アクティビティーとは非同期で行われます。多くの場合、複数の操作が1つのアクティビティー・トレース・データ・メッセージに書き込まれます。非同期操作により、アプリケーション動作の発生から、その動作を記録したトレース・データを受け取るまでには遅延時間が生じます。

アプリケーション・アクティビティー・トレースに対するサブスクリプションの作成

特定のトピックに対するサブスクリプションを作成して、アプリケーション・アクティビティー・トレース・データを収集することができます。

特定のシステム・トピック・ストリングに対するサブスクリプションを作成すると、該当するアクティビティー・トレースの PCF データ・メッセージがそのサブスクリプションに自動的にパブリッシュされます。トピックに対するサブスクリプションの詳細については、[メッセージのパブリッシュ/サブスクライブ](#)を参照してください。

トピック・ストリングの形式は次のとおりです。

```
$SYS/MQ/INFO/QMGR/qmgr_name/ActivityTrace/resource_type/resource_identifier
```

説明

- *qmgr_name* は、トレース対象のアプリケーションに接続されているキュー・マネージャーを指定します。*qmgr_name* は、すべての末尾空白文字を除去し、スラッシュ (/) 文字をアンパーサンド (&) に置き換えたキュー・マネージャーの名前です。文字。
- *resource_type* は、データを収集するリソースのタイプを指定します。以下のいずれかのストリングです。
 - *AppName* は、アプリケーションを指定します。要求は、*resource_identifier* によって指定されたアプリケーション名と一致するアプリケーション名を持つすべての IBM MQ 接続にサブスクライブします。
 - *ChannelName* は、IBM MQ チャンネルを指定します。
 - *ConnectionId* は、IBM MQ 接続を指定します。
- *resource_identifier* は、実際のリソースを指定します。形式は、次のようにリソース・タイプによって異なります。
 - リソース・タイプが *AppName* の場合、*resource_identifier* は、キュー・マネージャーによって認識されるアプリケーション名の末尾部分 (最後の「/」または「¥」に続く値) から末尾空白文字を除去したものです。この値は API 出口のコンテキスト構造体 (MQAXC) の *AppName* 値と一致します。MQSC コマンド **DISPLAY CONN** を使用すると、接続の *AppName* が *APPLTAG* 値として返されません。

- リソース・タイプが ChannelName の場合、*resource_identifier* はトレースされるチャンネルの名前です。チャンネル名が SVRCONN チャンネルを示す場合、接続されたクライアントのすべてのアプリケーション・アクティビティーがトレースされます。チャンネル名がキュー・マネージャー間のチャンネルを示す場合、着信メッセージと出力メッセージがトレースされます。*resource_identifier* は、すべての末尾ブランク文字が削除され、「/」文字が「&」に置き換えられたチャンネル名です。文字。
- リソース・タイプが ConnectionId の場合、*resource_identifier* は各接続に割り当てられた固有の接続 ID です。トピック・ストリング内の接続 ID は、16 進数ストリングとして記述された完全な 24 バイト値です。この値は、EXTCONN の後に、MQSC コマンド **DISPLAY CONN** から返された CONN 値を連結させたものです。

resource_identifier 内でワイルドカードを使用して、単一のサブスクリプションで複数のリソース ID とマッチングさせることができます。ワイルドカードは、デフォルトのトピック・スタイル（「#」または「+」）、あるいは文字スタイル（「*」または「?」）とすることができます。トピック・スタイルのワイルドカードを使用する場合は、それをリソース名の一部と組み合わせることはできません。それは、候補となるすべてのアプリケーション、チャンネル、または接続とマッチングさせるためにしか使用できません。ワイルドカードを使用すると、生成されるトレース・データのレベルが高くなり、パフォーマンスに影響を与える場合があります。

これらのトピック・ストリングをサブスクライブするには、「サブスクライブ」権限が必要です。システム・トピックは、キュー・マネージャーのトピック・ツリーのルートから権限を継承しません。ユーザーには、トピック・ツリー内の \$SYS/MQ ポイントまたはその下位にある管理トピック・オブジェクトへのアクセス権限を付与する必要があります。SYSTEM.ADMIN.TOPIC へのアクセス権限がある場合にはサブスクライブできますが、この場合、アクティビティー・トレースだけではなくすべての \$SYS/MQ トピック・ストリングへのアクセス権限が付与されます。アクセス権限をより厳密に制御するために、ツリーの下位のポイント（すべてのアクティビティー・トレース、または特定のアプリケーション名やチャンネル名など）に対して、新しい管理トピック・オブジェクトを定義できます。

例

以下の例は、Windows システム上で実行されている amqsput という名前のアプリケーションのトピック・ストリングを示しています。

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqsputc.exe
```

次の例は、チャンネルに対するトピック・ストリングを示しています。

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/SYSTEM.DEF.SVRCONN
```

次の例は、接続に対するトピック・ストリングを示しています。

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ConnectionId/  
414D5143514D4752312020202020202020206B576B5420000701
```

次の例は、キュー・マネージャー QMGR1 上のすべてのチャンネルのデータをトレースするサブスクリプションを作成するトピック・ストリングを示しています。

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/#
```

以下の例は、「amqs」で始まる名前を持つアプリケーションのデータをトレースするためのサブスクリプションを作成するトピック・ストリングを示しています（「*」ワイルドカードを使用する場合は、文字ワイルドカード・モデルを使用してサブスクリプションを作成する必要があります）。

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqs*
```

関連概念

[291 ページの『モニターとアクティビティー・トレースに関するシステム・トピック』](#)

キュー・マネージャー・トピック・ツリー内のシステム・トピックは、リソース・モニター (一部は統計メッセージの内容に似ています)、およびアプリケーション・アクティビティ・トレースのコンシューム方法として使用されます。

amqsact を使用したトレース・メッセージの表示

amqsact プログラム を使用して、トレース・メッセージを生成および表示することができます。

amqsact プログラムは、IBM MQ のサンプルです。このサンプルを使用するには、クライアント接続の実行可能ファイル **amqsactc** を使用する必要があります。この実行可能ファイルは、次のサンプル・ディレクトリにあります。

- Linux および UNIX プラットフォームの場合、`MQ_INSTALLATION_PATH/samp/bin64`
- Windows プラットフォームの場合、`MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`

amqsact は、次の 2 つの方法で使用できます。

表示モード

SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE に送信されるアクティビティ・トレース・データ・メッセージをフォーマット設定して表示します。

動的モード

amqsact を実行することによって、一連のリソースに対するサブスクリプションを作成し、生成されたアクティビティ・トレースを表示します。

表示モード

デフォルトでは、表示モードの **amqsact** はメッセージを SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE で処理します。この動作は、キュー名またはトピック・ストリングを指定することで指定変更できます。アクティビティ・トレースを、『[アプリケーションのアクティビティ・トレース情報の収集](#)』で説明しているいずれかの方法を使用して有効にする必要があります。表示されるトレース期間を制御したり、表示後にアクティビティ・トレース・メッセージを除去するのかわ保持するのかわを指定したりできます。表示モードでは、**amqsact** は以下の引数をとります。

-m queue_manager_name

必須。トレース・メッセージを収集するキュー・マネージャーを指定します。

-q queue_name

指定したキューに関連するトレース・メッセージのみを表示します。

-t topic_string

指定したトピックに関連するトレース・メッセージのみを表示します。

-b

表示後もトレース・メッセージを保持するように指定します。

-v

冗長モードでトレース・メッセージを表示します。

-d depth

表示するメッセージの数。

-w Timeout

タイムアウトを指定します。この期間内にトレース・メッセージが出現しない場合、**amqsact** は終了します。

-s start_time

この引数と **-e** 引数を使用して、期間を指定できます。指定した期間のトレース・メッセージが表示されます。

-e end_time

この引数と **-s** 引数を使用して、期間を指定できます。指定した期間のトレース・メッセージが表示されます。

例えば、以下のコマンドを使用すると、SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE に保持されているアクティビティ・トレース・メッセージが表示され、表示された後にメッセージが削除されます。

```
amqsact -m QMGR1
```

以下のコマンドを使用すると、指定のキュー SUB.QUEUE 上のアクティビティー・トレース・メッセージが表示され、表示された後にメッセージが削除されます。メッセージは、新しいメッセージがない期間が 30 秒間経過するまで表示され続けます。このコマンドは、アクティビティー・トレース・システムのトピック・ストリングに対するサブスクリプションなどと共に使用できます。

```
amqact -m QMGR1 -q SUB.QUEUE.1 -w 30
```

次のコマンドを使用すると、SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE に現在保持されているアクティビティー・トレース・データのうち、指定した 20 分間の間に発生したデータが詳細形式で表示されます。メッセージは、表示された後もキューに残されます。

```
amqsact -m QMGR1 -b -v -s 2014-12-31 23.50.00 -e 2015-01-01 00.10.00
```

動的モード

amqsact の引数としてアプリケーション名、チャンネル名、または接続 ID を指定することで、動的モードを有効にすることができます。名前にはワイルドカード文字を使用できます。動的モードでは、システム・トピックへの非永続サブスクリプションを使用することで、サンプルの始めでアクティビティー・トレース・データが有効になります。アクティビティー・トレース・データの収集は、**amqsact** が停止した時に停止します。動的モードでは、**amqsact** のタイムアウトを指定する必要があります。**amqsact** の複数のコピーを同時に実行し、各インスタンスで任意のアクティビティー・トレース・データのコピーを受信することができます。動的モードでは、**amqsact** は以下の引数をとります。

-m queue_manager_name

必須。トレース・メッセージを収集するキュー・マネージャーを指定します。

-w Timeout

必須。タイムアウトを指定します。この期間内にトレース・メッセージが出現しない場合、**amqsact** は終了します。

-a application_name

メッセージを収集するアプリケーションを指定します。

-c channel_name

メッセージの収集対象となるチャンネルを指定します。

-i connection_id

メッセージの収集対象となる接続を指定します。

-v

冗長モードでトレース・メッセージを表示します。

例えば、以下のコマンドは "amqsget.exe" という名前のアプリケーションによって行われるすべての接続について、アクティビティー・トレース・メッセージを生成し、表示します。30 秒間アクティビティーがないと、**amqsact** プログラムは終了し、新しいアクティビティー・トレース・データは生成されません。

```
amqsactc -m QMGR1 -w 30 -a amqsget.exe
```

以下のコマンドは、テキスト "amqs" で始まるアプリケーションによって行われるすべての接続について、アクティビティー・トレース・メッセージを生成し、表示します。30 秒間アクティビティーがないと、**amqsact** プログラムは終了し、新しいアクティビティー・トレース・データは生成されません。

```
amqsactc -m QMGR1 -w 30 -a amqs*
```


以下のコマンドは、QMGR1.TO.QMGR2 チャンネルでのアクティビティに関するアクティビティ・トレース・メッセージを生成して表示します。10 秒間アクティビティがないと、**amqsact** プログラムは終了し、新しいアクティビティ・トレース・データは生成されません。

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

以下のコマンドを使用すると、すべてのチャンネルのすべてのアクティビティについてのアクティビティ・トレース・メッセージが生成され、表示されます。10 秒間アクティビティがないと、**amqsact** プログラムは終了し、新しいアクティビティ・トレース・データは生成されません。

```
amqsactc -m QMGR1 -w 10 -c #
```

以下のコマンドは、CONN が「6B576B5420000701」で EXTCNN が「414D5143514D47523120202020202020」である既存の IBM MQ 接続でのアクティビティに関する冗長アクティビティ・トレース・メッセージを生成して表示します。1 分間アクティビティがないと、**amqsact** プログラムは終了し、新しいアクティビティ・トレース・データは生成されません。

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

mqat.ini を使用したトレース・レベル

キュー・マネージャーのトレース・レベルを構成するには、mqat.ini 構成ファイルの AllActivityTrace スタンザに値を設定します。

AllActivityTrace スタンザには、以下の値を設定できます。

ActivityInterval

トレース・メッセージ間の時間間隔 (秒単位)。アクティビティ・トレースではタイマー・スレッドが使用されないため、トレース・メッセージは、この時間が経過した瞬間に書き込まれるのではなく、この時間が経過した後初めて MQI 操作が実行されたときに書き込まれます。この値が 0 の場合、トレース・メッセージは、接続が切断されたとき (またはアクティビティ・カウントに達したとき) に書き込まれます。デフォルトは 1 です。

ActivityCount

トレース・メッセージ間の MQI 操作の数。この値が 0 の場合、トレース・メッセージは、接続が切断されたとき (またはアクティビティ間隔が経過したとき) に書き込まれます。デフォルトは 100 です。

TraceLevel

各操作でトレースされるパラメーター詳細の量。個々の操作の説明では、どのパラメーターが各トレース・レベルに含まれているかを詳しく示しています。LOW, MEDIUM、または HIGH に設定します。デフォルトは MEDIUM です。

TraceMessageData

MQGET、MQPUT、MQPUT1、およびコールバック操作に関してトレースされるメッセージ・データの量 (バイト単位)。デフォルトは 0 です。

StopOnGetTraceMsg

ON または OFF に設定できます。デフォルトは ON です。

SubscriptionDelivery

BATCHED または IMMEDIATE に設定できます。1 つ以上のアクティビティ・トレース・サブスクリプションが存在するときに、ActivityInterval および ActivityCount パラメーターを使用するかどうかを決定します。このパラメーターを IMMEDIATE に設定すると、マッチングするサブスクリプションがトレース・データにある場合、ActivityInterval および ActivityCount の値は有効値の 1 でオーバーライドされます。すべてのアクティビティ・トレース・レコードが、同じ接続に由来する他のレコードとまとめられることなく、遅延なしで即時にサブスクリプションに配信されます。IMMEDIATE 設定にすると、アクティビティ・トレース・データを収集するためのパフォーマンス・オーバーヘッドが増大します。デフォルトの設定値は BATCHED です。

アプリケーション・アクティビティ・トレース・メッセージ参照

このページでは、アプリケーション・アクティビティ・トレース・メッセージの形式およびこれらのメッセージに返される情報についての概要を説明します。

アプリケーション・アクティビティ・トレース・メッセージは、メッセージ記述子とメッセージ・データを含む標準 IBM MQ メッセージです。メッセージ・データには、IBM MQ アプリケーションによって実行された MQI 操作に関する情報、または IBM MQ システムで発生したアクティビティに関する情報が含まれます。

メッセージ記述子

- MQMD 構造

メッセージ・データ

- PCF ヘッダー (MQCFH)
- 常に返されるアプリケーション・アクティビティ・トレース・メッセージ・データ
- 操作固有のアプリケーション・アクティビティ・トレース・メッセージ・データ

アプリケーション・アクティビティ・トレース・メッセージ MQMD (メッセージ記述子)

このページを使用して、アプリケーション・アクティビティ・トレース・メッセージのメッセージ記述子と、イベント・メッセージのメッセージ記述子との間の違いを理解します。

アプリケーション・アクティビティ・トレース・メッセージのメッセージ記述子に含まれるパラメータと値は、イベント・メッセージのメッセージ記述子にあるものと同じですが、以下の例外があります。

Format

説明: メッセージ・データの形式名。

値: **MQFMT_ADMIN**
 管理メッセージ。

CorrelId

説明: 相関 ID。

値: アプリケーションの ConnectionId で初期化されます

MQCFH (PCF ヘッダー)

このページを使用して、アクティビティ・トレース・メッセージの MQCFH 構造に含まれた PCF 値を確認します。

アクティビティ・トレース・メッセージの場合、MQCFH 構造には以下の値が含まれます。

Type

説明: メッセージの内容を識別する構造タイプ。

データ型: MQLONG

値: MQCFT_APP_ACTIVITY

StrucLength

説明: MQCFH 構造のバイト単位の長さ。

データ型: MQLONG

値: MQCFH_STRUC_LENGTH

Version

説明: 構造体のバージョン番号。
データ型: MQLONG
値: MQCFH_VERSION_3

Command

説明: コマンド ID。このフィールドは、メッセージのカテゴリを識別します。
データ型: MQLONG
値: MQCMD_ACTIVITY_TRACE

MsgSeqNumber

説明: メッセージ順序番号 このフィールドは、関連メッセージのグループ内のメッセージの順序番号です。
データ型: MQLONG
値: 1

Control

説明: 制御オプション。
データ型: MQLONG
値: MQCFC_LAST

CompCode

説明: 完了コード。
データ型: MQLONG
値: MQCC_OK

Reason

説明: 完了コードを修飾する理由コード。
データ型: MQLONG
値: MQRC_NONE

ParameterCount

説明: パラメーター構造のカウンタ。このフィールドは、MQCFH 構造に続くパラメーター構造の数です。グループ構造 (MQCFGR) とそれに含まれるパラメーター構造は、1つの構造としてカウンタされます。
データ型: MQLONG
値: 1 以上

アプリケーションのアクティビティ・トレース・メッセージ・データ

PCFヘッダーの直後に、アクティビティ・トレースの時間間隔を記述するパラメーター・セットがあります。これらのパラメーターは、メッセージが書き込まれる際のメッセージの順序も示します。ヘッダーの後のフィールドの順序や数は決められていないので、将来に追加の情報を加えることができます。

メッセージ名: アクティビティ・トレース・メッセージ。
システム・キュー: SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

QueueManager

説明: キュー・マネージャーの名前。
ID: MQCA_Q_MGR_NAME
データ型: MQCFST
最大長: MQ_Q_MGR_NAME_LENGTH

QSGName



説明: キュー・マネージャーがメンバーになっているキュー共有グループの名前 (z/OS のみ)。
ID: MQCA_QSG_NAME
データ型: MQCFST
最大長: MQ_Q_MGR_NAME_LENGTH

HostName

説明: キュー・マネージャーを実行しているマシンのホスト名。
ID: MQCACF_HOST_NAME
データ型: MQCFST

IntervalStartDate

説明: モニター期間の開始日。
ID: MQCAMO_START_DATE
データ型: MQCFST
最大長: MQ_DATE_LENGTH

IntervalStartTime

説明: モニター期間の開始時刻。
ID: MQCAMO_START_TIME
データ型: MQCFST
最大長: MQ_TIME_LENGTH

IntervalEndDate

説明: モニター期間の終了日。
ID: MQCAMO_END_DATE
データ型: MQCFST
最大長: MQ_DATE_LENGTH

IntervalEndTime

説明: モニター期間の終了時刻。
ID: MQCAMO_END_TIME
データ型: MQCFST
最大長: MQ_TIME_LENGTH

CommandLevel

説明: IBM MQ のコマンド・レベル。
ID: MQIA_COMMAND_LEVEL
データ型: MQCFIN

SeqNumber

説明: 順序番号。通常はゼロです。この値は、長期間実行される接続の後続のレコードごとに増加します。
ID: MQIACF_SEQUENCE_NUMBER
データ型: MQCFIN

ApplicationName

説明: アプリケーションの名前(プログラム名)。
ID: MQCACF_APPL_NAME
データ型: MQCFST
最大長: MQ_APPL_NAME_LENGTH

ApplClass

説明: アクティビティを実行したアプリケーションのタイプ。有効値: MQAT_*
ID: MQIA_APPL_TYPE
データ型: MQCFIN

ApplicationPid

説明: アプリケーションのオペレーティング・システムのプロセス ID。
ID: MQIACF_PROCESS_ID
データ型: MQCFIN

UserId

説明: アプリケーションのユーザー ID コンテキスト。
ID: MQCACF_USER_IDENTIFIER
データ型: MQCFST
最大長: MQ_USER_ID_LENGTH

APICallerType

説明: アプリケーションのタイプ。有効値: MQXACT_EXTERNAL または MQXACT_INTERNAL。
ID: MQIACF_API_CALLER_TYPE
データ型: MQCFIN

Environment

説明: アプリケーションのランタイム環境。有効値: MQXE_*
ID: MQIACF_API_ENVIRONMENT

データ型: MQCFIN

ChannelName

説明: 接続に関連したチャンネル名。このパラメーターは、Environment パラメーターの値が MQXE_MCA または MQXE_MCA_SVRCONN のいずれかである場合にのみ返されます。

ID: MQCACH_CHANNEL_NAME

データ型: MQCFST

最大長: MQ_CHANNEL_NAME_LENGTH

ConnectionName

説明: 接続に関連したネットワーク接続名。このパラメーターは、Environment パラメーターの値が MQXE_MCA または MQXE_MCA_SVRCONN のいずれかである場合にのみ返されます。

ID: MQCACH_CONNECTION_NAME

データ型: MQCFST

最大長: MQ_CONN_NAME_LENGTH

ChannelType

説明: 接続に関連付けられているチャンネルのタイプ。このパラメーターは、Environment パラメーターの値が MQXE_MCA または MQXE_MCA_SVRCONN のいずれかである場合にのみ返されます。有効値: MQCHT_*

ID: MQIACH_CHANNEL_TYPE

データ型: MQCFIN

RemoteProduct

説明: 接続に関連付けられているリモート製品の ID。このパラメーターは、Environment パラメーターの値が MQXE_MCA または MQXE_MCA_SVRCONN のいずれかである場合にのみ返されます。

ID: MQCACH_REMOTE_PRODUCT

データ型: MQCFST

最大長: MQ_REMOTE_PRODUCT_LENGTH

RemoteVersion

説明: 接続に関連付けられているリモート製品のバージョン。このパラメーターは、Environment パラメーターの値が MQXE_MCA または MQXE_MCA_SVRCONN のいずれかである場合にのみ返されます。

ID: MQCACH_REMOTE_VERSION

データ型: MQCFST

最大長: MQ_REMOTE_VERSION_LENGTH

FunctionName

説明: 初期スレッドによって開始された最後の高水準機能の名前。

ID: MQCACF_APPL_FUNCTION

データ型: MQCFST

FunctionType

説明:	初期スレッドによって開始された最後の高水準機能のタイプ。有効値: MQFUN_*
ID:	MQIACF_APPL_FUNCTION_TYPE
データ型:	MQCFIN

Detail

説明:	接続に関して記録される詳細レベル。有効値: 1=LOW 2=MEDIUM 3=HIGH
ID:	MQIACF_TRACE_DETAIL
データ型:	MQCFIN

TraceDataLength

説明:	この接続に関してトレースされるメッセージ・データの長さ (バイト単位)。
ID:	MQIACF_TRACE_DATA_LENGTH
データ型:	MQCFIN

PointerSize

説明:	アプリケーションを実行しているプラットフォーム上のポインターの長さ (バイト単位) (バイナリー構造の解釈の支援用)。
ID:	MQIACF_POINTER_SIZE
データ型:	MQCFIN

Platform

説明:	キュー・マネージャーを実行しているプラットフォーム。有効値: MQPL_*
ID:	MQIA_PLATFORM
データ型:	MQCFIN

アプリケーション・アクティビティ MQL 操作の変数パラメーター

アプリケーション・アクティビティ・データ MQCFGR 構造体に、実行する操作に対応する PCF パラメーターのセットが続きます。各操作のパラメーターは、次のセクションで定義されています。

トレース・レベルは、トレースに含まれるパラメーターで必要なトレース細分度のレベルを示します。指定できるトレース・レベル値は以下のとおりです。

1. 低

このパラメーターは、アプリケーションに"低"、"中"、"高"のアクティビティ・トレースが設定されている場合に含まれます。この設定値は、パラメーターが常に操作の AppActivityData グループに含まれることを意味します。このパラメーターのセットは、アプリケーションが行う MQL 呼び出しをトレースし、それらが成功したかどうかを確認できるものです。

2. 中

このパラメーターは、アプリケーションに対して "medium" または "high" アクティビティ・トレースが構成されている場合にのみ、操作の AppActivityData グループに組み込まれます。このパラメーターのセットは、アプリケーションにより使用されるリソース (例えば、キューとトピック名) に関する情報を追加します。

3. 高

このパラメーターは、アプリケーションに対して "高" アクティビティ・トレースが構成されている場合にのみ、操作の AppActivityData グループに組み込まれます。このパラメーターのセットは、MQL

と XA 関数に渡された構造体のメモリー・ダンプを含みます。この理由で、これは MQI と XA 呼び出しで使用されるパラメーターに関するさらに詳細な情報を含みます。構造体メモリー・ダンプは、構造体のシャロー・コピーです。ポインターの間接参照を誤って試行しないようにするため、構造体中のポインター値は NULL に設定されます。

注: ダンプされる構造体のバージョンは、アプリケーションにより使用されるバージョンと同じである必要はありません。構造体は、API 交差出口、アクティビティー・トレース・コード、またはキュー・マネージャーで変更できます。キュー・マネージャーは、構造体をより最近のバージョンに変更することはできますが、構造体の以前のバージョンに変更することはできません。これを行うと、データを失うおそれがあります。

MQBACK

アプリケーションによって MQBACK MQI 機能が開始されました。

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ MQCFIN

Reason

説明: 操作の結果を示す理由コード
PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ MQCFIN

QMGrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION
トレース・レベル: 2
タイプ MQCFIN64

MQBEGIN

アプリケーションによって MQBEGIN MQI 機能が開始されました。

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ MQCFIN

Reason

説明: 操作の結果を示す理由コード

PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ MQCFIN

MQBO

説明: MQBEGIN オプションの構造。MQBEGIN 呼び出しで NULL ポインターが使用される場合、このパラメーターは含まれません。

PCF パラメーター: MQBACF_MQBO_STRUCT
トレース・レベル: 3
タイプ MQCFBS
長さ: MQBO 構造の長さ (バイト)。

QMgrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION
トレース・レベル: 2
タイプ MQCFIN64

MQCALLBACK

アプリケーションによって MQCALLBACK AX 機能が開始されました。

ObjectHandle

説明: オブジェクト・ハンドル

PCF パラメーター: MQIACF_HOBJ
トレース・レベル: 1
タイプ MQCFIN

CallType

説明: 関数が呼び出された理由。MQCBCT_* 値の 1 つです。

PCF パラメーター: MQIACF_CALL_TYPE
トレース・レベル: 1
タイプ MQCFIN

MsgBuffer

説明: メッセージ・データ。

PCF パラメーター: MQBACF_MESSAGE_DATA
トレース・レベル: 1
タイプ MQCFBS

長さ: 長さは、APPTRACE 構成で設定された TRACEDATA() パラメーターによって制御されます。TRACEDATA=NONE の場合は、このパラメーターは省略されます。

MsgLength

説明: メッセージの長さ。(MQCBC 構造体の DataLength フィールドから取られます)。

PCF パラメーター: MQIACF_MSG_LENGTH

トレース・レベル: 1

タイプ MQCFIN

HighResTime

説明: 1970 年 1 月 1 日深夜 0 時 (UTC) からのマイクロ秒で表す操作の時刻。

注: このタイマーの精度は、プラットフォームの高精度タイマー・サポートに従って変わります。

PCF パラメーター: MQIAMO64_HIGHRES_TIME

トレース・レベル: 2

タイプ MQCFIN64

ReportOptions

説明: レポート・メッセージのオプション

PCF パラメーター: MQIACF_REPORT

トレース・レベル: 2

タイプ MQCFIN

MsgType

説明: メッセージのタイプ

PCF パラメーター: MQIACF_MSG_TYPE

トレース・レベル: 2

タイプ MQCFIN

Expiry

説明: メッセージの存続時間。

PCF パラメーター: MQIACF_EXPIRY

トレース・レベル: 2

タイプ MQCFIN

Format

説明: メッセージ・データの形式名。

PCF パラメーター: MQCACH_FORMAT_NAME

トレース・レベル: 2

タイプ MQCFST

長さ: MQ_FORMAT_LENGTH

Priority

説明: メッセージ優先順位
PCF パラメーター: MQIACF_PRIORITY
トレース・レベル: 2
タイプ MQCFIN

Persistence

説明: メッセージの持続性
PCF パラメーター: MQIACF_PERSISTENCE
トレース・レベル: 2
タイプ MQCFIN

MsgId

説明: メッセージ ID
PCF パラメーター: MQBACF_MSG_ID
トレース・レベル: 2
タイプ MQCFBS
長さ: MQ_MSG_ID_LENGTH

CorrelId

説明: 相関 ID
PCF パラメーター: MQBACF_CORREL_ID
トレース・レベル: 2
タイプ MQCFBS
長さ: MQ_CORREL_ID_LENGTH

ObjectName

説明: オープンされたオブジェクトの名前。
PCF パラメーター: MQCACF_OBJECT_NAME
トレース・レベル: 2
タイプ MQCFST
長さ: MQ_Q_NAME_LENGTH

ResolvedQName

説明: メッセージが取り出されたキューのローカル名。
PCF パラメーター: MQCACF_RESOLVED_Q_NAME
トレース・レベル: 2
タイプ MQCFST
長さ: MQ_Q_NAME_LENGTH

ReplyToQueue

説明: MQ_Q_NAME_LENGTH
PCF パラメーター: MQCACF_REPLY_TO_Q
トレース・レベル: 2
タイプ MQCFST

ReplyToQMgr

説明: MQ_Q_MGR_NAME_LENGTH
PCF パラメーター: MQCACF_REPLY_TO_Q_MGR
トレース・レベル: 2
タイプ MQCFST

CodedCharSetId

説明: メッセージ・データの文字セット ID
PCF パラメーター: MQIA_CODED_CHAR_SET_ID
トレース・レベル: 2
タイプ MQCFIN

Encoding

説明: メッセージ・データの数値エンコード。
PCF パラメーター: MQIACF_ENCODING
トレース・レベル: 2
タイプ MQCFIN

PutDate

説明: MQ_PUT_DATE_LENGTH
PCF パラメーター: MQCACF_PUT_DATE
トレース・レベル: 2
タイプ MQCFST

PutTime

説明: MQ_PUT_TIME_LENGTH
PCF パラメーター: MQCACF_PUT_TIME
トレース・レベル: 2
タイプ MQCFST

ResolvedQName

説明: ResolvedType が MQOT_Q の場合に、ObjectHandle によって参照されるキュー名。
PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME
トレース・レベル: 2
タイプ MQCFST

長さ: MQ_Q_NAME_LENGTH。

ResObjectString

説明: ResolvedType が MQOT_TOPIC の場合に、ObjectHandle によって参照されるオブジェクト名。

PCF パラメーター: MQCACF_RESOLVED_OBJECT_STRING

トレース・レベル: 2

タイプ MQCFST

長さ: 長さは可変です。

ResolvedType

説明: ObjectHandle によって参照されるオブジェクトのタイプ。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。

PCF パラメーター: MQIACF_RESOLVED_TYPE

トレース・レベル: 2

タイプ MQCFIN

PolicyName

説明: このメッセージに適用されたポリシー名。

注: AMS で保護されたメッセージのみ

PCF パラメーター: MQCA_POLICY_NAME

トレース・レベル: 2

タイプ MQCFST

長さ: MQ_OBJECT_NAME_LENGTH

XmitqMsgId

説明: 伝送キュー・ヘッダー内のメッセージのメッセージ ID。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQBACF_XQH_MSG_ID

トレース・レベル: 2

タイプ MQCFBS

長さ: MQ_MSG_ID_LENGTH

XmitqCorrelId

説明: 伝送キュー・ヘッダー内のメッセージの相関 ID。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQBACF_XQH_CORREL_ID

トレース・レベル: 2

タイプ MQCFBS

長さ: MQ_CORREL_ID_LENGTH

XmitqPutTime

説明: 伝送キュー・ヘッダー内のメッセージの書き込み時刻。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_PUT_TIME
トレース・レベル: 2
タイプ MQCFST
長さ: MQ_PUT_TIME_LENGTH

XmitqPutDate

説明: 伝送キュー・ヘッダー内のメッセージの書き込み日付。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_PUT_DATE
トレース・レベル: 2
タイプ MQCFST
長さ: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

説明: 伝送キュー・ヘッダー内のメッセージのリモート・キュー宛先。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_REMOTE_Q_Name
トレース・レベル: 2
タイプ MQCFST
長さ: MQ_Q_NAME_LENGTH

XmitqRemoteQMGr

説明: 伝送キュー・ヘッダー内のメッセージのメッセージ ID。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_REMOTE_Q_MGR
トレース・レベル: 2
タイプ MQCFST
長さ: MQ_MSG_ID_LENGTH

MsgDescStructure

説明: MQMD 構造体。バージョン 4 の MQGMO を使用して、MQMD の代わりにメッセージ・ハンドルを戻すよう要求した場合は、このパラメーターは省略されます。

PCF パラメーター: MQBACF_MQMD_STRUCT
トレース・レベル: 3
タイプ MQCFBS
長さ: MQMD 構造体のバイト単位の長さです (実際のサイズは構造体のバージョンに応じて異なります)。

GetMsgOptsStructure

説明: MQGMO 構造体。
PCF パラメーター: MQBACF_MQGMO_STRUCTURE
トレース・レベル: 3
タイプ MQCFBS
長さ: MQGMO 構造体のバイト単位の長さです (実際のサイズは構造体のバージョンに応じて異なります)。

MQCBCContextStructure

説明: MQCBC 構造体。
PCF パラメーター: MQBACF_MQCBC_STRUCTURE
トレース・レベル: 3
タイプ MQCFBS
長さ: MQCBC 構造体のバイト単位の長さです (実際のサイズは構造体のバージョンに応じて異なります)。

QMgrOpDuration

説明: キュー・マネージャー内でのおおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION
トレース・レベル: 2
タイプ MQCFIN64

MQCB

アプリケーションがコールバック管理 MQI 関数を開始済みです。

CallbackOperation

説明: コールバック管理関数の操作。MQOP_* 値の 1 つに設定します。
PCF パラメーター: MQIACF_MQCB_OPERATION
トレース・レベル: 1
タイプ MQCFIN

CallbackType

説明: コールバック関数のタイプ (MQCBD 構造体からの CallbackType フィールド)。MQCBT_* 値の 1 つに設定します。
PCF パラメーター: MQIACF_MQCB_TYPE
トレース・レベル: 1
タイプ MQCFIN

CallbackOptions

説明: コールバック・オプション。MQCBDO_* 値の 1 つに設定します。
PCF パラメーター: MQIACF_MQCB_OPTIONS
トレース・レベル: 1
タイプ MQCFIN

CallbackFunction

説明: 関数呼び出しとして開始された場合の、コールバック関数を指すポインター。
PCF パラメーター: MQBACF_MQCB_FUNCTION
トレース・レベル: 1
タイプ MQCFBS
長さ: MQPTR のサイズです。

CallbackName

説明: 動的にリンクされたプログラムとして開始された場合の、コールバック関数の名前。
PCF パラメーター: MQCACF_MQCB_NAME
トレース・レベル: 1
タイプ MQCFST
長さ: MQCHAR128 のサイズです。

ObjectHandle

説明: オブジェクト・ハンドル
PCF パラメーター: MQIACF_HOBJ
トレース・レベル: 1
タイプ MQCFIN

MaxMsgLength

説明: 最大メッセージ長。整数または特殊値 MQCBD_FULL_MSG_LENGTH に設定されます。
PCF パラメーター: MQIACH_MAX_MSG_LENGTH
トレース・レベル: 2
タイプ MQCFIN

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ MQCFIN

Reason

説明: 操作の結果を示す理由コード

PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ MQCFIN

ResolvedQName

説明: ResolvedType が MQOT_Q の場合に、ObjectHandle によって参照されるキュー名。
PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME
トレース・レベル: 2
タイプ MQCFST
長さ: MQ_Q_NAME_LENGTH。

ResObjectString

説明: ResolvedType が MQOT_TOPIC の場合に、ObjectHandle によって参照されるオブジェクト名。
PCF パラメーター: MQCACF_RESOLVED_OBJECT_STRING
トレース・レベル: 2
タイプ MQCFST
長さ: 長さは可変です。

ResolvedType

説明: ObjectHandle によって参照されるオブジェクトのタイプ。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。
PCF パラメーター: MQIACF_RESOLVED_TYPE
トレース・レベル: 2
タイプ MQCFIN

Callback DescriptorStructure

説明: MQCBD 構造体。NULL MQCBC 値が MQCB 呼び出しに渡される場合は、このパラメーターは省略されます。
PCF パラメーター: MQBACF_MQCBD_STRUCT
トレース・レベル: 3
タイプ MQCFBS
長さ: MQCBC 構造体のバイト単位の長さです。

MsgDescStructure

説明: MQMD 構造体。NULL MQMD 値が MQCB 呼び出しに渡される場合は、MsgDescStructure パラメーターは省略されます。
PCF パラメーター: MQBACF_MQMD_STRUCT
トレース・レベル: 3
タイプ MQCFBS
長さ: MQMD 構造体のバイト単位の長さ (実際のサイズは構造体のバージョンによって異なる)

GetMsgOptsStructure

説明: MQGMO 構造体。NULL MQGMO 値が MQCB 呼び出しに渡される場合は、このパラメーターは省略されます。

PCF パラメーター: MQBACF_MQGMO_STRUCTURE

トレース・レベル: 3

タイプ MQCFBS

長さ: MQGMO 構造体のバイト単位の長さ (実際のサイズは構造体のバージョンによって異なる)

QMgrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。

所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。

注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION

トレース・レベル: 2

タイプ MQCFIN64

MQCLOSE

アプリケーションによって MQCLOSE MQI 機能が開始されました。

ObjectHandle

説明: オブジェクト・ハンドル

PCF パラメーター: MQIACF_HOBJ

トレース・レベル: 1

タイプ MQCFIN

CloseOptions

説明: 閉じるときのオプション

PCF パラメーター: MQIACF_CLOSE_OPTIONS

トレース・レベル: 1

タイプ MQCFIN

CompCode

説明: 操作の結果を示す完了コード

PCF パラメーター: MQIACF_COMP_CODE

トレース・レベル: 1

タイプ MQCFIN

Reason

説明: 操作の結果を示す理由コード

PCF パラメーター: MQIACF_REASON_CODE

トレース・レベル: 1
タイプ MQCFIN

ResolvedQName

説明: ResolvedType が MQOT_Q の場合に、ObjectHandle によって参照されるキュー名。
PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME
トレース・レベル: 2
タイプ MQCFST
長さ: MQ_Q_NAME_LENGTH。

ResObjectString

説明: ResolvedType が MQOT_TOPIC の場合に、ObjectHandle によって参照されるオブジェクト名。
PCF パラメーター: MQCACF_RESOLVED_OBJECT_STRING
トレース・レベル: 2
タイプ MQCFST
長さ: 長さは可変です。

ResolvedType

説明: ObjectHandle によって参照されるオブジェクトのタイプ。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。
PCF パラメーター: MQIACF_RESOLVED_TYPE
トレース・レベル: 2
タイプ MQCFIN

QMGrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。
PCF パラメーター: MQIAMO64_QMGR_OP_DURATION
トレース・レベル: 2
タイプ MQCFIN64

MQCMIT

アプリケーションによって MQCMIT MQI 機能が開始されました。

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1

タイプ MQCFIN

Reason

説明: 操作の結果を示す理由コード

PCF パラメーター: MQIACF_REASON_CODE

トレース・レベル: 1

タイプ MQCFIN

QMGrOpDuration

説明: キュー・マネージャー内でのおおよその API 呼び出し所要時間 (マイクロ秒単位)。

所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。

注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION

トレース・レベル: 2

タイプ MQCFIN64

MQCONN および MQCONNX

アプリケーションが MQCONN または MQCONNX MQI 関数を開始済みです。

ConnectionId

説明: 使用できる場合は接続 ID、使用できない場合は MQCONNID_NONE

PCF パラメーター: MQBACF_CONNECTION_ID

トレース・レベル: 1

タイプ: MQCFBS

最大長: MQ_CONNECTION_ID_LENGTH

QueueManagerName

説明: MQCONN(X) 呼び出しで使用されるキュー・マネージャーの (未解決の) 名前

PCF パラメーター: MQCA_Q_MGR_NAME

トレース・レベル: 1

タイプ: MQCFST

最大長: MQ_Q_MGR_NAME_LENGTH

CompCode

説明: 操作の結果を示す完了コード

PCF パラメーター: MQIACF_COMP_CODE

トレース・レベル: 1

タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード
PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ: MQCFIN

ConnectOptions

説明: MQCNO_* 値から派生する接続オプション
注: MQCONNX のみ
PCF パラメーター: MQIACF_CONNECT_OPTIONS
トレース・レベル: 2
タイプ: MQCFIN

ConnectionOptionsStructure

説明: MQCNO 構造体
注: MQCONNX のみ
PCF パラメーター: MQBACF_MQCNO_STRUCT
トレース・レベル: 3
タイプ: MQCFBS
最大長: MQCNO 構造体のバイト単位の長さ (実際のサイズは構造体のバージョンに応じて異なります)

ChannelDefinitionStructure

説明: MQCD 構造体
注: クライアント接続のみ
PCF パラメーター: MQBACF_MQCD_STRUCT
トレース・レベル: 3
タイプ: MQCFBS
最大長: MQCD 構造体のバイト単位の長さ (実際のサイズは構造体のバージョンに応じて異なります)

QMgrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。
PCF パラメーター: MQIAMO64_QMGR_OP_DURATION
トレース・レベル: 2
タイプ: MQCFIN64

MQCTL

アプリケーションによって MQCTL MQI 機能が開始されました。

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード
PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ: MQCFIN

CtlOperation

説明: MQOP_* 値のいずれか
PCF パラメーター: MQIACF_CTL_OPERATION
トレース・レベル: 1
タイプ: MQCFIN

QMgrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION
トレース・レベル: 2
タイプ: MQCFIN64

MQDISC

アプリケーションによって MQDISC MQI 機能が開始されました。

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード
PCF パラメーター: MQIACF_REASON_CODE

トレース・レベル: 1
タイプ: MQCFIN

MQGET

アプリケーションによって MQGET MQI 機能が開始されました。

ObjectHandle

説明: オブジェクト・ハンドル
PCF パラメーター: MQIACF_HOBJ
トレース・レベル: 1
タイプ: MQCFIN

GetOptions

説明: MQGMO.Options の get オプション
PCF パラメーター: MQIACF_GET_OPTIONS
トレース・レベル: 1
タイプ: MQCFIN

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード
PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ: MQCFIN

MsgBuffer

説明: メッセージ・データ。 TRACEDATA=NONE の場合、このパラメーターは省略されます。
PCF パラメーター: MQBACF_MESSAGE_DATA
トレース・レベル: 1
タイプ: MQCFBS
最大長: 長さは、APPTRACE 構成で設定された TRACEDATA() パラメーターによって制御されます。(トレース・メッセージに MQIACF_TRACE_DATA_LENGTH として組み込まれます。)

MsgLength

説明: メッセージの長さ。
PCF パラメーター: MQIACF_MSG_LENGTH
トレース・レベル: 1

タイプ: MQCFIN

HighResTime

説明: 1970年1月1日午前0時(UTC)からの操作時間(マイクロ秒)

注: このタイマーの精度は、プラットフォームの高精度タイマー・サポートに従って変わります。

PCF パラメーター: MQIAMO64_HIGHRES_TIME

トレース・レベル: 2

タイプ: MQCFIN64

BufferLength

説明: アプリケーションが提供するバッファの長さ

PCF パラメーター: MQIACF_BUFFER_LENGTH

トレース・レベル: 2

タイプ: MQCFIN

ObjectName

説明: オープンされたオブジェクトの名前

PCF パラメーター: MQCACF_OBJECT_NAME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH

ResolvedQName

説明: メッセージが取り出されたキューのローカル名。

PCF パラメーター: MQCACF_RESOLVED_Q_NAME

トレース・レベル: 2

タイプ: MQCFST

最大長: MQ_Q_NAME_LENGTH

ReportOptions

説明: メッセージ・レポート・オプション

PCF パラメーター: MQIACF_REPORT

トレース・レベル: 2

タイプ: MQCFIN

MsgType

説明: メッセージのタイプ

PCF パラメーター: MQIACF_MSG_TYPE

トレース・レベル: 2

タイプ: MQCFIN

Expiry

説明: メッセージの存続時間。
PCF パラメーター: MQIACF_EXPIRY
トレース・レベル: 2
タイプ: MQCFIN

Format

説明: メッセージ・データの形式名。
PCF パラメーター: MQCACH_FORMAT_NAME
トレース・レベル: 2
タイプ: MQCFST
最大長: MQ_FORMAT_LENGTH

Priority

説明: メッセージ優先順位
PCF パラメーター: MQIACF_PRIORITY
トレース・レベル: 2
タイプ: MQCFIN

Persistence

説明: メッセージの持続性
PCF パラメーター: MQIACF_PERSISTENCE
トレース・レベル: 2
タイプ: MQCFIN

MsgId

説明: メッセージ ID
PCF パラメーター: MQBACF_MSG_ID
トレース・レベル: 2
タイプ: MQCFBS
最大長: MQ_MSG_ID_LENGTH

CorrelId

説明: 相関 ID
PCF パラメーター: MQBACF_CORREL_ID
トレース・レベル: 2
タイプ: MQCFBS
最大長: MQ_CORREL_ID_LENGTH

ReplyToQueue

説明:

PCF パラメーター: MQCACF_REPLY_TO_Q
トレース・レベル: 2
タイプ: MQCFST
最大長: MQ_Q_NAME_LENGTH

ReplyToQMgr

説明:
PCF パラメーター: MQCACF_REPLY_TO_Q_MGR
トレース・レベル: 2
タイプ: MQCFST
最大長: MQ_Q_MGR_NAME_LENGTH

CodedCharSetId

説明: メッセージ・データの文字セット ID
PCF パラメーター: MQIA_CODED_CHAR_SET_ID
トレース・レベル: 2
タイプ: MQCFIN

Encoding

説明: メッセージ・データの数値エンコード。
PCF パラメーター: MQIACF_ENCODING
トレース・レベル: 2
タイプ: MQCFIN

PutDate

説明:
PCF パラメーター: MQCACF_PUT_DATE
トレース・レベル: 2
タイプ: MQCFST
最大長: MQ_PUT_DATE_LENGTH

PutTime

説明:
PCF パラメーター: MQCACF_PUT_TIME
トレース・レベル: 2
タイプ: MQCFST
最大長: MQ_PUT_TIME_LENGTH

ResolvedQName

説明: ResolvedType が MQOT_Q の場合に、ObjectHandle によって参照されるキュー名。
PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME

トレース・レベル: 2
タイプ MQCFST
長さ: MQ_Q_NAME_LENGTH。

ResObjectString

説明: ResolvedType が MQOT_TOPIC の場合に、ObjectHandle によって参照されるオブジェクト名。
PCF パラメーター: MQCACF_RESOLVED_OBJECT_STRING
トレース・レベル: 2
タイプ MQCFST
長さ: 長さは可変です。

ResolvedType

説明: ObjectHandle によって参照されるオブジェクトのタイプ。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。
PCF パラメーター: MQIACF_RESOLVED_TYPE
トレース・レベル: 2
タイプ MQCFIN

PolicyName

説明: このメッセージに適用されたポリシー名。
注: AMS で保護されたメッセージのみ
PCF パラメーター: MQCA_POLICY_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_OBJECT_NAME_LENGTH

XmitqMsgId

説明: 伝送キュー・ヘッダー内のメッセージのメッセージ ID。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ
PCF パラメーター: MQBACF_XQH_MSG_ID
トレース・レベル: 2
タイプ: MQCFBS
長さ: MQ_MSG_ID_LENGTH

XmitqCorrelId

説明: 伝送キュー・ヘッダー内のメッセージの相関 ID。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ
PCF パラメーター: MQBACF_XQH_CORREL_ID
トレース・レベル: 2
タイプ: MQCFBS

長さ: MQ_CORREL_ID_LENGTH

XmitqPutTime

説明: 伝送キュー・ヘッダー内のメッセージの書き込み時刻。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_PUT_TIME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_PUT_TIME_LENGTH

XmitqPutDate

説明: 伝送キュー・ヘッダー内のメッセージの書き込み日付。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_PUT_DATE

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

説明: 伝送キュー・ヘッダー内のメッセージのリモート・キュー宛先。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_REMOTE_Q_NAME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH

XmitqRemoteQMgr

説明: 伝送キュー・ヘッダー内のメッセージのリモート・キュー・マネージャー宛先。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_REMOTE_Q_MGR

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH

MsgDescStructure

説明: MQMD 構造体。

PCF パラメーター: MQBACF_MQMD_STRUCT

トレース・レベル: 3

タイプ: MQCFBS

最大長: MQMD 構造体のバイト単位の長さ (実際のサイズは構造体のバージョンによって異なる)

GetMsgOptsStructure

説明: MQGMO 構造体。

PCF パラメーター: MQBACF_MQGMO_STRUCT

トレース・レベル: 3

タイプ: MQCFBS

最大長: MQGMO 構造体のバイト単位の長さ (実際のサイズは構造体のバージョンによって異なる)

QMGrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。

所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。

注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION

トレース・レベル: 2

タイプ: MQCFIN64

MQINQ

アプリケーションが MQINQ MQI 関数を開始しています

ObjectHandle

説明: オブジェクト・ハンドル

PCF パラメーター: MQIACF_HOBJ

トレース・レベル: 1

タイプ: MQCFIN

CompCode

説明: 操作の結果を示す完了コード

PCF パラメーター: MQIACF_COMP_CODE

トレース・レベル: 1

タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード

PCF パラメーター: MQIACF_REASON_CODE

トレース・レベル: 1

タイプ: MQCFIN

SelectorCount

説明: Selectors 配列で提供されるセレクターのカウント。
PCF パラメーター: MQIACF_SELECTOR_COUNT
トレース・レベル: 2
タイプ: MQCFIN

Selectors

説明: 値が MQINQ によって戻される必要がある属性のリスト (整数または文字)。
PCF パラメーター: MQIACF_SELECTORS
トレース・レベル: 2
タイプ: MQCFIL

ResolvedQName

説明: ResolvedType が MQOT_Q の場合に、ObjectHandle によって参照されるキュー名。
PCF パラメーター: MQCACF_RESOLVED_Q_NAME
トレース・レベル: 2
タイプ: MQCFST
最大長: MQ_Q_NAME_LENGTH

ResObjectString

説明: ResolvedType が MQOT_TOPIC の場合に、ObjectHandle によって参照されるオブジェクト名。
PCF パラメーター: MQCACF_RESOLVED_OBJECT_STRING
トレース・レベル: 2
タイプ: MQCFST
最大長: 長さは可変です

ResolvedType

説明: ObjectHandle によって参照されるオブジェクトのタイプ。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。
PCF パラメーター: MQIACF_RESOLVED_TYPE
トレース・レベル: 2
タイプ: MQCFIN

IntAttrCount

説明: 照会操作によって戻される整数属性の数
PCF パラメーター: MQIACF_INTATTR_COUNT
トレース・レベル: 3
タイプ: MQCFIN

IntAttrrs

説明: 照会操作によって戻される整数属性の値。このパラメーターは、MQINQによって戻されるときに IntAttrCount が > 0 である場合にのみ存在します。

PCF パラメーター: MQIACF_INT_ATTRS

トレース・レベル: 3

タイプ: MQCFIL

CharAttrrs

説明: 照会操作によって戻される文字属性。値は一緒に連結されます。このパラメーターは、MQINQによって戻されるときに CharAttrLength が > 0 である場合にのみ含まれます。

PCF パラメーター: MQCACF_CHAR_ATTRS

トレース・レベル: 3

タイプ: MQCFST

QMgrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION

トレース・レベル: 2

タイプ: MQCFIN64

MQOPEN

アプリケーションが MQOPEN MQI 関数を開始しています

ObjectType

説明: MQOT.ObjectType で渡されるオブジェクト・タイプ

PCF パラメーター: MQIACF_OBJECT_TYPE

トレース・レベル: 1

タイプ: MQCFIN

ObjectName

説明: 何らかのキュー・ネーム・レゾリュションが試行される前に、MQI 呼び出しに渡されるオブジェクトの名前。

PCF パラメーター: MQCACF_OBJECT_NAME

トレース・レベル: 1

タイプ: MQCFST

最大長: MQ_Q_NAME_LENGTH

ObjectQMgrName

説明: 何らかのキュー・ネーム・レゾリューションが試行される前に、MQI呼び出しに渡されるオブジェクト・キュー・マネージャーの名前。

PCF パラメーター: MQCACF_OBJECT_Q_MGR_NAME

トレース・レベル: 1

タイプ: MQCFST

最大長: MQ_Q_MGR_NAME_LENGTH

ObjectHandle

説明: オブジェクト・ハンドル

PCF パラメーター: MQIACF_HOBJ

トレース・レベル: 1

タイプ: MQCFIN

CompCode

説明: 操作の結果を示す完了コード

PCF パラメーター: MQIACF_COMP_CODE

トレース・レベル: 1

タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード

PCF パラメーター: MQIACF_REASON_CODE

トレース・レベル: 1

タイプ: MQCFIN

OpenOptions

説明: オブジェクトをオープンするために使用されるオプション

PCF パラメーター: MQIACF_OPEN_OPTIONS

トレース・レベル: 1

タイプ: MQCFIN

AlternateUserId

説明: MQOO_ALTERNATE_USER_AUTHORITY が指定されている場合だけ含まれます。
is specified

PCF パラメーター: MQCACF_ALTERNATE_USERID

トレース・レベル: 2

タイプ: MQCFST

最大長: MQ_USER_ID_LENGTH

RecsPresent

説明: 存在しているオブジェクト名レコードの数。MQOD バージョン >= MQOD_VERSION_2 の場合だけ含まれます。

PCF パラメーター: MQIACF_RECS_PRESENT

トレース・レベル: 1

タイプ: MQCFIN

KnownDestCount

説明: オープンに成功したローカル・キューの数。MQOD バージョン >= MQOD_VERSION_2 の場合だけ含まれます。

PCF パラメーター: MQIACF_KNOWN_DEST_COUNT

トレース・レベル: 1

タイプ: MQCFIN

UnknownDestCount

説明: オープンに成功したリモート・キューの数。MQOD バージョン >= MQOD_VERSION_2 の場合だけ含まれます。

PCF パラメーター: MQIACF_UNKNOWN_DEST_COUNT

トレース・レベル: 1

タイプ: MQCFIN

InvalidDestCount

説明: オープンに失敗したキューの数。MQOD バージョン >= MQOD_VERSION_2 の場合だけ含まれます。

PCF パラメーター: MQIACF_INVALID_DEST_COUNT

トレース・レベル: 1

タイプ: MQCFIN

DynamicQName

説明: MQOPEN 呼び出しへの入力として渡された、動的キュー名。

PCF パラメーター: MQCACF_DYNAMIC_Q_NAME

トレース・レベル: 2

タイプ: MQCFST

最大長: MQ_Q_NAME_LENGTH

ResolvedLocalQName 1 2

説明: ネーム解決が実行された後の、ローカル・キュー名を含みます。(例えば、リモート・キューの場合、これは伝送キューの名前です。)

PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME

トレース・レベル: 2

タイプ: MQCFST

範囲: MQOD.Version が MQOD_VERSION_3 より小さい場合、これには MQOPEN 呼び出しが完了した後の MQOD.ObjectName フィールドの値が含まれます。
MQOD.Version が MQOD_VERSION_3 以上になっている場合、これには MQOD の値が含まれます。 ResolvedQName フィールド。

最大長: MQ_Q_NAME_LENGTH

ResolvedLocalQMgrName ^{1 2}

説明: ネーム解決が実行された後の、ローカル・キュー・マネージャー名。

PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_MGR

トレース・レベル: 2

タイプ: MQCFST

範囲: MQOD バージョン >= MQOD_VERSION_3 の場合のみ

最大長: MQ_Q_MGR_NAME_LENGTH

ResolvedQName ^{1 2}

説明: ネーム解決が実行された後のキュー名。

PCF パラメーター: MQCACF_RESOLVED_Q_NAME

トレース・レベル: 2

タイプ: MQCFST

範囲: MQOD.Version が MQOD_VERSION_3 より小さい場合、これには MQOPEN 呼び出しが完了した後の MQOD.ObjectName フィールドの値が含まれます。
MQOD.Version が MQOD_VERSION_3 以上になっている場合、これには MQOD の値が含まれます。 ResolvedQName フィールド。

最大長: MQ_Q_NAME_LENGTH

ResolvedQMgrName ^{1 2}

説明: ネーム解決が実行された後の、キュー・マネージャー名を含みます。
MQOD.Version が MQOD_VERSION_3 より小さい場合、この値は MQOD の値を含んでいます。 MQOPEN 呼び出しの完了後の ObjectQMgrName フィールド。
MQOD.Version が MQOD_VERSION_3 以上になっている場合、これには MQOD の値が含まれます。 ResolvedQMgrName フィールド。

PCF パラメーター: MQCACF_RESOLVED_Q_MGR

トレース・レベル: 2

タイプ: MQCFST

最大長: MQ_Q_MGR_NAME_LENGTH

AlternateSecurityId

説明: 代替セキュリティ ID。 MQOD.Version が MQOD_VERSION_3 以上で、MQOO_ALTERNATE_USER_AUTHORITY が指定されており、MQOD.AlternateSecurityId が MQSID_NONE に等しくない場合だけ、現れます。

PCF パラメーター: MQBACF_ALTERNATE_SECURITYID

トレース・レベル: 2

タイプ: MQCFBS

最大長: MQ_SECURITY_ID_LENGTH

ObjectString

説明:	長いオブジェクト名。MQOD.Version が MQOD_VERSION_4 以上で、MQOD.ObjectString の VSLength フィールドが MQVS_NULL_TERMINATED かゼロより大きい場合にだけ含まれます。
PCF パラメーター:	MQCACF_OBJECT_STRING
トレース・レベル:	2
タイプ:	MQCFST
最大長:	長さは可変です。

SelectionString

説明:	選択ストリング。MQOD.Version が MQOD_VERSION_4 以上で、かつ MQOD の VSLength フィールドの場合にのみ含まれます。SelectionString が MQVS_NULL_TERMINATED またはゼロより大きい値になっています。
PCF パラメーター:	MQCACF_SELECTION_STRING
トレース・レベル:	2
タイプ:	MQCFST
最大長:	長さは可変です。

ResObjectString

説明:	ObjectName フィールドで提供されている名前をキュー・マネージャーが解決した後の、長いオブジェクト名です。MQOD.Version が MQOD_VERSION_4 以上で、VSLength が MQVS_NULL_TERMINATED かゼロより大きい場合にだけ、トピック・オブジェクトを参照するトピックとキュー別名として含まれます。
PCF パラメーター:	MQCACF_RESOLVED_OBJECT_STRING
トレース・レベル:	2
タイプ:	MQCFST
最大長:	長さは可変です。

ResolvedType

説明:	開いている解決済み (ベース) オブジェクトのタイプ。MQOD.Version が MQOD_VERSION_4 以上の場合にだけ含まれます。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。
PCF パラメーター:	MQIACF_RESOLVED_TYPE
トレース・レベル:	2
タイプ:	MQCFIN

QMgrOpDuration

説明:	キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。 所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。 注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。
PCF パラメーター:	MQIAMO64_QMGR_OP_DURATION

トレース・レベル: 2

タイプ MQCFIN64

アプリケーション・アクティビティー配布リスト PCF グループのヘッダー構造

MQOPEN 関数が配布リストをオープンすると、MQOPEN パラメーターにはその配布リスト内のキューごとに1つずつ AppActivityDistList PCF グループが組み込まれ、最大で RecsPresent に指定されている構造体の数まで組み込まれます。Ap-pActivityDistList PCF グループは、MQOR 構造体と MQRR 構造体からの情報を結合し、キュー名を識別し、キュー上でのオープン操作の結果を示します。AppActivityDistList グループは、常に以下の MQCFGR 構造体を使用して開始されます。

MQCFGR フィールド	値	説明
タイプ	MQCFT_GROUP	
StrucLength	MQCFGR 構造体の長さ (バイト)	
パラメーター	MQGACF_APP_DIST_LIST	配布リスト・グループ・パラメーター
ParameterCount	4	このグループ内に含まれる MQCFGR 構造体の後に続くパラメーター構造の数

ObjectName

説明: 配布リスト内のキューの名前 MQ_Q_NAME_LENGTH。MQOR 構造体が提供されている場合のみ組み込まれます。

PCF パラメーター: MQCACF_OBJECT_NAME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH。MQOR 構造体が提供されている場合のみ組み込まれます。

ObjectQMgrName

説明: ObjectName で名前指定されたキューが定義されているキュー・マネージャーの名前。

PCF パラメーター: MQCACF_OBJECT_Q_MGR_NAME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_MGR_NAME_LENGTH MQOR 構造体が提供されている場合のみ組み込まれます。

CompCode

説明: このオブジェクトのオープンの結果を示す完了コード。MQRR 構造体が提供されており、MQOPEN の理由コードが MQRC_MULTIPLE_REASONS の場合のみ組み込まれます。

¹ このパラメーターは、オープンされているオブジェクトがキューに解決され、キューが MQOO_INPUT_*, MQOO_OUTPUT、または MQOO_BROWSE に対してオープンされている場合にのみ、組み込まれます。

² ResolvedLocalQName パラメーターが含まれるのは、ResolvedQName パラメーターと異なる場合のみです。

PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 2
タイプ: MQCFIN

Reason

説明: このオブジェクトのオープンの結果を示す理由コード。MQRR 構造体が提供されており、MQOPEN の理由コードが MQRC_MULTIPLE_REASONS の場合のみ組み込まれます。

PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 2
タイプ: MQCFIN

MQPUT

アプリケーションによって MQPUT MQI 機能が開始されました。

ObjectHandle

説明: オブジェクト・ハンドル
PCF パラメーター: MQIACF_HOBJ
トレース・レベル: 1
タイプ: MQCFIN

PutOptions

説明: MQPMO.Options の書き込みオプション
PCF パラメーター: MQIACF_PUT_OPTIONS
トレース・レベル: 1
タイプ: MQCFIN

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード
PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ: MQCFIN

MsgBuffer

説明: メッセージ・データ。
PCF パラメーター: MQBACF_MESSAGE_DATA
トレース・レベル: 1

タイプ: MQCFBS
長さ: 長さは、APPTTRACE 構成で設定された TRACEDATA() パラメーターによって制御されます。TRACEDATA=NONE の場合は、このパラメーターは省略されます。

MsgLength

説明: メッセージの長さ。
PCF パラメーター: MQIACF_MSG_LENGTH
トレース・レベル: 1
タイプ: MQCFIN

RecsPresent

説明: 書き込みメッセージ・レコードの数、または存在する応答レコードの数。MQPMO バージョン >= MQPMO_VERSION_2 の場合だけ含まれます。
PCF パラメーター: MQIACF_RECS_PRESENT
トレース・レベル: 1
タイプ: MQCFIN

KnownDestCount

説明: ローカル・キューへの送信が成功したメッセージの数
PCF パラメーター: MQIACF_KNOWN_DEST_COUNT
トレース・レベル: 1
タイプ: MQCFIN

UnknownDestCount

説明: リモート・キューへの送信が成功したメッセージの数
PCF パラメーター: MQIACF_UNKNOWN_DEST_COUNT
トレース・レベル: 1
タイプ: MQCFIN

InvalidDestCount

説明: 送信できなかったメッセージの数
PCF パラメーター: MQIACF_INVALID_DEST_COUNT
トレース・レベル: 1
タイプ: MQCFIN

HighResTime

説明: 1970 年 1 月 1 日深夜 0 時 (UTC) からのマイクロ秒で表す操作の時刻。
注: このタイマーの精度は、プラットフォームの高精度タイマー・サポートに従って変わります。
PCF パラメーター: MQIAMO64_HIGHRES_TIME
トレース・レベル: 2
タイプ: MQCFIN64

ObjectName

説明: オープンされたオブジェクトの名前。
PCF パラメーター: MQCACF_OBJECT_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_NAME_LENGTH

ResolvedQName

説明: キュー名の解決が実行された後のキューの名前。
PCF パラメーター: MQCACF_RESOLVED_Q_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_NAME_LENGTH

ResolvedQMgrName

説明: ネーム解決が実行された後の、キュー・マネージャー名。
PCF パラメーター: MQCACF_RESOLVED_Q_MGR
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_MGR_NAME_LENGTH

ResolvedLocalQName³

説明: ネーム解決が実行された後の、ローカル・キュー名を含みます。
PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME
トレース・レベル: 2
タイプ: MQCFST

ResolvedLocalQMgrName³

説明: ネーム解決が実行された後の、ローカル・キュー・マネージャー名を含みます。
PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_MGR
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_MGR_NAME_LENGTH

ReportOptions

説明: メッセージ・レポート・オプション
PCF パラメーター: MQIACF_REPORT
トレース・レベル: 2
タイプ: MQCFIN

MsgType

説明: メッセージのタイプ
PCF パラメーター: MQIACF_MSG_TYPE
トレース・レベル: 2
タイプ: MQCFIN

Expiry

説明: メッセージの存続時間。
PCF パラメーター: MQIACF_EXPIRY
トレース・レベル: 2
タイプ: MQCFIN

Format

説明: メッセージ・データの形式名。
PCF パラメーター: MQCACH_FORMAT_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_FORMAT_LENGTH

Priority

説明: メッセージ優先順位
PCF パラメーター: MQIACF_PRIORITY
トレース・レベル: 2
タイプ: MQCFIN

Persistence

説明: メッセージの持続性
PCF パラメーター: MQIACF_PERSISTENCE
トレース・レベル: 2
タイプ: MQCFIN

MsgId

説明: メッセージ ID
PCF パラメーター: MQBACF_MSG_ID
トレース・レベル: 2
タイプ: MQCFBS
長さ: MQ_MSG_ID_LENGTH

CorrelId

説明: 相関 ID
PCF パラメーター: MQBACF_CORREL_ID

トレース・レベル: 2
タイプ: MQCFBS
長さ: MQ_CORREL_ID_LENGTH

ReplyToQueue

説明:
PCF パラメーター: MQCACF_REPLY_TO_Q
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_NAME_LENGTH

ReplyToQMGr

説明:
PCF パラメーター: MQCACF_REPLY_TO_Q_MGR
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_MGR_NAME_LENGTH

CodedCharSetId

説明: メッセージ・データの文字セット ID
PCF パラメーター: MQIA_CODED_CHAR_SET_ID
トレース・レベル: 2
タイプ: MQCFIN

Encoding

説明: メッセージ・データの数値エンコード。
PCF パラメーター: MQIACF_ENCODING
トレース・レベル: 2
タイプ: MQCFIN

PutDate

説明:
PCF パラメーター: MQCACF_PUT_DATE
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_PUT_DATE_LENGTH

PutTime

説明:
PCF パラメーター: MQCACF_PUT_TIME
トレース・レベル: 2

タイプ: MQCFST
長さ: MQ_PUT_TIME_LENGTH

ResolvedQName

説明: ResolvedType が MQOT_Q の場合に、ObjectHandle によって参照されるキュー名。
PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_NAME_LENGTH。

ResObjectString

説明: ResolvedType が MQOT_TOPIC の場合に、ObjectHandle によって参照されるオブジェクト名。
PCF パラメーター: MQCACF_RESOLVED_OBJECT_STRING
トレース・レベル: 2
タイプ: MQCFST
長さ: 長さは可変です。

ResolvedType

説明: ObjectHandle によって参照されるオブジェクトのタイプ。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。
PCF パラメーター: MQIACF_RESOLVED_TYPE
トレース・レベル: 2
タイプ: MQCFIN

PolicyName

説明: このメッセージに適用されたポリシー名。
注: AMS で保護されたメッセージのみ
PCF パラメーター: MQCA_POLICY_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_OBJECT_NAME_LENGTH

XmitqMsgId

説明: 伝送キュー・ヘッダー内のメッセージのメッセージ ID。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ
PCF パラメーター: MQBACF_XQH_MSG_ID
トレース・レベル: 2
タイプ: MQCFBS
長さ: MQ_MSG_ID_LENGTH

XmitqCorrelId

説明: 伝送キュー・ヘッダー内のメッセージの相関 ID。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQBACF_XQH_CORREL_ID
トレース・レベル: 2
タイプ: MQCFBS
長さ: MQ_CORREL_ID_LENGTH

XmitqPutTime

説明: 伝送キュー・ヘッダー内のメッセージの書き込み時刻。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_PUT_TIME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_PUT_TIME_LENGTH

XmitqPutDate

説明: 伝送キュー・ヘッダー内のメッセージの書き込み日付。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_PUT_DATE
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

説明: 伝送キュー・ヘッダー内のメッセージのリモート・キュー宛先。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_REMOTE_Q_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_NAME_LENGTH

XmitqRemoteQMgr

説明: 伝送キュー・ヘッダー内のメッセージのリモート・キュー・マネージャー宛先。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_REMOTE_Q_MGR
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_NAME_LENGTH

PutMsgOptsStructure

説明:	MQPMO 構造体
PCF パラメーター:	MQBACF_MQPMO_STRUCTURE
トレース・レベル:	3
タイプ:	MQCFBS
長さ:	MQPMO 構造体のバイト単位の長さ (実際のサイズは構造体のバージョンによって異なる)

QMgrOpDuration

説明:	キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。 所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。 注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。
PCF パラメーター:	MQIAMO64_QMGR_OP_DURATION
トレース・レベル:	2
タイプ	MQCFIN64

MQPUT アプリケーション・アクティビティー配布リスト PCF グループ・ヘッダー構造

MQPUT 関数が配布リストに書き込まれる場合、MQPUT パラメーターには 1 つの AppActivityDistList PCF グループが含まれます。配布リスト内の各キューについては、[256 ページの『アプリケーション・アクティビティー配布リスト PCF グループのヘッダー構造』](#)を参照してください。AppActivityDistList PCF グループは、MQPMR 構造体と MQRR 構造体からの情報を結合して PUT パラメーターを識別し、各キュー上での PUT 操作の結果を示します。MQPUT 操作の場合、AppActivityDistList グループには以下のパラメーターの一部またはすべてが含まれます (理由コードが MQRC_MULTIPLE_REASONS の場合は CompCode と Reason が存在し、その他のパラメーターは MQPMO.PutMsgRecFields フィールドによって決まります)。

CompCode

説明:	操作の結果を示す完了コード。MQRR 構造体が提供されており、MQPUT の理由コードが MQRC_MULTIPLE_REASONS の場合のみ組み込まれます。
PCF パラメーター:	MQIACF_COMP_CODE
トレース・レベル:	2
タイプ:	MQCFIN

Reason

説明:	このオブジェクトの書き込みの結果を示す理由コード。MQRR 構造体が提供されており、MQPUT の理由コードが MQRC_MULTIPLE_REASONS の場合のみ組み込まれます。
PCF パラメーター:	MQIACF_REASON_CODE
トレース・レベル:	2
タイプ:	MQCFIN

³ ResolvedLocalQName パラメーターが含まれるのは、ResolvedQName パラメーターと異なる場合のみです。

MsgId

説明: メッセージ ID。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_MSG_ID が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQBACF_MSG_ID

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_MSG_ID_LENGTH

CorrelId

説明: 相関 ID。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_CORREL_ID が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQBACF_CORREL_ID

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_CORREL_ID_LENGTH

GroupId

説明: グループ ID。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_GROUP_ID が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQBACF_GROUP_ID

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_GROUP_ID_LENGTH

Feedback

説明: フィードバック。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_FEEDBACK が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQIACF_FEEDBACK

トレース・レベル: 2

タイプ: MQCFIN

AccountingToken

説明: AccountingToken。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_ACCOUNTING_TOKEN が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQBACF_ACCOUNTING_TOKEN

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_ACCOUNTING_TOKEN_LENGTH

MQPUT1

アプリケーションによって MQPUT1 MQI 機能が開始されました。

ObjectType

説明: MQOT.ObjectType で渡されるオブジェクト・タイプ

PCF パラメーター: MQIACF_OBJECT_TYPE
トレース・レベル: 1
タイプ: MQCFIN

ObjectName

説明: 何らかのキュー・ネーム・レゾリューションが試行される前に、MQI 呼び出しに渡されるオブジェクトの名前。
PCF パラメーター: MQCACF_OBJECT_NAME
トレース・レベル: 1
タイプ: MQCFST
長さ: MQ_Q_NAME_LENGTH

ObjectQMgrName

説明: 何らかのキュー・ネーム・レゾリューションが試行される前に、MQI 呼び出しに渡されるオブジェクト・キュー・マネージャーの名前。
PCF パラメーター: MQCACF_OBJECT_Q_MGR_NAME
トレース・レベル: 1
タイプ: MQCFST
長さ: MQ_Q_MGR_NAME_LENGTH

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード
PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ: MQCFIN

PutOptions

説明: MQPMO.Options の書き込みオプション
PCF パラメーター: MQIACF_PUT_OPTIONS
トレース・レベル: 1
タイプ: MQCFIN

AlternateUserId

説明: MQPMO_ALTERNATE_USER_AUTHORITY が指定されている場合だけ含まれません。
PCF パラメーター: MQCACF_ALTERNATE_USERID

トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_USER_ID_LENGTH

RecsPresent

説明: 存在しているオブジェクト名レコードの数
PCF パラメーター: MQIACF_RECS_PRESENT
トレース・レベル: 1
タイプ: MQCFIN

KnownDestCount

説明: オープンに成功したローカル・キューの数
PCF パラメーター: MQIACF_KNOWN_DEST_COUNT
トレース・レベル: 1
タイプ: MQCFIN

UnknownDestCount

説明: オープンに成功したリモート・キューの数
PCF パラメーター: MQIACF_UNKNOWN_DEST_COUNT
トレース・レベル: 1
タイプ: MQCFIN

InvalidDestCount

説明: オープンに失敗したキューの数
PCF パラメーター: MQIACF_INVALID_DEST_COUNT
トレース・レベル: 1
タイプ: MQCFIN

MsgBuffer

説明: メッセージ・データ。
PCF パラメーター: MQBACF_MESSAGE_DATA
トレース・レベル: 1
タイプ: MQCFBS
長さ: 長さは、APPTTRACE 構成で設定された TRACEDATA() パラメーターによって制御されます。TRACEDATA=NONE の場合は、このパラメーターは省略されます。

MsgLength

説明: メッセージの長さ。
PCF パラメーター: MQIACF_MSG_LENGTH
トレース・レベル: 1
タイプ: MQCFIN

HighResTime

説明: 1970年1月1日深夜0時(UTC)からのマイクロ秒で表す操作の時刻。
注: このタイマーの精度は、プラットフォームの高精度タイマー・サポートに従って変わります。

PCF パラメーター: MQIAMO64_HIGHRES_TIME

トレース・レベル: 2

タイプ: MQCFIN64

ResolvedQName

説明: キュー名の解決が実行された後のキューの名前。

PCF パラメーター: MQCACF_RESOLVED_Q_NAME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH

ResolvedQMgrName

説明: ネーム解決が実行された後の、キュー・マネージャー名。

PCF パラメーター: MQCACF_RESOLVED_Q_MGR

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_MGR_NAME_LENGTH

ResolvedLocalQName⁴

説明: ネーム解決が実行された後の、ローカル・キュー名を含みます。

PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME

トレース・レベル: 2

タイプ: MQCFST

ResolvedLocalQMgrName⁴

説明: ネーム解決が実行された後の、ローカル・キュー・マネージャー名を含みます。

PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_MGR

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_MGR_NAME_LENGTH

AlternateSecurityId

説明: 代替セキュリティ ID。MQOD.Version が MQOD_VERSION_3 以上で、MQOD.AlternateSecurityId が MQSID_NONE に等しくない場合だけ、現れます。

PCF パラメーター: MQBACF_ALTERNATE_SECURITYID

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_SECURITY_ID_LENGTH

ObjectString

説明: 長いオブジェクト名。MQOD.Version が MQOD_VERSION_4 以上で、MQOD.ObjectString の VLength フィールドが MQVS_NULL_TERMINATED かゼロより大きい場合にだけ含まれます。

PCF パラメーター: MQCACF_OBJECT_STRING

トレース・レベル: 2

タイプ: MQCFST

長さ: 長さは可変です。

ResObjectString

説明: ObjectName フィールドで提供されている名前をキュー・マネージャーが解決した後の、長いオブジェクト名です。MQOD.Version が MQOD_VERSION_4 以上で、VLength が MQVS_NULL_TERMINATED かゼロより大きい場合にだけ、トピック・オブジェクトを参照するトピックとキュー別名として含まれます。

PCF パラメーター: MQCACF_RESOLVED_OBJECT_STRING

トレース・レベル: 2

タイプ: MQCFST

長さ: 長さは可変です。

ResolvedType

説明: 開いている解決済み (ベース) オブジェクトのタイプ。MQOD.Version が MQOD_VERSION_4 以上の場合にだけ含まれます。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。

PCF パラメーター: MQIACF_RESOLVED_TYPE

トレース・レベル: 2

タイプ: MQCFIN

ReportOptions

説明: メッセージ・レポート・オプション

PCF パラメーター: MQIACF_REPORT

トレース・レベル: 2

タイプ: MQCFIN

MsgType

説明: メッセージのタイプ

PCF パラメーター: MQIACF_MSG_TYPE

トレース・レベル: 2

タイプ: MQCFIN

Expiry

説明: メッセージの存続時間。

PCF パラメーター: MQIACF_EXPIRY

トレース・レベル: 2
タイプ: MQCFIN

Format

説明: メッセージ・データの形式名。
PCF パラメーター: MQCACH_FORMAT_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_FORMAT_LENGTH

Priority

説明: メッセージ優先順位
PCF パラメーター: MQIACF_PRIORITY
トレース・レベル: 2
タイプ: MQCFIN

Persistence

説明: メッセージの持続性
PCF パラメーター: MQIACF_PERSISTENCE
トレース・レベル: 2
タイプ: MQCFIN

MsgId

説明: メッセージ ID
PCF パラメーター: MQBACF_MSG_ID
トレース・レベル: 2
タイプ: MQCFBS
長さ: MQ_MSG_ID_LENGTH

CorrelId

PCF パラメーター: 相関 ID
説明: MQBACF_CORREL_ID
トレース・レベル: 2
タイプ: MQCFBS
長さ: MQ_CORREL_ID_LENGTH

ReplyToQueue

説明:
PCF パラメーター: MQCACF_REPLY_TO_Q
トレース・レベル: 2
タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH

ReplyToQMgr

説明:

PCF パラメーター: MQCACF_REPLY_TO_Q_MGR

トレース・レベル: 2

タイプ: MQCFST

長さ: MQCFST

CodedCharSetId

説明: メッセージ・データの文字セット ID

PCF パラメーター: MQIA_CODED_CHAR_SET_ID

トレース・レベル: 2

タイプ: MQCFIN

Encoding

説明: メッセージ・データの数値エンコード。

PCF パラメーター: MQIACF_ENCODING

トレース・レベル: 2

タイプ: MQCFIN

PutDate

説明:

PCF パラメーター: MQCACF_PUT_DATE

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_PUT_DATE_LENGTH

PutTime

説明:

PCF パラメーター: MQCACF_PUT_TIME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_PUT_TIME_LENGTH

PolicyName

説明: このメッセージに適用されたポリシー名。

注: AMS で保護されたメッセージのみ

PCF パラメーター: MQCA_POLICY_NAME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_OBJECT_NAME_LENGTH

XmitqMsgId

説明: 伝送キュー・ヘッダー内のメッセージのメッセージ ID。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQBACF_XQH_MSG_ID

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_MSG_ID_LENGTH

XmitqCorrelId

説明: 伝送キュー・ヘッダー内のメッセージの相関 ID。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQBACF_XQH_CORREL_ID

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_CORREL_ID_LENGTH

XmitqPutTime

説明: 伝送キュー・ヘッダー内のメッセージの書き込み時刻。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_PUT_TIME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_PUT_TIME_LENGTH

XmitqPutDate

説明: 伝送キュー・ヘッダー内のメッセージの書き込み日付。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_PUT_DATE

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_PUT_DATE_LENGTH

XmitqRemoteQName

説明: 伝送キュー・ヘッダー内のメッセージのリモート・キュー宛先。

注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_REMOTE_Q_NAME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH

XmitqRemoteQMGr

説明: 伝送キュー・ヘッダー内のメッセージのリモート・キュー・マネージャー宛先。
注: Format が MQFMT_XMIT_Q_HEADER の場合のみ

PCF パラメーター: MQCACF_XQH_REMOTE_Q_MGR

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH

PutMsgOptsStructure

説明: MQPMO 構造体

PCF パラメーター: MQBACF_MQPMO_STRUCT

トレース・レベル: 3

タイプ: MQCFBS

長さ: MQPMO 構造体のバイト単位の長さ (実際のサイズは構造体のバージョンによって異なる)

QMGrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。

所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。

注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION

トレース・レベル: 2

タイプ: MQCFIN64

MQPUT1 AppActivityDistList PCF グループ・ヘッダー構造

MQPUT1 関数が配布リストに書き込まれる場合、variable パラメーターには 1 つの AppActivityDistList PCF グループが含まれます。配布リスト内の各キューについては、[256 ページの『アプリケーション・アクティビティ配布リスト PCF グループのヘッダー構造』](#)を参照してください。AppActivityDistList PCF グループは、MQOR、MQPMR、および MQRR の各構造体からの情報を結合してオブジェクトと PUT パラメーターを識別し、各キュー上での PUT 操作の結果を示します。MQPUT1 操作の場合、AppActivityDistList グループには以下のパラメーターの一部またはすべてが含まれています (理由コードが MQRC_MULTIPLE_REASONS の場合は CompCode、Reason、ObjectName、および ObjectQMGrName が存在し、その他のパラメーターは MQPMO.PutMsgRecFields フィールドによって決まります)。

CompCode

説明: このオブジェクトの書き込みの結果を示す完了コード。MQRR 構造体が提供されており、MQPUT1 の理由コードが MQRC_MULTIPLE_REASONS の場合のみ組み込まれます。

⁴ ResolvedLocalQName パラメーターが含まれるのは、ResolvedQName パラメーターと異なる場合のみです。

PCF パラメーター: MQIACF_COMP_CODE

トレース・レベル: 2

タイプ: MQCFIN

Reason

説明: このオブジェクトの書き込みの結果を示す理由コード。MQRR 構造体が提供されており、MQPUT1 の理由コードが MQRC_MULTIPLE_REASONS の場合のみ組み込まれます。

PCF パラメーター: MQIACF_REASON_CODE

トレース・レベル: 2

タイプ: MQCFIN

ObjectName

説明: 配布リスト内のキューの名前。MQOR 構造体が提供されている場合のみ組み込まれます。

PCF パラメーター: MQCACF_OBJECT_NAME

トレース・レベル: 2

タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH

MsgId

説明: メッセージ ID。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_MSG_ID が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQBACF_MSG_ID

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_MSG_ID_LENGTH

CorrelId

説明: 相関 ID。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_CORREL_ID が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQBACF_CORREL_ID

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_CORREL_ID_LENGTH

GroupId

説明: グループ ID。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_GROUP_ID が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQBACF_GROUP_ID

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_GROUP_ID_LENGTH

Feedback

説明: フィードバック。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_FEEDBACK が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQIACF_FEEDBACK

トレース・レベル: 2

タイプ: MQCFIN

AccountingToken

説明: AccountingToken. MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF_ACCOUNTING_TOKEN が含まれている場合にのみ組み込まれます。

PCF パラメーター: MQBACF_ACCOUNTING_TOKEN

トレース・レベル: 2

タイプ: MQCFBS

長さ: MQ_ACCOUNTING_TOKEN_LENGTH

MQSET

アプリケーションが MQSET MQI 関数を開始しています

ObjectHandle

説明: オブジェクト・ハンドル

PCF パラメーター: MQIACF_HOBJ

トレース・レベル: 1

タイプ: MQCFIN

CompCode

説明: 操作の結果を示す完了コード

PCF パラメーター: MQIACF_COMP_CODE

トレース・レベル: 1

タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード

PCF パラメーター: MQIACF_REASON_CODE

トレース・レベル: 1

タイプ: MQCFIN

SelectorCount

説明: Selectors 配列で提供されるセレクターのカウンント。

PCF パラメーター: MQIACF_SELECTOR_COUNT

トレース・レベル: 2

タイプ: MQCFIN

Selectors

説明: 値が MQSET によって更新される属性のリスト (整数または文字)。
PCF パラメーター: MQIACF_SELECTORS
トレース・レベル: 2
タイプ: MQCFIL

ResolvedQName

説明: ResolvedType が MQOT_Q の場合に、ObjectHandle によって参照されるキュー名。
PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_NAME_LENGTH。

ResObjectString

説明: ResolvedType が MQOT_TOPIC の場合に、ObjectHandle によって参照されるオブジェクト名。
PCF パラメーター: MQCACF_RESOLVED_OBJECT_STRING
トレース・レベル: 2
タイプ: MQCFST
長さ: 長さは可変です。

ResolvedType

説明: ObjectHandle によって参照されるオブジェクトのタイプ。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。
PCF パラメーター: MQIACF_RESOLVED_TYPE
トレース・レベル: 2
タイプ: MQCFIN

IntAttrCount

説明: 設定操作によって更新される整数属性の数。
PCF パラメーター: MQIACF_INTATTR_COUNT
トレース・レベル: 3
タイプ: MQCFIN

IntAttrs

説明: 整数属性の値。
PCF パラメーター: MQIACF_INT_ATTRS
トレース・レベル: 3
タイプ: MQCFIL
範囲: このパラメーターは、IntAttrCount > 0 である場合にのみ存在します。

CharAttrs

説明: 設定操作によって更新される文字属性。値は一緒に連結されます。
PCF パラメーター: MQCACF_CHAR_ATTRS
トレース・レベル: 3
タイプ: MQCFST
範囲: このパラメーターは、CharAttrLength > 0 である場合にのみ存在します。

QMgrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION
トレース・レベル: 2
タイプ: MQCFIN64

MQSUB

アプリケーションによって MQSUB MQI 機能が開始されました。

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード
PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ: MQCFIN

SubHandle

説明: サブスクリプション・ハンドル
PCF パラメーター: MQIACF_HSUB
トレース・レベル: 1
タイプ: MQCFIN

ObjectHandle

説明: オブジェクト・ハンドル
PCF パラメーター: MQIACF_HOBJ
トレース・レベル: 1

タイプ: MQCFIN

Options

説明: サブスクリプション・オプション

PCF パラメーター: MQIACF_SUB_OPTIONS

トレース・レベル: 1

タイプ: MQCFIN

ObjectName

説明: オブジェクトの名前。

PCF パラメーター: MQCACF_OBJECT_NAME

トレース・レベル: 1

タイプ: MQCFST

長さ: MQ_Q_NAME_LENGTH

ObjectString

説明: 長いオブジェクト名。

PCF パラメーター: MQCACF_OBJECT_STRING

トレース・レベル: 1

タイプ: MQCFST

範囲: MQSD.ObjectString の VSLength フィールドがゼロより大きいか、または MQVS_NULL_TERMINATED である場合にだけ含まれます。

長さ: 長さは可変です。

AlternateUserId

説明:

PCF パラメーター: MQCACF_ALTERNATE_USERID

トレース・レベル: 2

タイプ: MQCFST

範囲: MQSO_ALTERNATE_USER_AUTHORITY が指定されている場合だけ含まれます。

長さ: MQ_USER_ID_LENGTH

AlternateSecurityId

説明: 代替セキュリティー ID。

PCF パラメーター: MQBACF_ALTERNATE_SECURITYID

トレース・レベル: 2

タイプ: MQCFBS

範囲: MQSO_ALTERNATE_USER_AUTHORITY が指定され、MQSD.AlternateSecurityId が MQSID_NONE に等しくない場合にだけ現れます。

長さ: MQ_SECURITY_ID_LENGTH

SubName

説明: サブスクリプション名
PCF パラメーター: MQCACF_SUB_NAME
トレース・レベル: 2
タイプ: MQCFST
範囲: MQSD.SubName の VSLength フィールドがゼロより大きいか、または MQVS_NULL_TERMINATED である場合にだけ含まれます。
長さ: 長さは可変です。

SubUserData

説明: サブスクリプション・ユーザー・データ
PCF パラメーター: MQCACF_SUB_USER_DATA
トレース・レベル: 2
タイプ: MQCFST
範囲: MQSD.SubName の VSLength フィールドがゼロより大きいか、または MQVS_NULL_TERMINATED である場合にだけ含まれます。
長さ: 長さは可変です。

SubCorrelId

説明: サブスクリプション相関 ID
PCF パラメーター: MQBACF_SUB_CORREL_ID
トレース・レベル: 2
タイプ: MQCFBS
長さ: MQ_CORREL_ID_LENGTH

SelectionString

説明: 選択ストリング。
PCF パラメーター: MQCACF_SELECTION_STRING
トレース・レベル: 2
タイプ: MQCFST
範囲: MQSD の VSLength フィールドがある場合にのみ含まれます。 SelectionString が MQVS_NULL_TERMINATED またはゼロより大きい値になっています。
長さ: 長さは可変です。

ResolvedQName

説明: ResolvedType が MQOT_Q の場合に、ObjectHandle によって参照されるキュー名。
PCF パラメーター: MQCACF_RESOLVED_LOCAL_Q_NAME
トレース・レベル: 2
タイプ: MQCFST
長さ: MQ_Q_NAME_LENGTH。

ResObjectString

説明: ResolvedType が MQOT_TOPIC の場合に、ObjectHandle によって参照されるオブジェクト名。

PCF パラメーター: MQCACF_RESOLVED_OBJECT_STRING

トレース・レベル: 2

タイプ: MQCFST

長さ: 長さは可変です。

ResolvedType

説明: ObjectHandle によって参照されるオブジェクトのタイプ。可能な値は、MQOT_Q、MQOT_TOPIC、または MQOT_NONE です。

PCF パラメーター: MQIACF_RESOLVED_TYPE

トレース・レベル: 2

タイプ: MQCFIN

SubDescriptorStructure

説明: MQSD 構造体。

PCF パラメーター: MQBACF_MQSD_STRUCT

トレース・レベル: 3

タイプ: MQCFBS

長さ: MQSD 構造体のバイト単位の長さです。

QMgrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。

所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。

注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION

トレース・レベル: 2

タイプ: MQCFIN64

MQSUBRQ

アプリケーションによって MQSUBRQ MQI 機能が開始されました。

CompCode

説明: 操作の結果を示す完了コード

PCF パラメーター: MQIACF_COMP_CODE

トレース・レベル: 1

タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード

PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ: MQCFIN

SubHandle

説明: サブスクリプション・ハンドル
PCF パラメーター: MQIACF_HSUB
トレース・レベル: 1
タイプ: MQCFIN

SubOptions

説明: MQSB.Options のサブオプション
PCF パラメーター: MQIACF_SUBRQ_OPTIONS
トレース・レベル: 2
タイプ: MQCFIN

Action

説明: サブスクリプション要求アクション (MQSR_*)
PCF パラメーター: MQIACF_SUBRQ_ACTION
トレース・レベル: 2
タイプ: MQCFIN

NumPubs

説明: (MQSB.NumPubs からの) この呼び出しの結果として送信されたパブリケーションの数
PCF パラメーター: MQIACF_NUM_PUBS
トレース・レベル: 2
タイプ: MQCFIN

QMgrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。
PCF パラメーター: MQIAMO64_QMGR_OP_DURATION
トレース・レベル: 2
タイプ: MQCFIN64

MQSTAT

アプリケーションが MQSTAT MQI 関数を開始しています

CompCode

説明: 操作の結果を示す完了コード
PCF パラメーター: MQIACF_COMP_CODE
トレース・レベル: 1
タイプ: MQCFIN

Reason

説明: 操作の結果を示す理由コード
PCF パラメーター: MQIACF_REASON_CODE
トレース・レベル: 1
タイプ: MQCFIN

Type

説明: 要求された状況情報のタイプ
PCF パラメーター: MQIACF_STATUS_TYPE
トレース・レベル: 2
タイプ: MQCFIN

StatusStructure

説明: MQSTS 構造体
PCF パラメーター: MQBACF_MQSTS_STRUCT
トレース・レベル: 3
タイプ: MQCFBS
長さ: MQSTS 構造体のバイト単位の長さ (実際のサイズは構造体のバージョンに応じて異なります)

QMgrOpDuration

説明: キュー・マネージャー内でのおよその API 呼び出し所要時間 (マイクロ秒単位)。
所要時間には、キュー・マネージャー外部で経過した時間は含まれません。例えば、IBM MQ クライアントとして経過した時間などが該当します。
注: このタイマーの精度は、社内で使用しているプラットフォームによって異なります。

PCF パラメーター: MQIAMO64_QMGR_OP_DURATION
トレース・レベル: 2
タイプ: MQCFIN64

アプリケーション・アクティビティ XA 操作の変数パラメーター

XA 操作は、MQ がトランザクションに参加できるようにするためにアプリケーションが作成できる API 呼び出しです。各操作のパラメーターは、次のセクションで定義されています。

トレース・レベルは、トレースに含まれるパラメーターで必要なトレース細分度のレベルを示します。指定できるトレース・レベル値は以下のとおりです。

1. 低

このパラメータは、アプリケーションに"低"、"中"、"高"のアクティビティ・トレースが設定されている場合に含まれます。この設定値は、パラメーターが常に操作の AppActivityData グループに含まれることを意味します。このパラメーターのセットは、アプリケーションが行う MQI 呼び出しをトレースし、それらが成功したかどうかを確認できるものです。

2. 中

このパラメーターは、アプリケーションに対して "medium" または "high" アクティビティ・トレースが構成されている場合にのみ、操作の AppActivityData グループに組み込まれます。このパラメーターのセットは、アプリケーションにより使用されるリソース (例えば、キューとトピック名) に関する情報を追加します。

3. 高

このパラメーターは、アプリケーションに対して "高" アクティビティ・トレースが構成されている場合にのみ、操作の AppActivityData グループに組み込まれます。このパラメーターのセットは、MQI と XA 関数に渡された構造体のメモリー・ダンプを含みます。この理由で、これは MQI と XA 呼び出しで使用されるパラメーターに関するさらに詳細な情報を含みます。構造体メモリー・ダンプは、構造体のシャロー・コピーです。ポインターの間接参照を誤って試行しないようにするため、構造体中のポインター値は NULL に設定されます。

注: ダンプされる構造体のバージョンは、アプリケーションにより使用されるバージョンと同じである必要はありません。構造体は、API 交差出口、アクティビティ・トレース・コード、またはキュー・マネージャーで変更できます。キュー・マネージャーは、構造体をより最近のバージョンに変更することはできますが、構造体の以前のバージョンに変更することはできません。これを行うと、データを失うおそれがあります。

AXREG

アプリケーションによって AXREG AX 機能が開始されました。

XID

説明: XID 構造体
PCF パラメーター: MQBACF_XA_XID
トレース・レベル: 1
タイプ: MQCFBS
長さ: Sizeof(XID)

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明: フラグ
PCF パラメーター: MQIACF_XA_FLAGS
トレース・レベル: 1
タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE

トレース・レベル: 1
タイプ: MQCFIN

AXUNREG

アプリケーションによって AXUNREG AX 機能が開始されました。

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明: フラグ
PCF パラメーター: MQIACF_XA_FLAGS
トレース・レベル: 1
タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE
トレース・レベル: 1
タイプ: MQCFIN

XACLOSE

アプリケーションによって XACLOSE AX 機能が開始されました。

Xa_info

説明: リソース・マネージャーの初期化に使用される情報。
PCF パラメーター: MQCACF_XA_INFO
トレース・レベル: 1
タイプ: MQCFST

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明: フラグ
PCF パラメーター: MQIACF_XA_FLAGS
トレース・レベル: 1
タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE
トレース・レベル: 1
タイプ: MQCFIN

XACOMMIT

アプリケーションによって XACOMMIT AX 機能が開始されました。

XID

説明: XID 構造体
PCF パラメーター: MQBACF_XA_XID
トレース・レベル: 1
タイプ: MQCFBS
長さ: Sizeof(XID)

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明: フラグ
PCF パラメーター: MQIACF_XA_FLAGS
トレース・レベル: 1
タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE
トレース・レベル: 1
タイプ: MQCFIN

XACOMplete

アプリケーションによって XACOMplete AX 機能が開始されました。

Handle

説明: 非同期操作のためのハンドル
PCF パラメーター: MQIACF_XA_HANDLE
トレース・レベル: 1
タイプ: MQCFIN

Retval

説明: 非同期関数の戻り値
PCF パラメーター: MQIACF_XA_RETVAL
トレース・レベル: 1
タイプ: MQCFINMQCFBS

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明: フラグ
PCF パラメーター: MQIACF_XA_FLAGS
トレース・レベル: 1
タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE
トレース・レベル: 1
タイプ: MQCFIN

XAEND

アプリケーションによって XAEND AX 機能が開始されました。

XID

説明: XID 構造体
PCF パラメーター: MQBACF_XA_XID
トレース・レベル: 1
タイプ: MQCFBS
長さ: Sizeof(XID)

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明: フラグ
PCF パラメーター: MQIACF_XA_FLAGS

トレース・レベル: 1
タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE
トレース・レベル: 1
タイプ: MQCFIN

XAFORGET

アプリケーションによって AXREG AX 機能が開始されました。

XID

説明: XID 構造体
PCF パラメーター: MQBACF_XA_XID
トレース・レベル: 1
タイプ: MQCFBS
長さ: Sizeof(XID)

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明: フラグ
PCF パラメーター: MQIACF_XA_FLAGS
トレース・レベル: 1
タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE
トレース・レベル: 1
タイプ: MQCFIN

XAOPEN

アプリケーションによって XAOPEN AX 機能が開始されました。

Xa_info

説明: リソース・マネージャーの初期化に使用される情報。
PCF パラメーター: MQCACF_XA_INFO
トレース・レベル: 1

タイプ: MQCFST

Rmid

説明: リソース・マネージャー ID

PCF パラメーター: MQIACF_XA_RMID

トレース・レベル: 1

タイプ: MQCFIN

Flags

説明: フラグ

PCF パラメーター: MQIACF_XA_FLAGS

トレース・レベル: 1

タイプ: MQCFIN

XARetCode

説明: 戻りコード

PCF パラメーター: MQIACF_XA_RETCODE

トレース・レベル: 1

タイプ: MQCFIN

XAPREPARE

アプリケーションによって XAPREPARE AX 機能が開始されました。

XID

説明: XID 構造体

PCF パラメーター: MQBACF_XA_XID

トレース・レベル: 1

タイプ: MQCFBS

長さ: Sizeof(XID)

Rmid

説明: リソース・マネージャー ID

PCF パラメーター: MQIACF_XA_RMID

トレース・レベル: 1

タイプ: MQCFIN

Flags

説明: フラグ

PCF パラメーター: MQIACF_XA_FLAGS

トレース・レベル: 1

タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE
トレース・レベル: 1
タイプ: MQCFIN

XARECOVER

アプリケーションによって XARECOVER AX 機能が開始されました。

Count

説明: XID のカウント
PCF パラメーター: MQIACF_XA_COUNT
トレース・レベル: 1
タイプ: MQCFIN

XIDs

説明: XID 構造体
注: この PCF パラメーターには複数のインスタンス (XID 構造体ごとに 1 つから
カウント XID まで) があります。
PCF パラメーター: MQBACF_XA_XID
トレース・レベル: 1
タイプ: MQCFBS
長さ: Sizeof(XID)

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明: フラグ
PCF パラメーター: MQIACF_XA_FLAGS
トレース・レベル: 1
タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE
トレース・レベル: 1
タイプ: MQCFIN

XAROLLBACK

アプリケーションによって XAROLLBACK AX 機能が開始されました。

XID

説明: XID 構造体
PCF パラメーター: MQBACF_XA_XID
トレース・レベル: 1
タイプ: MQCFBS
長さ: Sizeof(XID)

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明: フラグ
PCF パラメーター: MQIACF_XA_FLAGS
トレース・レベル: 1
タイプ: MQCFIN

XARetCode

説明: 戻りコード
PCF パラメーター: MQIACF_XA_RETCODE
トレース・レベル: 1
タイプ: MQCFIN

XASTART

アプリケーションによって XASTART AX 機能が開始されました。

XID

説明: XID 構造体
PCF パラメーター: MQBACF_XA_XID
トレース・レベル: 1
タイプ: MQCFBS
長さ: Sizeof(XID)

Rmid

説明: リソース・マネージャー ID
PCF パラメーター: MQIACF_XA_RMID
トレース・レベル: 1
タイプ: MQCFIN

Flags

説明:	フラグ
PCF パラメーター:	MQIACF_XA_FLAGS
トレース・レベル:	1
タイプ:	MQCFIN

XARetCode

説明:	戻りコード
PCF パラメーター:	MQIACF_XA_RETCODE
トレース・レベル:	1
タイプ:	MQCFIN

Multi

モニターとアクティビティ・トレースに関するシステム・トピック

キュー・マネージャー・トピック・ツリー内のシステム・トピックは、リソース・モニター (一部は統計メッセージの内容に似ています)、およびアプリケーション・アクティビティ・トレースのコンシューム方法として使用されます。

キュー・マネージャー・トピック・ツリーの \$SYS/MQ ブランチ

各キュー・マネージャーのトピック・ツリーには、\$SYS/MQ ブランチが含まれています。キュー・マネージャーはこのブランチのトピック・ストリングにパブリッシュします。許可されたユーザーは、これらのトピック・ストリングにサブスクライブして、キュー・マネージャーおよびそれに対するアクティビティに関する情報を受け取ることができます。これらのシステム・トピックは、アプリケーション・アクティビティのトレースとモニターに使用されます。詳しくは、[トピック・ツリー](#)を参照してください。

\$SYS/MQ ブランチのルートは、SYSTEM.ADMIN.TOPIC トピック・オブジェクトによって表されます。トピック・ツリーの \$SYS/MQ ブランチは、以下の方法で、残りのトピック・ツリーから切り分けられます。

- ツリー内の \$SYS/MQ より上位の位置でワイルドカード文字が使用されているサブスクリプションは、\$SYS/MQ ブランチ内のどのトピック・ストリングにも一致しません。SYSTEM.ADMIN.TOPIC のワイルドカード操作は "ブロック" に設定されており、変更できません。この制限は、トピック・ツリーのノードを表示するために、ワイルドカード文字を **runmqsc** コマンドの DISPLAY TPSTATUS で使用する場合にも適用されます。\$SYS/MQ ブランチ内のトピック・ノードを表示するには、トピック・ストリングを \$SYS/MQ で開始します。例えば、\$SYS/MQ/# を使用して、すべてのノードを表示します。
- \$SYS/MQ 以下で許可されていて、\$SYS/MQ トピック・ツリーを使用する権限が付与されている必要があります。トピック・ストリングにサブスクライブする許可は、トピック・ツリーのトピック・ストリング以上にある管理対象トピック・オブジェクトに対して付与されている許可に基づきます。ルート (SYSTEM.BASE.TOPIC) で付与されている許可は、ユーザー権限をすべてのトピック・ストリングに付与します。ただし、\$SYS/MQ ブランチの場合、\$SYS/MQ より上位に付与されたアクセス権限は \$SYS/MQ トピック・ストリングに適用されません。
- トピック・ツリーの \$SYS/MQ ブランチは、ツリーの上位に設定されたトピック属性から切り分けられません。SYSTEM.ADMIN.TOPIC は、トピック・ツリーの上位で定義されたトピック・オブジェクトから属性を継承しません。例えば、SYSTEM.BASE.TOPIC の属性を変更しても、\$SYS/MQ ブランチの動作に影響しません。

\$SYS/MQ で始まるすべてのトピック・ストリングは、IBM MQ による使用のために予約されています。これらのトピック・ストリングの制約事項は以下のとおりです。

- トピック・ツリーの \$SYS/MQ ブランチからマルチキャストを有効にすることはできません。
- クラスタ化は \$SYS/MQ ブランチではサポートされません。
- プロキシ・サブスクリプション・メカニズムを "強制" に設定できません。
- アプリケーションは \$SYS/MQ トピック・ストリングにパブリッシュできません。

- パブリケーションとサブスクリプションの有効範囲は、デフォルトでローカル・キュー・マネージャーにのみ設定されます。
- 以下の位置で使用できるワイルドカード文字はありません。
 - \$SYS/MQ/
 - \$SYS/MQ/INFO
 - \$SYS/MQ/INFO/QMGR
 - \$SYS/MQ/INFO/QMGR/queue_manager_name
 - \$SYS/MQ/INFO/QMGR/queue_manager_name/ActivityTrace

これらの位置でワイルドカード文字を使用しようとすると、理由 MQRC_ADMIN_TOPIC_STRING_ERROR でサブスクリプション障害が発生します。

パブリッシュ/サブスクライブで使用できるパフォーマンス・モニター API

単純なパブリッシュ/サブスクライブ・メカニズムと、amqsrua サンプル・アプリケーションまたは独自のアプリケーションのいずれかを使用して、キュー・マネージャー統計をモニターできます。ユーザーがリソースをモニターするために役立つ統計は、\$SYS/MQ/INFO/QMGR の下のシステム・トピックにパブリッシュされます。これらの統計は、amqsrua サンプル・アプリケーションを実行するか、または **amqsrua** と同様の方法でリソース・モニター・システム・トピックにサブスクライブするアプリケーションを作成することによって表示できます。

関連概念

204 ページの『アプリケーション・アクティビティー・トレース』

アプリケーション・アクティビティー・トレースは、キュー・マネージャーに接続されたアプリケーションの動作に関する詳細な情報を生成します。これは、IBM MQ リソースと相互作用する際に、アプリケーションの動作をトレースし、アプリケーションによって使用されるパラメーターの詳細なビューを提供します。さらに、アプリケーションによって発行される MQI 呼び出しの順序も表示します。

Windows Linux amqsrua コマンドを使用したシステム・リソース使用量のモニター

amqsrua コマンドを使用して、キュー・マネージャーのシステム・リソース使用量に関連したパフォーマンス・データを照会できます。

このタスクについて

amqsrua サンプル・アプリケーションによって、IBM MQ のモニター・パブリケーションをコンシュームしたり、キュー・マネージャーからパブリッシュされるパフォーマンス・データを表示したりする方法を確認できます。このデータには、CPU、メモリー、およびディスクの使用量に関する情報を含めることができます。また、STATMQI PCF 統計データに相当するデータを見ることもできます。このデータは 10 秒ごとにパブリッシュされ、コマンドの実行中に報告されます。

キュー・マネージャー名のみを指定してこのコマンドを実行して、対話式にステップスルーし、各ステップでキュー・マネージャーで使用可能な **CLASS** パラメーター、**TYPE** パラメーター、および **object** パラメーターを選択することができます。情報を表示する **CLASS**、**TYPE**、**object** の名前を把握している場合は、**amqsrua** コマンドの実行時にそれらを指定できます。

デフォルトでは、amqsrua アプリケーションはトピック・ツリー \$SYS/MQ/INFO/QMGR の下のキュー・マネージャーによってパブリッシュされた統計を検索します。その他のコンポーネントまたはアプリケーションは、同様のメカニズムを使用して別のトピック開始点の下にパブリッシュできます。IBM MQ 9.1.0 以降では、**-p** パラメーターを使用して、amqsrua が Linux と Windows の両方でこれらの他のコンポーネントの統計を検索する場所を指定できます。

-m

キュー・マネージャー名。キュー・マネージャーが実行中でなければなりません。キュー・マネージャー名を指定しない場合は、デフォルトのキュー・マネージャーの統計が表示されます。

-c

クラス名。IBM MQ のリソース使用状況パブリケーションにはクラスが関連付けられています。それらのクラスが、使用できるリソース使用情報を表すメタデータ・ツリーの最上位に相当します。

CPU

CPU 使用量に関する情報を返します。

DISK

ディスク使用量に関する情報を返します。

STATMQI

MQI 使用量に関する情報を返します。

STATQ

キュー別の MQI 使用量に関する情報を返します。

STATAPP

指定されたアプリケーションの使用統計に関する情報を返します。

メタデータのトピック・ツリーを指定する方法については、295 ページの『独自のリソース・モニター・アプリケーションの開発』、アプリケーションの名前に使用できる文字については、サポートされているプログラミング言語でのアプリケーション名の使用をそれぞれ参照してください。

注: STATAPP のモニター対象のアプリケーション名と STATQ のキュー名を指定するには、**-o** オプションを使用します。

-t

タイプ名。クラス内では、IBM MQ のリソース使用状況パブリケーションにタイプが関連付けられています。各パブリケーションにはクラスとタイプが含まれており、これにより `class/type/element` の定義が見つかり、結果としてパブリケーションが処理されます。`class/type/element` 記述は、キュー・マネージャーの始動時にメタデータとしてパブリッシュされます。

-o

オブジェクト名 リソース使用状況パブリケーションは、一連の PCF エlement で構成された PCF メッセージです。クラス/タイプのペアごとにパブリッシュされる PCF エlement が、メタデータに入れて通知されます。各エlement を記述したリーフがツリーに保管されるので、エlement の処理が可能になります。

-p

メタデータ接頭部。キュー・マネージャーによってパブリッシュされた統計を `amqsrua` で検索するトピック・ツリー開始点を指定します。デフォルトのトピック・ツリーは `$SYS/MQ/INFO/QMGR` ですが、他のコンポーネントまたはアプリケーションは、別のトピック・ツリー開始点の下に統計をパブリッシュする可能性があります。

-n

パブリケーション・カウント。コマンド終了までに返すレポートの数を指定できます。データは約 10 秒ごとにパブリッシュされるので、50 という値を入力すると、このコマンドは 500 秒にわたって 50 個のレポートを返します。このパラメーターを指定しない場合、エラーが発生するかキュー・マネージャーがシャットダウンするまでコマンドが実行されます。

-s

モデル・キュー オプションで、使用するモデル・キューを指定します。(デフォルトでは、`amqsrua` は `SYSTEM.DEFAULT.MODEL.QUEUE` キュー。)

-h

使用法

手順

1. サンプル・ディレクトリーから以下のコマンドを実行して、キュー・マネージャーで使用できるデータを表示します。

•  Linux Linux では、`MQ_INSTALLATION_PATH/samp/bin:`

```
./amqsrua -m QMgrName
```


- **Windows** Windows では、`MQ_INSTALLATION_PATH\tools\c\Samples\Bin64:`

```
amqsrua -m QMgrName
```

ここで、`QMgrName` は照会するキュー・マネージャーの名前を指定します。キュー・マネージャーが実行中でなければなりません。キュー・マネージャーの名前を指定しない場合、デフォルトのキュー・マネージャーが使用されます。

使用可能なオプションは次のとおりです。

```
CPU : Platform central processing units
DISK : Platform persistent data stores
STATMQI : API usage statistics
STATQ : API per-queue usage statistics
Enter Class selection
==>
```

2. CLASS オプションのリストから `STATMQI` を入力します。

```
==> 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. TYPE オプションのリストから `PUT` を入力します。

```
==>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
```

タスクの結果

amqsrua サンプル・アプリケーションを対話式に使用して、キュー・マネージャーがメタデータ接頭部 `$/SYS/MQ/INFO/QMGR` の下のシステム・トピックにパブリッシュする統計を確認しました。

注: キュー・マネージャーで使用できるリソース・パブリケーションのクラスとタイプは、構成、バージョン、プラットフォームによって異なります。**amqsrua** を対話式に使用すると、対象のキュー・マネージャーで使用できるクラス、タイプ、エレメントを確認できます。

次のタスク

独自のモニター・アプリケーションを開発する場合は、295 ページの『[独自のリソース・モニター・アプリケーションの開発](#)』を参照してください。

関連タスク

[Windows でのサンプル・プログラムの作成と実行](#)

[UNIX と Linux でのサンプル・プログラムの作成と実行](#)

ALW 独自のリソース・モニター・アプリケーションの開発

システム・リソースをモニターする独自のアプリケーションを開発できます。

各キュー・マネージャーは、トピックにリソース使用量データをパブリッシュします。このデータは、それらのトピックのサブスクライバーによって消費されます。キュー・マネージャーの始動時に、キュー・マネージャーは、メタトピックにメッセージのセットをパブリッシュします。これらのメッセージは、キュー・マネージャーによってサポートされているリソース使用量トピック、およびそれらのトピックにパブリッシュされるメッセージの内容を説明します。管理ツールは、メタデータにサブスクライブして、どのトピックのどのリソース使用量情報が使用可能かを見つけ、通知されるトピックにサブスクライブできます。

メタデータのトピック・ツリーの構造は、以下のとおりです。

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/instance]/type]
```

トピック・ツリーのメタデータには、以下の構造を使用します。

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/resourceid]/type]
```

最大 28 文字のアプリケーション名を持つモニター対象のリソースを指定していることを明確にし、内部メタデータ・タイプの属性と混同しないようにします。

有効なクラスのリストについては、292 ページの『[amqsrua コマンドを使用したシステム・リソース使用量のモニター](#)』を参照してください。

amqsrua アプリケーションのソース・コードが IBM MQ サンプルとして提供されています。独自のモニター・アプリケーションを作成するガイドとしてこのアプリケーションを使用できます。IBM MQ クライアント・インストールからサンプルのソースを取得できます。ソース・ファイルは amqsruaa.c という名前で、サンプル・ディレクトリーにあります。

- Linux AIX AIX and Linux プラットフォームの場合、MQ_INSTALLATION_PATH/samp/
- Windows Windows プラットフォームの場合、MQ_INSTALLATION_PATH\tools\c\Samples\

amqsrua アプリケーションは IBM MQ リソース使用量トピックにサブスクライブし、結果として生成されるパブリッシュ PCF データをフォーマットします。アプリケーションのソースは、このタイプの管理データをサブスクライブして取り込む方法の基本的な例を示しています。amqsrua アプリケーションは、以下のタスクを実行します。



- 入力パラメーターによって識別されるトピックの非永続サブスクリプションを作成します。
- MQGET を繰り返し呼び出してトピックからメッセージを取得し、stdout に書き込みます。
- MQI 理由コードごとにメッセージを書き込みます (MQRC_NONE 以外)。
- MQCC_FAILED の MQI 完了コードが存在する場合、またはリソース使用量パブリケーションの要求された数が消費された場合、停止します。

Multi システム・トピックにパブリッシュされるメトリック

メトリックはクラスに分類され、さらにタイプに細かく分類されます。それぞれのメトリック・クラスとタイプの下にパブリッシュされる各種メトリックがあります。

索引

- [296 ページの『CPU \(プラットフォームの中央演算処理装置\)』](#)
- [297 ページの『DISK \(プラットフォームの永続データ・ストア\)』](#)
- [298 ページの『STATMQI \(API 使用統計\)』](#)
- [300 ページの『STATQ \(API 使用統計 \(キュー単位\)\)』](#)
- [301 ページの『STATAPP \(使用統計 \(アプリケーション単位\)\)』](#)
- [301 ページの『NHAREPLICA \(インスタンスごとのネイティブ HA 統計\)』](#)

  リストされているオプション (NHAREPLICA を除く) のデータを収集する方法については、[292 ページの『amqsrua コマンドを使用したシステム・リソース使用量のモニター』](#) を参照してください。

[ALTER QMGR コマンド](#) を使用して STATMQI と STATQ をキュー・マネージャー・レベルでモニターすることも、ローカル・キュー属性 STATQ を使用して個々のキューをモニターすることもできます。このオプションについては、[ALTER QUEUES](#) を参照してください。

CPU (プラットフォームの中央演算処理装置)

概要

統計が 現行間隔を参照する場合、これはパブリッシュされたメッセージで MQIAMO64_MONITOR_INTERVAL パラメーターによって定義された間隔です。

統計は、少なくとも 1 つのアクティブなサブスクライバーが存在する限り、通常は 10 秒 (パブリッシュされた間隔) ごとにパブリッシュされますが、正確な間隔は常にメッセージから取得する必要があります。

重要: 特に指定されていない限り、メトリックは、キャプチャーされた時点の絶対値になります。

SystemSummary (CPU パフォーマンス - プラットフォーム全体)

ユーザー CPU 時間パーセンテージ X%

CPU が非特権コードであったときに CPU によって使用された時間の平均割合 (直前の 10 秒間隔にかかった時間の割合)。

システム CPU 時間パーセンテージ X%

CPU が特権コード内であったときに、CPU によって使用された時間の平均割合 (直近の 10 秒間にかかった時間の割合)。

CPU 負荷 - 1 分間の平均 X

1 分間の負荷平均。「負荷平均」は業界全体の用語ですが、報告される正確な値はプラットフォームによって異なる場合があります。

CPU 負荷 - 5 分間の平均 X

5 分間の負荷平均。「負荷平均」は業界全体の用語ですが、報告される正確な値はプラットフォームによって異なる場合があります。

CPU 負荷 - 15 分間の平均 X

15 分の負荷平均。「負荷平均」は業界全体の用語ですが、報告される正確な値はプラットフォームによって異なる場合があります。

CPU システムの要約

RAM 空き領域パーセンテージ X%

RAM 合計バイト数 XMB

RAM 空き領域パーセンテージ X%

RAM 合計バイト数 XMB

QMGrSummary (CPU パフォーマンス - 実行中のキュー・マネージャー)

ユーザー CPU 時間 - キュー・マネージャーのパーセンテージ見積もり X%

このキュー・マネージャーのプロセスが非特権コードであったときに CPU によって使用された時間の平均割合 (直前の 10 秒間にかかった時間の割合)。

システム CPU 時間 - キュー・マネージャーのパーセンテージ見積もり X%

このキュー・マネージャーのプロセスが特権コードに含まれていたときに、CPU が使用した時間の平均割合 (直前の 10 秒間にかかった時間)。

RAM 合計バイト数 - キュー・マネージャーの見積もり XMB

これは、キュー・マネージャーによって使用されるメモリーの概算です。

DISK (プラットフォームの永続データ・ストア)

SystemSummary および QMGrSummary は、キャプチャーの時点での絶対値です。公開間隔について詳しくは、[概要](#)を参照してください。

SystemSummary (ディスク使用量 - プラットフォーム全体)

MQ エラー・ファイル・システム - 使用中バイト数 XMB

MQ エラー・ファイル・システム - 空き領域 X%

MQ FDC ファイル数 X

MQ トレース・ファイル・システム - 使用中バイト数 XMB

MQ トレース・ファイル・システム - 空き領域 X%

QMGrSummary (ディスク使用量 - 実行中のキュー・マネージャー)

キュー・マネージャー・ファイル・システム - 使用中バイト数 XMB

キュー・マネージャー・ファイル・システム - 空き領域 X%

ログ (ディスク使用量 - キュー・マネージャー・リカバリー・ログ)

ログ - 使用中バイト数 X

ログ - 最大バイト数 X

1 次エクステントと 2 次エクステントがすべていっぱいになった場合にログに書き込むことができる最大バイト数。これはログ・ファイル・システムのサイズより小さくなっています

ログ・ファイル・システム - 使用中バイト数 X

ログ・ファイル・システム - 最大バイト数 X

ログ-現行インターバル X で書き込まれた物理バイト数。

現行間隔の定義については、「[概要](#)」を参照してください。

現行インターバル X で書き込まれたログ論理バイト数

ログ - 書き込み待ち時間 X uSec

ディスクへの単一書き込みにかかる時間を表すローリング平均。

LogWriteIntegrity=TripleWrite の場合、ディスクに書き込まれる物理バイト数は、書き込まれる論理バイト数より大きくなります。

ログ書き込みサイズ X (ローリング平均も同様)。

ログ-アーカイブを待機しているエクステント X によって占有されています。

logtype= linear かつ **LogManagement = archive** の場合にのみ公開されます。詳しくは、[qm.ini](#) ファイルのログ・スタンザを参照してください。

メディア・リカバリーに必要なログ・スペース (MB) X。

logtype= linear の場合にのみ公開されます。

再使用可能エクステントが占有するログ・スペース (MB) X

logtype= linear かつ **LogManagement** = automatic の場合にのみ公開されます。詳しくは、[qm.ini](#) ファイルのログ・スタンザを参照してください。

ログ-現在使用中の 1 次スペース X%。

使用中のログ・ファイル・スペース (1 次ログのパーセンテージ)。この値は 100% より大きくすることができます。

ログ・ワークロードの 1 次スペース使用率 X%。

最近のヒストリーでのローリング平均としての使用中のログ・ファイル・スペースの割合。

STATMQI (API 使用統計)

すべての API 使用統計には、公開された間隔のオカレンスまたはパーセンテージ (あるいはその両方) が反映されます。公開間隔の定義については、[概要](#) を参照してください。

統計には、失敗した MQI 呼び出しの数のカウントが出力されますが、これらの統計には失敗した MQI 呼び出しのすべてが表示されるわけではありません。実際、すべての MQI 呼び出しの失敗の統計が記録されるわけではありません。これは、MQI 呼び出しが統計が記録されるキュー・マネージャーの内部に到達する前に、MQI 呼び出しが失敗した理由の多くが診断されるためです。

この例としては、クライアント・アプリケーションに戻される MQRC_HCONN_ERROR があります。クライアント・アプリケーションが不良 **hconn** を渡すと、MQ クライアントはそのエラーを診断し、MQI 呼び出しをキュー・マネージャーに渡さずに MQRC_HCONN_ERROR を返します。したがって、失敗した MQI 呼び出しは、キュー・マネージャーによって記録された統計には表示されません。

失敗した MQI 呼び出しの統計は、不必要な失敗した MQI 呼び出しを生成する不適切に作成されたアプリケーションのトラブルシューティングを可能にし、パフォーマンスに影響を与えるため、役立ちます。統計に記録されるさまざまな MQI 呼び出しの失敗理由の例をいくつか以下に示します。

- MQCONN/MQCONN/MQOPEN は、クライアントではなくキュー・マネージャーによって診断されると、2035 MQRC_NOT_AUTHORIZED を返します。例えば、**amqsput** を nobody として実行します。
- MQPUT/MQPUT1 は、MAXDEPTH を超過したため、2053 MQRC_Q_FULL を返します。
- 空のキューを参照するか、空のキューから破壊的に取得すると、MQGET は 2033 MQRC_NO_MSG_AVAILABLE を返します。
- 保存メッセージがないため、MQSUBRQ は 2437 MQRC_NO_RETAINED_MSG を返します。

CONNDISC (MQCONN と MQDISC)

MQCONN/MQCONN の数 X

失敗した MQCONN/MQCONN の数 X

同時接続 - 最高水準点 X

MQDISC の数 X

OPENCLOSE (MQOPEN と MQCLOSE)

MQOPEN の数 X Y/秒

失敗した MQOPEN の数 X

MQCLOSE の数 X Y/秒

失敗した MQCLOSE の数 X

INQSET (MQINQ と MQSET)

MQINQ の数 X

失敗した MQINQ の数 X

MQSET の数 X

失敗した MQSET の数 X

PUT (MQPUT)

MQPUT/MQPUT1 の数のインターバルにおける合計 X

MQPUT/MQPUT1 のバイト数のインターバルにおける合計 X Y/秒

非持続メッセージ MQPUT の数 X
持続メッセージ MQPUT の数 X
失敗した MQPUT の数 X
非持続メッセージ MQPUT1 の数 X
持続メッセージ MQPUT1 の数 X
失敗した MQPUT1 の数 X
書き込まれた非持続メッセージ - バイト数 $X Y$ /秒
書き込まれた持続メッセージ - バイト数 X
MQSTAT の数 X

GET (MQGET)

破壊的 GET のインターバルにおける合計 - 数 X
破壊的 GET のインターバルにおける合計 - バイト数 $X Y$ /秒
非持続メッセージの破壊的 GET - 数 X
持続メッセージの破壊的 GET - 数 X
失敗した MQGET - 数 X
取得された非持続メッセージ - バイト数 $X Y$ /秒
取得された持続メッセージ - バイト数 X
非持続メッセージのブラウズ - 数 X
持続メッセージのブラウズ - 数 X
失敗したブラウズの数 X
非持続メッセージのブラウズ - バイト数 $X Y$ /秒
持続メッセージのブラウズ - バイト数 X
期限切れメッセージ数 X
ページされたキュー数 X
MQCB の数 X
失敗した MQCB の数 X
MQCTL の数 X

SYNCPOINT (コミットとロールバック)

コミットの数 X
ロールバックの数 X

SUBSCRIBE (サブスクライブ)

永続サブスクリプション作成回数 X
永続サブスクリプション変更回数 X
永続サブスクリプション再開回数 X
非永続サブスクリプション作成回数 X
サブスクリプションの作成/変更/再開に失敗した回数 X
永続サブスクリプション削除回数 X
非永続サブスクリプション削除回数 X
サブスクリプション削除に失敗した回数 X
MQSUBRQ の数 X
失敗した MQSUBRQ の数 X
永続サブスクライバー - 最高水準点 X
永続サブスクライバー - 最低水準点 X
非永続サブスクライバー - 最高水準点 X

非永続サブスクライバー - 最低水準点 X

PUBLISH (パブリッシュ)

トピック MQPUT/MQPUT1 のインターバルにおける合計 X

書き込まれたトピック・バイト数のインターバルにおける合計 X Y/秒

サブスクライバーへのパブリッシュ - メッセージ数 X

サブスクライバーへのパブリッシュ - バイト数 X

非持続 - トピック MQPUT/MQPUT1 の数 X

持続 - トピック MQPUT/MQPUT1 の数 X

失敗したトピック MQPUT/MQPUT1 の数 X

STATQ (API 使用統計 (キュー単位))

GENERAL (一般)

メッセージの有効期限が切れた X (IBM MQ 9.3.0 以降の CD バージョンの [GET](#) から移動)

キュー・ページ数 X (IBM MQ 9.3.0 およびそれ以降の CD バージョンの [GET](#) から移動)

平均キュー時間 X uSec (IBM MQ 9.3.0 およびそれ以降の CD バージョンの [GET](#) から移動)

キュー項目数 X (IBM MQ 9.3.0 およびそれ以降の CD バージョンの [GET](#) から移動)

OPENCLOSE (MQOPEN と MQCLOSE)

MQOPEN の数 X

MQCLOSE の数 X

INQSET (MQINQ と MQSET)

MQINQ の数 X

MQSET の数 X

PUT (MQPUT と MQPUT1)

MQPUT/MQPUT1 の数 X

MQPUT バイト数 X

MQPUT 非持続メッセージ数 X

MQPUT 持続メッセージ数 X

ロールバックされた MQPUT の数 X

MQPUT1 非持続メッセージ数 X

MQPUT1 持続メッセージ数 X

非持続バイト数 X

持続バイト数 X

ロック競合 $X\%$

キューをロックしようとした結果、別のプロセスが最初にロックを解放するのを待機した試行の割合。現在ロックされていないロックを取得することは、ロックが解放されるのを待機するよりも効率的であるため、ロック競合が減少すると、システムの最大スループットが増加する可能性があります。

キュー回避 PUT $X\%$

待機中の getter があるときにメッセージがキューに書き込まれる場合、そのメッセージを即時に getter に渡すことができるため、メッセージをキューに入れる必要がない場合があります。したがって、このメッセージはキューを回避したと言われ、「キュー回避書き込み」はそのようなメッセージのカウントです。キューの回避を増やすと、システムの最大スループットが増加する可能性があります。これは、メッセージをキューに入れ、再びキューから取り出すコストを回避できるためです。

キュー回避バイト $X\%$

待機中の getter があるときにメッセージがキューに書き込まれる場合、そのメッセージを即時に getter に渡すことができるため、メッセージをキューに入れる必要がない場合があります。したがって、このメッセージはキューを回避したと言われ、「キュー回避バイト」はそのようなバイトのカウントです。

キューの回避を増やすと、システムの最大スループットが増加する可能性があります。これは、メッセージをキューに入れ、再びキューから取り出すコストを回避できるためです。

GET (MQGET)

MQGET の数 X

MQGET バイト数 X

破壊的 MQGET 非持続メッセージ数 X

破壊的 MQGET 持続メッセージ数 X

ロールバックされた MQGET の数 X

破壊的 MQGET 非持続バイト数 X

破壊的 MQGET 持続バイト数 X

MQGET ブラウズ非持続メッセージ数 X

MQGET ブラウズ持続メッセージ数 X

MQGET ブラウズ非持続バイト数 X

MQGET ブラウズ持続バイト数 X

有効期限が切れたメッセージ X (IBM MQ 9.3 から [GENERAL](#) に移動)

キュー・ページ数 X (IBM MQ 9.3 から [GENERAL](#) に移動)

平均キュー時間 X uSec (IBM MQ 9.3 から [GENERAL](#) に移動)

キュー項目数 X (IBM MQ 9.3 から [GENERAL](#) に移動)

破壊的 MQGET 失敗数 X

MQRC_NO_MSG_AVAILABLE が原因の破壊的 MQGET 失敗数 X

MQRC_TRUNCATED_MSG_FAILED が原因の破壊的 MQGET 失敗数 X

MQGET ブラウズ失敗数 X

MQRC_NO_MSG_AVAILABLE が原因の MQGET ブラウズ失敗数 X

MQRC_TRUNCATED_MSG_FAILED が原因の MQGET ブラウズ失敗数 X

STATAPP (使用統計 (アプリケーション単位))

INSTANCE (インスタンス統計)

インスタンス・カウント X absolute

移動可能インスタンス・カウント X 絶対

インスタンス不足カウント X absolute

開始されたインスタンス X 間隔

開始されたアウトバウンド・インスタンス移動 X 間隔

完了したアウトバウンド・インスタンス移動 X 間隔

再接続中に終了したインスタンス X 間隔

終了したインスタンス X 間隔

NHAREPLICA (インスタンスごとのネイティブ HA 統計)

REPLICATION (レプリケーション統計)

送信された同期ログ・バイト数 X

送信されたキャッチアップ・ログ・バイト数 X

V 9.4.0 送信された同期圧縮ログ・バイト数 X

V 9.4.0 送信されたキャッチアップ圧縮ログ・バイト数 X

V 9.4.0 送信された同期非圧縮ログ・バイト数 X

V 9.4.0 送信されたキャッチアップ非圧縮ログ・バイト数 X

ログ書き込みの平均確認応答待ち時間 X uSec

ログ書き込みの平均確認応答サイズ X

バックログ・バイト X

バックログ平均バイト X

関連情報

[IBM MQ Operator の使用時にパブリッシュされるメトリック](#)

リアルタイム・モニター

リアルタイム・モニターは、キュー・マネージャー内のキューおよびチャネルの現在の状態を判別する際に使用できる技法です。返される情報は、コマンドが発行された時点において正確なものです。

多くのコマンドが使用可能です。これらのコマンドは、発行されるとキューおよびチャネルに関するリアルタイム情報を返します。1つ以上のキューまたはチャネルに関する情報を返すことができます。情報の量はまちまちです。リアルタイム・モニターは、以下の作業の中で使用できます。

- システム管理者が IBM MQ システムの定常状態を把握するのを支援する。これは、システムで問題が発生したときの問題診断に役立ちます。
- 任意の時点でのキュー・マネージャーの状態を判別する。特定のイベントまたは問題が検出されなかった場合でも、同様に状態を判別することができます。
- システムにおける問題の原因の判別を支援する。

リアルタイム・モニターで、キューまたはチャネルのいずれかの情報が返されます。返されるリアルタイム情報の量は、キュー・マネージャー属性、キュー属性、およびチャネル属性によって制御されます。

- キューが正しくサービスを受けていることを確認するためのコマンドを発行して、キューをモニターします。キュー属性によっては、使用の前にそれらでリアルタイム・モニターを有効にする必要があります。
- チャネルが正しく稼働していることを確認するためのコマンドを発行して、チャネルをモニターします。チャネル属性によっては、使用の前にそれらでリアルタイム・モニターを有効にする必要があります。

キューおよびチャネルのリアルタイム・モニターは、パフォーマンス・モニターおよびチャネル・イベント・モニターとともに行われますが、それらとは区別されます。

リアルタイム・モニターを制御する属性

リアルタイム・モニターが有効になっている場合、いくつかのキュー状態属性とチャネル状態属性でモニター情報を保持します。リアルタイム・モニターが有効になっていない場合、モニター情報はこれらのモニター属性に保持されません。下の各例では、これらのキュー状態属性とチャネル状態属性をどのように使用できるかを示しています。

リアルタイム・モニターは、個々のキューまたはチャネル、あるいは複数のキューまたはチャネルに対して有効または無効にすることができます。個々のキューまたはチャネルを制御するには、キュー属性 MONQ またはチャネル属性 MONCHL をリアルタイム・モニターを有効または無効にするように設定します。複数のキューまたはチャネルを同時に制御する場合は、キュー・マネージャー属性 MONQ および MONCHL を使用して、リアルタイム・モニターをキュー・マネージャー・レベルで有効または無効にできます。モニター属性にデフォルト値 QMGR が指定されているキュー・オブジェクトおよびチャネル・オブジェクトについてはすべて、リアルタイム・モニターはキュー・マネージャー・レベルで制御されます。

自動定義されたクラスター送信側チャネルは IBM MQ オブジェクトではないため、チャネル・オブジェクトが持っているような属性は持っていません。自動定義されたクラスター送信側チャネルを制御するには、キュー・マネージャー属性 MONACLS を使用します。この属性により、キュー・マネージャー内で自動定義されたクラスター送信側チャネルのチャネル・モニターを有効にするか無効にするかが決まります。

チャネルのリアルタイム・モニターの場合は、MONCHL 属性を、低、中、高の 3 つのモニター・レベルのいずれかに設定できます。モニター・レベルは、オブジェクト・レベルまたはキュー・マネージャー・レベルのいずれかに設定できます。レベルの選択は、ご使用のシステムによって決まります。モニター・データの収集には、計算に相対的に負荷の高い命令(システム時刻の取得など)がいくつか必要になる場合があります。リアルタイム・モニターの影響を低減するために、中および低のモニター・オプションでは、

常にデータを収集するのではなく、一定の間隔でデータのサンプルを測定します。303 ページの表 30 に、チャンネルのリアルタイム・モニターで使用可能な モニター・レベルの要約を示します。

レベル	説明	使用法
低	少数のデータ・サンプルを通常の間隔で測定します。	大量のメッセージを処理するオブジェクトで使用されます。
中	データ・サンプルを通常の間隔で測定します。	ほとんどのオブジェクトで使用されます。
高	すべてのデータを通常の間隔で測定します。	最新情報が重要である場合に、秒あたりに少数のメッセージのみを処理するオブジェクトで使用されます。

キューのリアルタイム・モニターの場合は、MONQ 属性を、低、中、高の 3 つのモニター・レベルのいずれかに設定できます。ただし、これら 3 つの値の違いはありません。これらの値はすべて、データ収集を可能にしますが、サンプルのサイズには影響しません。

例

以下の各例では、必要なキュー属性、チャンネル属性、およびキュー・マネージャー属性を設定してモニターのレベルを制御する方法を示しています。どの例でも、モニターを有効にするとき、キュー・オブジェクトとチャンネル・オブジェクトのモニター・レベルは中に設定されています。

1. すべてのキューとチャンネルに対し、キュー・モニターとチャンネル・モニターの両方をキュー・マネージャー・レベルで有効にするには、次のコマンドを使用します。

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(QMGR)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(QMGR)
```

2. ローカル・キュー Q1 と送信側チャンネル QM1.TO.QM2 を除く、すべてのキューとチャンネルのモニターを有効にするには、次のコマンドを使用します。

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(OFF)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(OFF)
```

3. ローカル・キュー Q1 と送信側チャンネル QM1.TO.QM2 を除く、すべてのキューとチャンネルのキュー・モニターとチャンネル・モニターの両方を無効にするには、次のコマンドを使用します。

```
ALTER QMGR MONQ(OFF) MONCHL(OFF)
ALTER QL(Q1) MONQ(MEDIUM)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(MEDIUM)
```

4. すべてのキューとチャンネルに対し、個々のオブジェクト属性とは無関係に、キュー・モニターとチャンネル・モニターの両方を無効にするには、次のコマンドを使用します。

```
ALTER QMGR MONQ(NONE) MONCHL(NONE)
```

5. 自動的に定義されたクラスター送信側チャンネルのモニター機能を制御するには、次のコマンドを使用します。

```
ALTER QMGR MONACLS(MEDIUM)
```

6. 自動的に定義されたクラスター送信側チャンネルが、チャンネル・モニター用にキュー・マネージャー設定を使用することを指定するには、次のコマンドを使用します。

関連概念

302 ページの『リアルタイム・モニター』

リアルタイム・モニターは、キュー・マネージャー内のキューおよびチャネルの現在の状態を判別するために使用できる技法です。返される情報は、コマンドが発行された時点において正確なものです。

322 ページの『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.

関連タスク

304 ページの『キューおよびチャネルのモニター・データの表示』

キューまたはチャネルのリアルタイム・モニター情報を表示するには、IBM MQ Explorer または該当の MQSC コマンドのいずれかを使用します。モニター・フィールドによっては、コンマ区切りの標識の値のペアが表示されます。これらの値は、ご使用のキュー・マネージャーの操作のモニターに役立ちます。各例では、モニター・データをどのように表示できるかを示しています。

キュー・マネージャーの属性の表示および変更

モニター (MONCHL)

キューおよびチャネルのモニター・データの表示

キューまたはチャネルのリアルタイム・モニター情報を表示するには、IBM MQ Explorer または該当の MQSC コマンドのいずれかを使用します。モニター・フィールドによっては、コンマ区切りの標識の値のペアが表示されます。これらの値は、ご使用のキュー・マネージャーの操作のモニターに役立ちます。各例では、モニター・データをどのように表示できるかを示しています。

このタスクについて

コマンドで区切られた値のペアを表示するモニター・フィールドには、オブジェクトのモニターが有効になってから以後、またはキュー・マネージャーが開始されてから以後に測定された時間の短期標識および長期標識が提供されます。

- 短期標識はペアの最初の値で、より新しい測定ほど重み付けされ、この測定値への影響が大きくなるように計算されます。これにより、測定における最近の傾向が示されます。
- 長期標識はペアの 2 番目の値で、より新しい測定でもあまり重み付けされないように計算されます。これにより、リソースのパフォーマンスに関して長期のアクティビティーが示されます。

これらの標識の値が最も役立つのは、キュー・マネージャーの操作における変化を検出する場合です。この場合、時間の増加を検出するためには、通常使用時に標識が示す時間を知っておくことが必要です。これらの値を定期的に収集して確認することにより、キュー・マネージャーの操作における変動を検出できます。この変動が、パフォーマンスにおける変化を表している可能性があります。

リアルタイム・モニター情報は以下のようにして取得します。

手順

1. キューのリアルタイム・モニター情報を表示するには、IBM MQ Explorer を使用するか、またはオプション・パラメーター MONITOR を指定した MQSC コマンド DISPLAY QSTATUS を使用します。
2. チャネルのリアルタイム・モニター情報を表示するには、IBM MQ Explorer を使用するか、またはオプション・パラメーター MONITOR を指定した MQSC コマンド DISPLAY CHSTATUS を使用します。

例

キュー Q1 の属性 MONQ はデフォルト値 QMGR に設定されており、このキューを所有するキュー・マネージャの属性 MONQ は MEDIUM に設定されています。このキューについて収集されたモニター・フィールドを表示するには、次のコマンドを使用します。

```
DISPLAY QSTATUS(Q1) MONITOR
```

キュー Q1 のモニター・フィールドとモニター・レベルは、以下のように表示されます。

```
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)
```

送信側チャンネル QM1.TO.QM2 の属性 MONCHL はデフォルト値 QMGR に設定されており、このキューを所有するキュー・マネージャの属性 MONCHL は MEDIUM に設定されています。この送信側チャンネルについて収集されたモニター・フィールドを表示するには、次のコマンドを使用します。

```
DISPLAY CHSTATUS(QM1.TO.QM2) MONITOR
```

送信側チャンネル QM1.TO.QM2 のモニター・フィールドとモニター・レベルは、以下のように表示されます。

```
CHSTATUS(QM1.TO.QM2)
XMITQ(Q1)
CONNNAME(127.0.0.1)
CURRENT
CHLTYPE(SDR)
STATUS(RUNNING)
SUBSTATE(MQGET)
MONCHL(MEDIUM)
XQTIME(755394737,755199260)
NETTIME(13372,13372)
EXITTIME(0,0)
XBATCHSZ(50,50)
COMPTIME(0,0)
STOPREQ(NO)
RQMNAME(QM2)
```

関連概念

[302 ページの『リアルタイム・モニター』](#)

リアルタイム・モニターは、キュー・マネージャ内のキューおよびチャンネルの現在の状態を判別する際に使用できる技法です。返される情報は、コマンドが発行された時点において正確なものです。

関連資料

表示キューの状況

キューのモニター

このページを使用して、キューとそのキューにサービスを提供するアプリケーションに関する問題の解決に役立つタスクを表示します。問題を調べるための各種モニター・オプションを使用できます。

多くの場合、サービスが提供されているキューの問題の最初の徴候は、キュー上のメッセージ数 (CURDEPTH) が増加することです。メッセージ数が1日のうち特定の時刻、または特定のワークロード下で増加することが予想されている場合、この増加は問題を示してはいない可能性があります。ただし、メッセージ数の増加が説明できないものである場合は、その原因を調査できます。

アプリケーション・キューのアプリケーションに問題があるか、または伝送キューのチャンネルに問題がある可能性があります。キューにサービスを提供するアプリケーションがチャンネルである場合に使用可能な、追加のモニター・オプションがあります。

以下の各例では、Q1 という特定のキューに関する問題を調べて、各種コマンドの出力で確認するフィールドについて説明しています。

アプリケーションでキューをオープンしているかどうかを確認する

キューに関する問題が存在する場合は、アプリケーションでキューをオープンしているかどうかを確認します。

このタスクについて

以下の手順を実行して、アプリケーションでキューをオープンしているかどうかを確認します。

手順

1. キューに対して実行中のアプリケーションが、対象となるアプリケーションであることを確認します。問題のキューに対して次のコマンドを発行します。

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) ALL
```

出力の APPLTAG フィールドを調べ、アプリケーションの名前が表示されていることを確認します。アプリケーションの名前が表示されない場合、またはまったく出力がない場合は、アプリケーションを開始します。

2. キューが伝送キューである場合は、出力の CHANNEL フィールドを調べます。チャンネル名が CHANNEL フィールドに表示されていない場合は、チャンネルが稼働中かどうかを調べます。
3. キューに対して実行中のアプリケーションに、入力用にオープンしているキューがあることを確認します。以下のコマンドを発行します。

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

出力の IPPROCS フィールドを調べて、入力用にオープンしているキューがアプリケーションにあるかどうかを確認します。この値が 0 で、これがユーザー・アプリケーションのキューである場合は、メッセージをキューから取り出すために、アプリケーションが入力用にキューをオープンしていることを確認します。

キュー上のメッセージが使用可能であることを確認する

キュー上に多数のメッセージがあり、アプリケーションでそれらのメッセージのいずれも処理していない場合は、キュー上のメッセージがアプリケーションで使用可能であることを確認してください。

このタスクについて

以下の手順を実行して、アプリケーションがキューからのメッセージを処理していない理由を調査します。

手順

1. キュー上のメッセージをすべて処理する必要がある場合は、アプリケーションが特定のメッセージ ID または関連 ID を要求していないことを確認します。
2. 現在のキュー・サイズはキュー上のメッセージ数が増加していることを示す場合がありますが、コミットされていないためにアプリケーションがキュー上のメッセージの一部を取得できない場合があります。現在のサイズには、キューに対してコミットされていないメッセージの MQPUT の数が含まれます。以下のコマンドを発行します。

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

出力の UNCOM フィールドを調べて、コミットされていないメッセージがキューにあるかどうかを確認します。

3. アプリケーションがキューからメッセージを取得しようとしている場合は、書き込み側のアプリケーションがメッセージを正しくコミットしているかどうかを確認します。このキューにメッセージを書き込んでいるアプリケーションの名前を確認するには、次のコマンドを発行します。

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) OPENTYPE(OUTPUT)
```

4. 続いて、次のコマンドを発行します。 *appltag* には、前のコマンドの出力から APPLTAG 値を入れます。

```
DISPLAY CONN(*) WHERE(APPLTAG EQ appltag) UOWSTDA UOWSTTI
```

これにより、この作業単位が開始された時刻が示され、アプリケーションが長期間実行される作業単位を作成しているかどうかを確認できます。書き込み側アプリケーションがチャンネルの場合は、バッチの完了に時間がかかる理由を調べることができます。

アプリケーションでメッセージをキューから取り出しているかどうかを確認する

キューおよびそのキューにサービスを提供するアプリケーションに関する問題が存在する場合は、アプリケーションでメッセージをキューから取り出しているかどうかを確認します。

このタスクについて

アプリケーションがメッセージをキューから取り出しているかどうかを確認するには、以下の確認を行います。

手順

1. キューに対して実行中のアプリケーションが、実際にキューからのメッセージを処理していることを確認します。以下のコマンドを発行します。

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

出力で、キューから最後に取得が実行された時刻を示す LGETDATE フィールドと LGETTIME フィールドを調べます。

2. このキューからの最後の取得が予想よりもずっと前であった場合は、アプリケーションが正しくメッセージを処理していることを確認します。
このアプリケーションがチャンネルの場合、メッセージがそのチャンネルを通過しているかどうかを確認します。

アプリケーションでメッセージを十分な速度で処理できるかどうかを調べる

メッセージがキュー上で増加していて、その他の検査でも処理上の問題が見つからない場合は、アプリケーションでメッセージを十分な速度で処理できるかどうかを確認します。このアプリケーションがチャンネルの場合、そのチャンネルでメッセージを十分な速度で処理できることを確認します。

このタスクについて

アプリケーションでメッセージを十分な速度で処理しているかどうかを調べるには、以下のテストを実行します。

手順

1. 次のコマンドを定期的に発行して、キューに関するパフォーマンス・データを収集します。

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

QTIME 標識の値が大きいか、または対象の期間に増加していて、キュー上のメッセージが使用可能であることを確認して長期間実行される作業単位の可能性がない場合、取得側アプリケーションが書き込み側アプリケーションの速度に追いついていない可能性があります。

2. 取得側アプリケーションが書き込み側アプリケーションに追いつけない場合は、キューを処理する別の取得側アプリケーションを追加することを検討してください。

別の取得側アプリケーションの追加が可能かどうかは、アプリケーションの設計、および複数のアプリケーションでキューを共有できるかどうかによって決まります。 相関 ID によるメッセージのグループ化や取得などの機能によって、2つのアプリケーションでキューを同時に処理できる場合があります。

現在のサイズが増加していない場合のキューの確認

現在のキュー・サイズが増加していない場合でも、アプリケーションでメッセージを正しく処理しているかどうかを確認するために、キューをモニターすることは有効です。

このタスクについて

キューに関するパフォーマンス・データを収集するには、下のコマンドを定期的に発行します。

手順

次のコマンドを定期的に発行してください。

```
DISPLAY QSTATUS(Q1) TYPE(Queue) MSGAGE QTIME
```

アプリケーションがすべてのメッセージを処理するように設計されている場合、この出力で MSGAGE の値が時間の経過とともに増加していると、いくつかのメッセージがまったく処理されていない可能性があります。

チャネルのモニター

このページを使用して、伝送キューとそのキューにサービスを提供するチャネルに関する問題の解決に役立つタスクを表示します。問題を調べるための各種チャネル・モニター・オプションを使用できます。

多くの場合、サービスが提供されているキューの問題の最初の徴候は、キュー上のメッセージ数 (CURDEPTH) が増加することです。メッセージ数が1日のうちの特定の時刻、または特定のワークロード下で増加することが予想されている場合、この増加は問題を示してはいない可能性があります。ただし、メッセージ数の増加が説明できないものである場合は、その原因を調査できます。

伝送キューにサービスを提供するチャネルに関する問題がある可能性があります。問題を調べるために役立つ各種チャネル・モニター・オプションを使用できます。

以下の各例では、QM2 という伝送キューと QM1.TO.QM2 というチャネルに関する問題を調査します。このチャネルは、キュー・マネージャー QM1 からキュー・マネージャー QM2 にメッセージを送信するために使用されます。キュー・マネージャー QM1 は送信側チャネルまたはサーバー・チャネルとして定義され、キュー・マネージャー QM2 は受信側チャネルまたは要求側チャネルとして定義されます。

チャネルが稼働中かどうかを確認する

伝送キューに関する問題が存在する場合は、チャネルが稼働中かどうかを確認します。

このタスクについて

以下の手順を実行して、伝送キューにサービスを提供しているチャネルの状況を確認します。

手順

1. 次のコマンドを発行して、伝送キュー QM2 を処理することになっているチャネルを確認します。

```
DIS CHANNEL(*) WHERE(XMITQ EQ QM2)
```


この例では、このコマンドの出力で、伝送キューに対してサービスを提供しているチャンネルが QM1.TO.QM2 であることが示されます。

2. 次のコマンドを発行して、チャンネル QM1.TO.QM2 の状況を確認します。

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

3. **CHSTATUS** コマンドからの出力の **STATUS** フィールドを検査します。

- **STATUS** フィールドの値が **RUNNING** の場合は、チャンネルがメッセージを移動していることを確認します。
- コマンドからの出力に状況が示されていないか、または **STATUS** フィールドの値が **STOPPED**、**RETRY**、**BINDING**、**REQUESTING** の場合は、以下の該当するステップを実行します。

4. オプション: **STATUS** フィールドの値に状況が示されていない場合、チャンネルが非アクティブのため、以下の手順を実行してください。

- a) チャンネルがトリガーによって自動的に開始されている場合は、伝送キュー上のメッセージが使用可能であることを確認します。

伝送キューに使用可能なメッセージがある場合は、伝送キューのトリガー設定が正しいことを確認します。

- b) 次のコマンドを発行して、手動でもう一度チャンネルを開始します。

```
START CHANNEL(QM1.TO.QM2)
```


5. オプション: **STATUS** フィールドの値が **STOPPED** である場合は、以下のステップを実行してください。

- a) エラー・ログを調べて、チャンネルが停止した原因を判別します。エラーが原因でチャンネルが停止した場合は、その問題を修正します。

また、チャンネルの再試行属性 **SHORTRTY** および **LONGRTY** に値を指定してください。これにより、ネットワーク・エラーなどの一時的な障害時に、チャンネルが自動的に再始動を試みます。

- b) 次のコマンドを発行して、手動でもう一度チャンネルを開始します。

```
START CHANNEL(QM1.TO.QM2)
```

 IBM MQ for z/OS では、コマンド・イベント・メッセージを使用することで、ユーザーがチャンネルをいつ停止したのかを検出できます。

6. オプション: **STATUS** フィールドの値が **RETRY** である場合は、以下のステップを実行してください。

- a) エラー・ログを確認してエラーを特定した後、この問題を修正します。

- b) 次のコマンドを発行して、手動でもう一度チャンネルを開始します。

```
START CHANNEL(QM1.TO.QM2)
```

またはチャンネルが次の再試行で正常に接続されるのを待ちます。

7. オプション: **STATUS** フィールドの値が **BINDING** または **REQUESTING** の場合は、チャンネルがまだパートナーに正常に接続されていません。以下のステップを実行します。

- a) 両側のチャンネルで次のコマンドを発行して、チャンネルの副状態を確認します。


```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

注:

- i) 一方のチャンネルだけに副状態がある場合があります。

- ii) 多くの副状態は一時的なものであるため、チャンネルが特定の副状態に留まっているかどうかを確認するには、コマンドを何度か発行してください。

- b) [310 ページの表 31](#) を確認して、次に行うアクションを決定します。

表 31. 状況がバインディングまたは要求の場合に見られる副状態		
開始側 MCA の副状態 ¹	応答側 MCA の副状態 ²	注
NAMESERVER		開始側 MCA が、ネーム・サーバー要求が完了するのを待っています。必ず、チャンネル属性 CONNAME に正しいホスト名を指定し、ネーム・サーバーを正しくセットアップするようにしてください。
SCYEXIT	SCYEXIT	MCA は現在、セキュリティー出口経由で通信中です。詳しくは、312 ページの『チャンネルでメッセージを十分な速度で処理できるかどうかを調べる』を参照してください。
	CHADEXIT	チャンネル自動定義出口が正しく実行中です。詳しくは、312 ページの『チャンネルでメッセージを十分な速度で処理できるかどうかを調べる』を参照してください。
RCVEXIT SENDEXIT MSGEXIT MREXIT	RCVEXIT SENDEXIT MSGEXIT MREXIT	出口は MQXR_INIT のチャンネル開始時に呼び出されます。長時間かかる場合は、出口のこの部分の処理について検討してください。詳しくは、312 ページの『チャンネルでメッセージを十分な速度で処理できるかどうかを調べる』を参照してください。
SERIALIZE	SERIALIZE	この副状態は、属性指定が SHARED のチャンネルにのみ適用されます。
NETCONNECT		この副状態は、ネットワーク構成が正しくないために、接続に遅延がある場合にのみ示されます。
SSLHANDSHAKE	SSLHANDSHAKE	TLS ハンドシェイクは、いくつかの送信と受信で構成されます。ネットワークの応答時間やルックアップ CRL への接続が遅い場合、これはハンドシェイクにかかる時間に影響します。  IBM MQ for z/OS では、この副状態は十分な SSLTASKS がないことを示している可能性もあります。

注：

- i) 開始側 MCA は、通信を開始した側のチャンネルです。これは送信側、クラスター送信側、完全修飾サーバー、および要求側のいずれかになります。サーバーと要求側のペアの場合は、チャンネルの開始側です。
- ii) 応答側 MCA は、通信の開始要求に応答した側のチャンネルです。受信側、クラスター受信側、要求側 (サーバーまたは送信側が開始された場合)、サーバー (要求側が開始された場合)、および送信側 (要求側チャンネルと送信側チャンネルのコールバック・ペアの場合) がこれに該当します。

チャンネルでメッセージが移動されていることを確認する

伝送キューに関する問題が存在する場合は、チャンネルでメッセージが移動されていることを確認します。

始める前に

コマンド DIS CHSTATUS(QM1.TO.QM2) ALL を発行します。STATUS フィールドの値が RUNNING の場合は、チャンネルはパートナー・システムに正常に接続されています。

伝送キュー上にコミットされていないメッセージがないことを 306 ページの『キュー上のメッセージが使用可能であることを確認する』で説明されているように確認します。

このタスクについて

チャンネルで取得および送信できるメッセージが存在する場合は、以下の確認を行ってください。

手順

1. チャンネル状況の表示コマンド DIS CHSTATUS(QM1.TO.QM2) ALL の出力で、以下のフィールドを調べます。

MSGS

このセッション中 (チャンネルが開始されてから以後) に送信または受信されたメッセージの数 (サーバー接続チャンネルの場合は、処理された MQI 呼び出しの数)。

BUFSENT

送信された送信バッファの数。これには、制御情報のみを送信する伝送が含まれます。

BYTSENT

このセッション中 (チャンネルが開始されてから以後) に送信されたバイト数。これには、メッセージ・チャンネル・エージェントによって送信された制御情報が含まれます。

LSTMSGDA

最後のメッセージが送信された日付、または MQI 呼び出しが処理された日付。LSTMSGTI を参照してください。

LSTMSGTI

最後のメッセージが送信された時刻、または MQI 呼び出しが処理された時刻。送信側またはサーバーの場合、これは最後のメッセージ (メッセージが分割されている場合は最後の部分) が送信された時刻です。要求側または受信側の場合、これは最後のメッセージがターゲット・キューに入れられた時刻です。サーバー接続チャンネルの場合、これは最後の MQI 呼び出しが完了した時刻です。

CURMSGGS

送信側チャンネルの場合、これは現在のバッチで送信されたメッセージの数です。受信側チャンネルの場合、これは現在のバッチで受信されたメッセージの数です。送信側チャンネルの場合も受信側チャンネルの場合も、バッチがコミットされると、この値はゼロにリセットされます。

状況

チャンネルの状況。これは、Starting、Binding、Initializing、Running、Stopping、Retrying、Paused、Stopped、または Requesting となる可能性があります。

SUBSTATE

チャンネルが現在実行中のアクション。

INDOUBT

チャンネルが現在未確定かどうか。これは、送信側メッセージ・チャンネル・エージェントが送信した一連のメッセージが正常に受信されたことを示す確認応答をその送信側メッセージ・チャンネル・エージェント自体が待機している間でのみ YES です。それ以外は NO です。メッセージの送信中の、肯定応答がまだ要求されていない期間も NO です。受信側チャンネルの場合、この値は常に NO です。

2. チャンネルが開始されてから以後に、メッセージを送信したかどうかを確認します。メッセージが送信された場合は、最後のメッセージがいつ送信されたかを確認します。
3. チャンネルがバッチを開始しており、そのバッチがまだ完了していない (CURMSGGS の値がゼロ以外) 可能性があります。INDOUBT が YES の場合、チャンネルは、チャンネルのもう一方の端がバッチを受信したという肯定応答の受信を待機しています。出力の SUBSTATE フィールドを調べて、[311 ページの表 32](#) を参照してください。

送信側 SUBSTATE	受信側 SUBSTATE	注
MQGET	RECEIVE	静止しているチャンネルの通常の状態。
SEND	RECEIVE	通常、SEND は一時的な状態です。SEND が示された場合は、通信プロトコル・バッファがいっぱいであることを示します。これはネットワークの問題を示している可能性があります。
RECEIVE		送信側が一定期間 RECEIVE 副状態であることが示される場合は、バッチの完了またはハートビートに対する応答を待っています。バッチの完了に時間がかかる理由を確認できます。

注: また、チャンネルでメッセージを十分な速度で処理できるかどうかを (特にチャンネルに終了処理に関連付けられた副状態がある場合) 調べることができます。

バッチの完了に時間がかかる理由を確認する

バッチの完了に長時間かかるのは、ネットワーク速度が遅いか、チャンネルがメッセージ再試行処理を使用していることが原因である可能性があります。

このタスクについて

送信側チャンネルは、メッセージのバッチの送信が完了すると、チャンネルがパイプライン化されていない限り、そのバッチの確認が受信側から送信されることを待機します。以下のタスクに記述されている要因が、送信側チャンネルの待機時間に影響を及ぼす可能性があります。

手順

- ネットワーク速度が低下しているかどうかの確認
NETTIME 値は、チャンネルのリモート・エンドにバッチ終了要求を送信してから応答を受け取るのにかかる時間から、バッチ終了要求を処理する時間を差し引いた時間 (マイクロ秒単位) です。この値は、以下のいずれかの理由で大きくなる可能性があります。
 - ネットワーク速度が遅い。 ネットワーク速度が低い場合は、バッチの完了にかかる時間に影響を与える可能性があります。 NETTIME フィールドの標識となる測定は、バッチの最後で行われます。ただし、ネットワークのスローダウンに影響を受ける最初のバッチが、NETTIME 値の変化によって示されることはありません。この値はバッチの終わりで測定されるためです。
 - 要求はリモート側でキューに入れられます。例えば、チャンネルが PUT を再試行している場合や、あるいは、ページ・セット I/O のために PUT 要求が遅くなっている場合があります。キューに入れられた要求が完了すると、バッチ要求の終了の所要時間が測定されます。そのため、NETTIME 値が大きい場合は、リモート・エンドに異常な処理がないか確認してください。
- チャンネルでメッセージ再試行を使用しているかどうかを確認します。
受信側チャンネルがターゲット・キューへのメッセージの書き込みに失敗すると、直ちに送達不能キューにメッセージを書き込むのではなく、メッセージの再試行処理を使用する場合があります。再試行処理によってバッチの速度が低下することがあります。次の MQPUT が試行されるまでの間、チャンネルの状況は、メッセージ再試行間隔が経過するのを待っていることを示す STATUS(PAUSED) になります。

チャンネルでメッセージを十分な速度で処理できるかどうかを調べる

伝送キュー上でメッセージが増加しているが、処理の問題が見つからない場合は、チャンネルでメッセージを十分な速度で処理できるかどうかを調べます。

始める前に

次のコマンドを一定期間、繰り返し発行して、チャンネルに関するパフォーマンス・データを収集します。

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

このタスクについて

伝送キューにコミットされていないメッセージがないことを確認 ([306 ページの『キュー上のメッセージが使用可能であることを確認する』](#)に説明あり) した後、チャンネル状況の表示コマンドからの出力にある XQTIME フィールドを調べます。XQTIME 標識の値が定常的に高いか、または測定期間に増加した場合は、チャンネルが書き込み側アプリケーションの速度に追いついていないことを示しています。

以下のテストを実行します。

手順

1. 出口が処理中かどうかを確認します。
メッセージを送信しているチャンネルで出口が使用されている場合は、メッセージの処理にかかる時間が増加することがあります。これが該当するかどうかを確認するには、以下の確認を行います。

a) コマンド DIS CHSTATUS(QM1.TO.QM2) ALL の出力で、EXITTIME フィールドを確認します。
出口でかかる時間が予想より長い場合は、不要なループや余分な処理 (特にメッセージ出口、送信出口、および受信出口での処理) がないか、出口の処理を検討してください。このような処理は、チャンネルを移動したすべてのメッセージに影響を与えます。

b) コマンド DIS CHSTATUS(QM1.TO.QM2) ALL の出力で、SUBSTATE フィールドを確認します。
チャンネルにかなりの時間にわたって以下の副状態のいずれかがある場合、出口にある処理を確認します。

- SCYEXIT
- RCVEXIT
- SENDEXIT
- MSGEXIT
- MREXIT

チャンネルの副状態について詳しくは、表 311 ページの表 32 を参照してください。

2. ネットワーク速度が低下しているかどうかの確認

メッセージがチャンネル間を十分な速度で移動していない場合は、ネットワークが遅いことが考えられます。これが該当するかどうかを確認するには、以下の確認を行います。

a) コマンド DIS CHSTATUS(QM1.TO.QM2) ALL の出力で、NETTIME フィールドを確認します。

この標識は、送信側チャンネルがパートナーに応答を要求するときに測定されます。この測定は、各バッチの終わりと、ハートビート中にチャンネルがアイドルである場合に行われます。

b) メッセージの往復に予想よりも長い時間がかかることをこの標識が示している場合は、他のネットワーク・モニター・ツールを使用して、ネットワークのパフォーマンスを調べてください。

3. チャンネルで圧縮を使用しているかどうかを確認します。

チャンネルが圧縮を使用している場合は、メッセージの処理時間が増加します。チャンネルで 1 つの圧縮アルゴリズムしか使用していない場合は、以下の確認を行います。

a) コマンド DIS CHSTATUS(QM1.TO.QM2) ALL の出力で、COMPTIME フィールドを確認します。

この標識は、圧縮または解凍にかかる時間を示します。

b) 選択した圧縮によって送信データ量が期待されるほど減っていない場合は、圧縮アルゴリズムを変更してください。

4. チャンネルで複数の圧縮アルゴリズムを使用している場合は、以下の確認を行います。

a) コマンド DIS CHSTATUS(QM1.TO.QM2) ALL の出力で、COMPTIME、COMPHDR、および COMPMSG の各フィールドを確認します。

b) 選択した圧縮率やアルゴリズムによって必要な圧縮やパフォーマンスが得られない場合は、チャンネル定義で指定されている圧縮アルゴリズムを変更するか、メッセージ出口を作成して特定のメッセージ用にチャンネルの圧縮アルゴリズムの選択を指定変更することを検討します。

クラスター・チャンネルに関する問題の解決

SYSTEM.CLUSTER.TRANSMIT.QUEUE キュー上にメッセージが溜まっている場合、問題を診断するための最初のステップは、メッセージの送信に問題のあるチャンネルを見つけることです。

このタスクについて

SYSTEM.CLUSTER.TRANSMIT.QUEUE を使用して、メッセージの送信に問題のあるチャンネルを見つけるためのものです。以下の確認を実行します。

手順

1. 以下のコマンドを発行します。

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```


注: 多数のメッセージが移動しているビジーなクラスターがある場合は、高い数値を指定してこのコマンドを実行し、送信可能なメッセージ数が少ないチャンネルを除去することを検討してください。

2. フィールド XQMSGSA の値が大きいチャンネルの出力を調べます。チャンネルがメッセージを移動していない原因、または十分な速度で移動していない原因を特定してください。308 ページの『チャンネルのモニター』で概略されているタスクを使用して、蓄積を引き起こしていることがわかったチャンネルの問題を診断します。

クラスターのモニター

クラスター内で、アプリケーション・メッセージ、制御メッセージ、およびログをモニターできます。キューの複数のインスタンス間でクラスターのロード・バランシングが行われる場合は、モニターに関する特別な考慮事項があります。

クラスター内のアプリケーション・メッセージのモニター

通常、キュー・マネージャーから出るすべてのクラスター・メッセージは、メッセージの送信にどのクラスター送信側チャンネルが使用されているかに関係なく、SYSTEM.CLUSTER.TRANSMIT.QUEUE を通過します。各チャンネルは、そのチャンネルを宛先とするメッセージを、他のすべてのクラスター送信側チャンネルと並行して、排出します。このキューに蓄積されるメッセージが徐々に増える場合は、1 つまたは複数のチャンネルで問題が発生している可能性があるため、以下のことを調べる必要があります。

- クラスターを設計するためには、キューの深さを適切にモニターする必要があります。
- 次のコマンドを実行すると、伝送キューで複数のメッセージが待機しているすべてのチャンネルが返されません。

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

すべてのクラスター・メッセージが単一のキューに入っている場合は、キューがいっぱいになり始めたときに、どのチャンネルに問題があるのかを確認するのが困難である可能性があります。このコマンドを使用すると、どのチャンネルが原因なのかを簡単に確認することができます。

複数の送信キューを使用するように、クラスター・キュー・マネージャーを構成できます。キュー・マネージャー属性 DEFCLXQ を CHANNEL に変更すると、すべてのクラスター送信側チャンネルが異なるクラスター送信キューに関連付けられます。また、送信キューを個別に手動で構成することもできます。クラスター送信側チャンネルに関連付けられているすべてのクラスター伝送キューを表示するには、次のコマンドを実行します。

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

キュー名の左側に特定の語幹を付けるというパターンに従うように、クラスター送信キューを定義します。これにより、汎用キュー名を使用して、**DISPLAY CLUSMGR** コマンドから返されたすべてのクラスター送信キューの深さを照会できます。

```
DISPLAY QUEUE (qname *) CURDEPTH
```

クラスター内の制御メッセージのモニター

SYSTEM.CLUSTER.COMMAND.QUEUE キューは、ローカル・キュー・マネージャーによって生成されるか、クラスター内の他のキュー・マネージャーからこのキュー・マネージャーに送信される、キュー・マネージャーのすべてのクラスター制御メッセージを処理するために使用されます。キュー・マネージャーが自らのクラスター状態を正しく維持している場合、このキューは 0 に近づく傾向にあります。このキューに蓄積されるメッセージの数が一時的に増加することがありますが、以下のように考えられます。

- キューに多くのメッセージがあるということは、クラスターの状態が激しく変動していることを示しています。
- 重大な変更を行う場合は、次の変更までの間にキューの状態を落ち着かせます。例えば、リポジトリを移動する場合は、第 2 のリポジトリを移動する前に、キューが 0 になるようにします。

このキューにメッセージのバックログがある間は、クラスター状態の更新およびクラスター関連のコマンドは処理されません。メッセージがこのキューから長時間削除されない場合は、さらなる調査が必要です。まず、この状況を引き起こしているプロセスを説明するキュー・マネージャーのエラー・ログ（または z/OS の CHINIT ログ）を調査します。

SYSTEM.CLUSTER.REPOSITORY.QUEUE は、クラスター・リポジトリ・キャッシュ情報を多数のメッセージとして保持します。このキューにメッセージが常に存在するのはよくあることであり、大規模なクラスターでは、そのようなことがより頻繁に起こります。したがって、このキューに蓄積されているメッセージの数は懸念すべき問題ではありません。

ログのモニター

クラスター内で発生する問題は、情報がキャッシュされること、およびクラスター化の分散されているという性質により、発生時から数日間（あるいは、数カ月間も）経っても、アプリケーションで表立った症状として現れないことがあります。ただし、元の問題は、多くの場合、IBM MQ エラー・ログ（および z/OS の CHINIT ログ）で報告されます。そのため、それらのログを頻繁に調べて、クラスター化に関連するメッセージが書き込まれていないか確認することが重要です。メッセージを読み取って理解し、必要であれば処置を取ってください。

例えば、クラスターは定期的に情報をリパブリッシュしてクラスター・リソースを再評価しているため、クラスター内のキュー・マネージャーとの通信が途絶えたという結果から、特定のクラスター・リソースが削除されたことを知ることができます。このようなイベントが発生する可能性があるという警告は、メッセージ AMQ9465 または CSQX465I (z/OS システムの場合)によって報告されます。これらのメッセージは、問題を調べる必要があることを示します。

ロード・バランシングに関する特別な考慮事項

キューの複数のインスタンスでクラスターのロード・バランシングが行われる場合は、各インスタンスでコンシューム側アプリケーションがメッセージを処理します。1つまたは複数のコンシューム側アプリケーションが強制終了されたり、メッセージの処理を停止したりすると、クラスター化ではキューのそれらのインスタンスにメッセージが引き続き送信される可能性があります。そのような場合、それらのメッセージは、アプリケーションが再び正常に機能するようになるまで処理されません。そのため、ソリューションではアプリケーションのモニターが重要であり、そのような状況でメッセージが再経路指定されるように処置をとる必要があります。そのようなモニターを自動化するためのメカニズムの例は、[クラスター・キュー・モニターのサンプル・プログラム \(AMQSCLM\)](#) で確認することができます。

関連概念

[377 ページの『分散パブリッシュ/サブスクライブ・ネットワークのチューニング』](#)

このセクションにあるチューニングのヒントを利用して、IBM MQ 分散パブリッシュ/サブスクライブのクラスターおよび階層のパフォーマンスを向上させます。

[383 ページの『パブリッシュ/サブスクライブ・ネットワークにおけるプロデューサーとコンシューマーのバランシング』](#)

非同期メッセージング・パフォーマンスの重要な概念はバランスを取ることです。メッセージ・コンシューマーとメッセージ・プロデューサーとのバランスが取れていない場合、生じる危険として、コンシュームされていないメッセージのバックログが増大し、複数のアプリケーションのパフォーマンスに深刻な影響を及ぼす場合があります。

伝送キュー切り替えのモニター

伝送キューを切り替えるクラスター送信側チャネルのプロセスをモニターして、企業への影響を最小限に抑えることが重要です。例えば、ワークロードが高い場合や、多数のチャネルを同時に切り替える場合には、このプロセスを試行しないでください。

チャネルの切り替えプロセス

チャネルの切り替えに使用されるプロセスは、以下のとおりです。

1. チャネルは、入力用に新しい伝送キューを開き、(関連 ID による取得を使用して) そのキューからメッセージの取得を開始します。

2. バックグラウンド・プロセスは、キュー・マネージャーによって開始され、チャンネルのキューに入れられたメッセージを古い伝送キューから新しい伝送キューに移動します。メッセージが移動されている間、チャンネルの新しいメッセージは、順序付けを保持するために古い伝送キューに入れられます。チャンネルの古い伝送キューに多数のメッセージがある場合、または新しいメッセージが急速に到着している場合は、このプロセスが完了するまでにしばらく時間がかかることがあります。
3. コミットされたメッセージまたはコミットされていないメッセージが、古い伝送キュー上のチャンネルのキューに残っていない場合、切り替えは完了します。新しいメッセージは、新しい伝送キューに直接書き込まれるようになりました。

多数のチャンネルが同時に切り替えられる事態を回避するために、IBM MQ は、実行されていない1つ以上のチャンネルの伝送キューを切り替える機能を提供します。日付:

- IBM MQ for Multiplatforms コマンドは **runswchl** という名前になります。
- IBM MQ for z/OS 代わりに CSQUTIL ユーティリティを使用して、SWITCH CHANNEL コマンドを処理することができます。

スイッチ操作の状況のモニター

スイッチ操作の状況を理解するために、管理者は以下のアクションを実行できます。

- キュー・マネージャーのエラー・ログ (AMQERR01.LOG)。操作中の以下のステージを示すメッセージが出力されます。
 - 切り替え操作が開始されました
 - メッセージの移動が開始されました
 - 移動するメッセージの数に関する定期的な更新 (切り替え操作がすぐに完了しない場合)
 - メッセージの移動が完了しました
 - 切り替え操作が完了しました

z/OS では、これらのメッセージはチャンネル・イニシエーターのジョブ・ログではなく、キュー・マネージャーのジョブ・ログに出力されます。ただし、チャンネルが開始時に切り替えを開始すると、チャンネルによってチャンネル・イニシエーターのジョブ・ログに単一のメッセージが出力されます。

- DISPLAY CLUSQMGR コマンドを使用して、各クラスター送信側チャンネルが現在使用している伝送キューを照会します。
- 照会モードで **runswchl** コマンド (または z/OS の場合は CSQUTIL) を実行して、1つ以上のチャンネルの切り替え状況を確認します。このコマンドの出力は、チャンネルごとに以下を識別します。
 - チャンネルに保留中の切り替え操作があるかどうか
 - チャンネルの切り替え元および切り替え先の伝送キュー
 - 古い伝送キューに残っているメッセージの数

各チャンネルの状況、構成変更の影響、およびすべてのスイッチ操作が完了したかどうかを1回の呼び出しで判別できるため、各コマンドは非常に便利です。

発生する可能性がある潜在的な問題

伝送キューの切り替え時に発生する可能性のあるいくつかの問題、その原因、および最も可能性の高い解決策のリストについては、[伝送キューの切り替え時に発生する可能性のある問題](#) を参照してください。

関連概念

377 ページの『[分散パブリッシュ/サブスクライブ・ネットワークのチューニング](#)』

このセクションにあるチューニングのヒントを利用して、IBM MQ 分散パブリッシュ/サブスクライブのクラスターおよび階層のパフォーマンスを向上させます。

383 ページの『[パブリッシュ/サブスクライブ・ネットワークにおけるプロデューサーとコンシューマーのバランス](#)』

非同期メッセージング・パフォーマンスの重要な概念はバランスを取ることです。メッセージ・コンシューマーとメッセージ・プロデューサーとのバランスが取れていない場合、生じうる危険として、コンシュー

ームされていないメッセージのバックログが増大し、複数のアプリケーションのパフォーマンスに深刻な影響を及ぼす場合があります。

Multi アプリケーション・バランシングのモニター

DISPLAY APSTATUS コマンドを使用して、均一クラスター全体のアプリケーション・バランシングの状態を監視し、これが予期しないものである場合にアプリケーションのバランスが取られない理由を調べることができます。

クラスター内のキュー・マネージャー全体でアプリケーションの現在の状態をモニターする

DIS APSTATUS コマンドを実行することにより、均一クラスターのキュー・マネージャーから、クラスターのすべてのキュー・マネージャーわたってアプリケーションの現在の状態の概要を取得できます。

IBM MQ 9.2.0 以降、**TYPE** フィールドも出力に表示されます。

例えば、キュー・マネージャーが起動した直後に、次のような出力が表示される場合があります。

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(UNKNOWN)
TYPE (APPL)
```

これは、均一クラスターに MYAPP というアプリケーションが存在し、現在、8つのインスタンスがあること、および8つすべてが均一クラスターで移動可能であると見なされていることを示しています。

UNKNOWN の BALANCED 値は一時値です。これは、キュー・マネージャーが、必要に応じたアプリケーションの再バランスをまだしていないことを示します。

しばらくすると、以下の出力が表示されるはずです。

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCCOUNT(8)
BALANCED(NO)
TYPE (APPL)
```

この出力は、アプリケーションに8つのインスタンスが含まれているものの、均一クラスター全体でバランスが取られていないことを示しています。この時点で、クラスター内のアプリケーションの分布を調べることをお勧めします。

これを行うには、**DIS APSTATUS** コマンドを再度実行します。このコマンドは、均一クラスター内の任意のキュー・マネージャーに対して実行できることに注意してください。

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(6)                      MOVCCOUNT(6)
BALSTATE(HIGH)                 LMSGDATE(2019-05-24)
LMSGTIME(13:11:10)             QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                      MOVCCOUNT(1)
BALSTATE(LOW)                  LMSGDATE(2019-05-24)
LMSGTIME(13:11:03)             QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                      MOVCCOUNT(1)
BALSTATE(LOW)                  LMSGDATE(2019-05-24)
LMSGTIME(13:11:07)             QMNAME(UNID003)
```



```
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)
```

この時点で、キュー・マネージャー UNID001 には 6 つのインスタンスがありますが、キュー・マネージャー UNID002 と UNID003 にはそれぞれ 1 つずつしかありません。BALSTATE 出力は、キュー・マネージャーが最後に報告した時点でのバランス状態を示します。一方、インスタンス数は BALSTATE フィールドよりも新しい場合があることに注意してください。

また、この出力は、均一クラスターがこのアプリケーションに関する情報を分散していることも示しています。アプリケーションは、均一クラスターのすべてのキュー・マネージャー・インスタンスに認識され、最終メッセージの日時は最新のものです。

また、ACTIVE フィールドは、クラスター内のすべてのキュー・マネージャーが相互に通信していることと見なされていることを示します。いずれかのキュー・マネージャーで ACTIVE が NO に設定されている場合、そのキュー・マネージャーとの定期的な通信が中断されていることを示します。

インスタンス自体の再バランスが取られるまでそのままにしておくと、出力は最終的に次のようなものになります。

```
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)
```

キュー・マネージャーの BALSTATE が OK になっています。これは、定常状態に達したことを示しています。

個々のアプリケーション・インスタンスのモニター

個々のアプリケーション・インスタンスを参照できますが、これはキュー・マネージャーごとに行います。UNID001 の出力を参照する場合:

```
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.
```

MOVABLE(YES) が存在する場合、必要に応じてクラスター内の別のキュー・マネージャーにこのインスタンスを移動できることを示します。次の例では、アプリケーション・インスタンスはクライアントとして接続されていないため、移動できません。

```
3 : DISPLAY APSTATUS('ServerApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ServerApp)
CONNNTAG(MQCT02DFE75C01800B20UNID001_2019-05-24_13.09.35ServerApp)
CONNS(1) IMMREASN(NOTCLIENT)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)
```

CONNNTAG フィールドを使用して、個々のキュー・マネージャーの接続をそのインスタンスから参照できます。これは、アプリケーション・インスタンスが IMMREASN(NOTRECONN) である場合に役立ちます。次の例では、クライアント・アプリケーションは移動不可であり、接続オプションを調べると、**CONNOPTS** フィールドに MQCNO_RECONNECT がいないことがわかります。

```
1 : DISPLAY APSTATUS('ClientApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ClientApp)
CONNNTAG(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(CONNNTAG 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)
CONNNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
TYPE (CONN)
```

関連概念

[自動アプリケーション・バランシング](#)

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 326](#) 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 321](#)
 - [“Using IBM MQ events” on page 323](#)
- z/OS service aids:
 - [“Using System Management Facility” on page 323](#)
- 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 327](#).

Information about interpreting the data gathered by the accounting trace is given in [“Interpreting IBM MQ for z/OS accounting data” on page 363](#).

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	
DISPLAY CLUSQMGR	Inquire Cluster Queue Manager

MQSC command	PCF command
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 323.

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 323](#) 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 ネットワークのモニター” 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 “イベント・タイプ” 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 325.

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 363.

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 331](#)).

SMF タイプ 115 およびタイプ 116 レコードを収集するために必要なコマンド

このトピックでは、タイプ 115 およびタイプ 116 の SMF レコードを収集するために必要なコマンドについて説明します。

START TRACE コマンドの使用

レコードを収集するには、以下のオプションを指定して動的バージョンの [START TRACE](#) コマンドを使用します。

- `START TRACE(STAT) DEST SMF CLASS(*)` および `START TRACE(ACCTG) DEST SMF CLASS(*)` は、クラス 1 から 3 のトレースを開始します。
- `START TRACE(STAT) DEST SMF CLASS(4)` および `START TRACE(ACCTG) DEST SMF CLASS(4)` は、それぞれチャンネル・イニシエーター統計およびチャンネル・アカウントティング・データを開始します。
クラス 4 チャンネル・イニシエーター情報については、[チャンネル・イニシエーター SMF データの計画](#)を参照してください。
- `START TRACE(STAT) DEST SMF CLASS(5)` はキュー統計を開始します。



重要: コンソール・バージョンのコマンドを使用している場合は、発行するコマンドの先頭に `cpf` を追加します。

CLASS (4) チャンネル統計の場合、CHANNEL 定義で STATCHL 属性を設定する必要があります。詳細については、[145 ページの『チャンネル統計収集の制御』](#)を参照してください。

CLASS (5) キュー統計の場合、キュー定義または QMGR 定義 (あるいはその両方) に STATQ 属性を設定する必要があります。詳細については、[144 ページの『キュー統計収集の制御』](#)を参照してください。

SMF タイプ 115 レコード- IBM MQ 統計

現在の統計収集を確認するには、[DISPLAY TRACE](#) コマンド DISPLAY TRACE (STAT) を発行します。次のように表示されます。

```
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
```

注: ZPARM SMFSTAT または ZPARM SMFACCT 属性をアスタリスクに設定すると、クラス 1、2、および 3 のみが制御されるようになりました。START TRACE コマンドを使用して、クラス 4 および 5 をオンにする必要があります。

SMF タイプ 116 クラス 3 & 4 データ- IBM MQ タスクおよびチャンネル・アカウントिंग・レコード

現在の統計収集を確認するには、[DISPLAY TRACE](#) コマンド DISPLAY TRACE (ACCT) を発行します。次のように表示されます。

```
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
```

以下のコマンドを使用して、必要なキュー・マネージャーのサイクルがないため、SMF116 データ収集を動的にオン/オフにすることができます。

```
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)
```

チャンネル・アカウントिंग・データには、SMF を作成するための追加のセットアップが必要です。手順は以下のとおりです。

- 以下のコマンドを発行して、すべての送信側受信側およびクライアント接続に関する情報の収集をオンにします。

```
ALTER QMGR STATCHL (HIGH)
```

- クラスタリングが使用中の場合は、以下のコマンドを発行して、自動定義クラスター・チャンネルの情報の収集をオンにします。

```
ALTER QMGR STATCLS (HIGH)
```

注: 上記のステートメントでは、HIGH、MED、または LOW の値は同じ効果があります。

- すべてのチャンネルで STATCHL が QMGR に設定されていることを確認します。

```
DISPLAY CHANNEL (*) STATCHL
```

STATIME の検証



重要:最新のツールに準拠するには、`SET SYSTEM` コマンドを使用して間隔を 5 分以下に設定する必要があります。ただし、デフォルトは 30 分です。

キュー共有グループ (QSG) を評価する場合は、STATIME が QSG 全体で整合していることを確認してください。

- 以下の `DISPLAY SYSTEM` コマンドを使用して、システム設定を表示します。

```
DISPLAY SYSTEM
```

- STATIME 値がゼロの場合 (統計間隔がデフォルトの LPAR 値に設定されていることを意味します)、これは通常は問題ありません。

非常に大容量の LPAR のインスタンスがあり、SMF データ生成の所要時間が変化しています。データの評価で期間が大きく異なる場合は、すべてのキュー・マネージャーに対して STATIME をゼロ以外の値に設定します。

- 以下の例では、STATIME 間隔を 5 分に設定します。

```
SET SYSTEM STATIME(05)
```

IBM MQ 9.2.4 以上の場合は、以下のようになります。

```
SET SYSTEM STATIME(05.00)
```

注:新しい間隔は、現在の間隔が経過するまで有効にならないため、データ収集を開始する前に変更を行う必要があります。

リリース・レベル IBM MQ 9.2.4 以上のキュー・マネージャーの ACCTIME 値の検証:

- ACCTIME が -1 に設定されている場合、STATIME 属性と同じです。
- この評価が QSG に関するものである場合は、ACCTIME が QSG 全体で整合していることを確認してください。そうでない場合は、データの収集を開始する前に、値を同じ間隔に設定してください。
- ACCTIME が 30 以上に設定されている場合は、`SET SYSTEM` コマンドを使用して、ACCTIME を 15 以下に変更してください。以下の例では、ACCTIME 間隔を 5 分に設定します。

```
SET SYSTEM ACCTIME(05.00)
```

注:新しい間隔は、現在の間隔が経過するまで有効にならないため、データ収集を開始する前に変更を行う必要があります。

キュー・マネージャーでの ACCTQ 設定の検証

- 次のコマンドを使用して、調査しているキュー・マネージャーの ACCTQ 設定を表示します。

```
DISPLAY QMGR ACCTQ
```

- 値が ACCTQ (ON) の場合、これ以上のアクションは不要です。それ以外の場合は、次のコマンドを発行します。

```
ALTER QMGR ACCTQ(ON)
```

- クラスタリングが使用中の場合は、`SYSTEM.CLUSTER.TRANSMIT.QUEUE` およびその他の指定されたクラスターは、キュー・マネージャーによってホストされるキューを送信します。クラスター伝送キューに対してアカウントングが使用可能かどうかを判別するには、次のようにします。

– 以下のコマンドを使用して、キューの ACCTQ 設定を表示します。

```
DISPLAY QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ
```

- 値が ACCTQ (ON) の場合、これ以上のアクションは不要です。それ以外の場合は、次のコマンドを発行します。

```
ALTER QL (SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ(ON)
```

- その他のすべての大量キューの場合は、ACCTQ の値が ON であることを確認するか、ACCTG を QMGR に設定します。

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 331](#)

Use this topic as a reference for the format of the SMF header.

[“Self-defining sections” on page 332](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 333](#)

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 331 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
6	6	Integer	4	SM115TM E	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)

Table 34. SMF record 115 header description (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
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	SM115EN D	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 332](#) 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
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

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	
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:

```
sssssssllllnnnn
```

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 333.

Note: Always use offsets in the self-defining sections to locate the statistics records.

z/OS Examples of SMF statistics records

Use this topic to understand some example SMF records.

Figure 20 on page 333 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	<u>02000000</u>	<u>5E730035</u>	<u>55750100</u>	<u>223FD4E5</u>	*.....;.....MV*
000010	<u>F4F1D4D8</u>	<u>F0F70001</u>	<u>F6F0F000</u>	<u>000001DC</u>	*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 334 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.

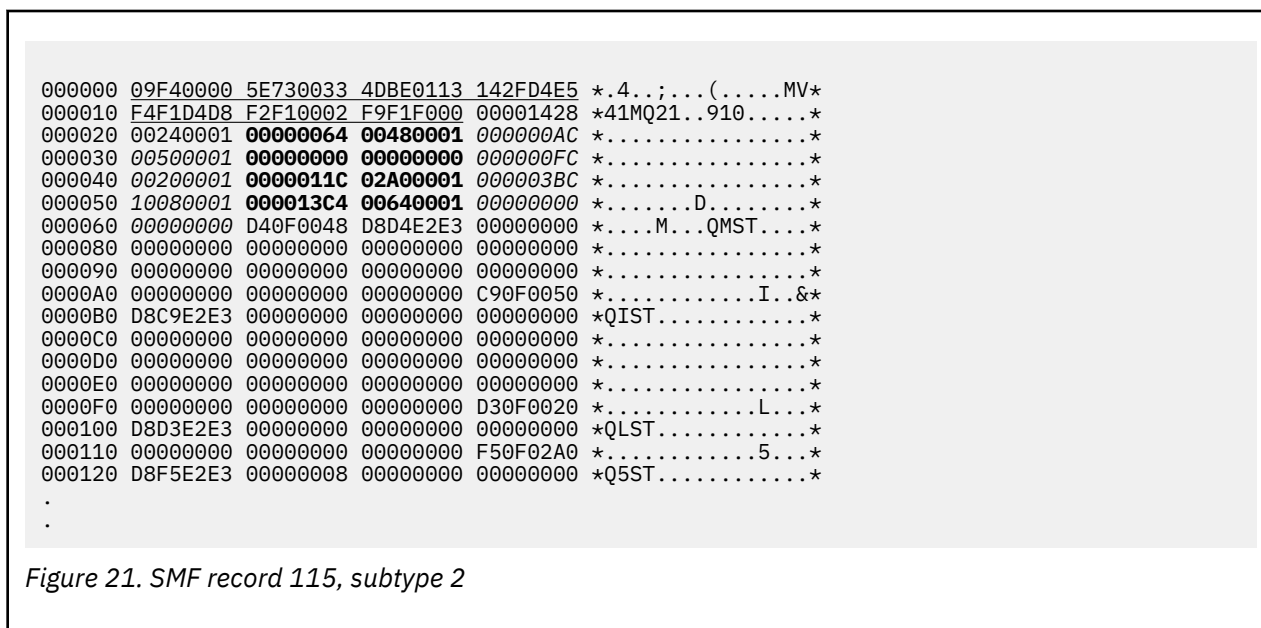


Figure 21. SMF record 115, subtype 2

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 323](#).

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 thlqual.SCSQMACS(CSQDQSRH).

ログ管理プログラム・データ・レコード

このトピックでは、ログ管理プログラム・データ・レコードの形式について説明します。

ログ・マネージャー統計レコードのフォーマットは、アセンブラー・マクロ thlqual.SCSQMACS(CSQDQJST)に記述されています。

統計では、以下のカウントは重要です。

1. ログ書き込み要求の合計数:

$$N_{\text{logwrite}} = \text{QJSTWRNW} + \text{QJSTWRF}$$

2. ログ読み取り要求の合計数:

$$N_{\text{logread}} = \text{QJSTRBUF} + \text{QJSTRACT} + \text{QJSTRARH}$$

以下の表に、ログ管理プログラム統計を使用して調査できる問題の症状を示します。

症状 1

QJSTWTB がゼロでない。

理由

ストレージ内バッファが活動ログに書き込まれている間、タスクが延期されています。

アクティブ・ログへの書き込みに関する問題が生じている可能性があります。

CSQ6LOGP 内の OUTBUFF パラメーターが小さすぎます。

アクション

活動ログへの書き込みに関する問題を調査します。

CSQ6LOGP 内の OUTBUFF パラメーターの値を増やします。

症状 2

比率 $QJSTWTL/N_{logread}$ が 1% を超えています。

理由

アーカイブ・ログから読み取りを行うためのログ読み取りが開始されましたが、MAXRTU で指定された数のデータ・セットがすでに割り振られていたため、IBM MQ はデータ・セットを割り振ることができませんでした。

アクション

MAXRTU を大きくします。

症状 3

比率 $QJSTRARH/N_{logread}$ が通常より大きくなっています。

理由

大部分のログ読み取り要求は、出力バッファまたは活動ログからのものになるはずですが、バックアウトの要求を満たすため、リカバリー単位レコードは、ストレージ内バッファ、活動ログ、および保存ログから読み取られます。

実行時間の長い(数分間に及ぶ)リカバリー単位では、ログ・レコードが多く異なるログに分散されることがあります。これにより、ログ・レコードの回復に余分な作業が必要となるため、パフォーマンスが低下します。

アクション

リカバリー単位の長さを減らすように、アプリケーションを変更します。また、1つのリカバリー単位が複数のログに分散される可能性を減らすために、アクティブ・ログのサイズを増やすことを検討してください。

その他のヒント

$N_{logread} / N_{logwrite}$ の比率は、バックアウトする必要がある作業の量を示します。

症状 4

QJSTLLCP が 1 時間あたり 10 を超える。

理由

使用頻度の高いシステムでは、通常、1 時間あたり 10 回のチェックポイントの発生が予期されます。QJSTLLCP 値がこれよりも大きい場合は、キュー・マネージャーのセットアップに問題があることを示しています。

考えられる原因は、CSQ6SYSP 内の LOGLOAD パラメーターが小さすぎることです。チェックポイントの原因となるその他のイベントとして、アクティブ・ログが満杯になり、次のアクティブ・ログ・データ・セットに切り替わるときがあります。使用しているログが小さすぎると、頻繁にチェックポイントが発生する可能性があります。

QJSTLLCP は、チェックポイントの総数のカウントです。

アクション

必要に応じて、LOGLOAD パラメーターを大きくするか、またはログ・データ・セットのサイズを大きくします。

症状 5

QJSTCmpFail が 0 より大きい、QJSTCmpComp が QJSTCmpUncmp に比べてあまり小さくない。

理由

キュー・マネージャーは、ログ・レコードを大幅に圧縮することができません。

QJSTCmpFail は、キュー・マネージャーがレコード長をまったく縮小できなかった回数です。この数値を QJSTCmpReq (圧縮要求の数) と比較して、失敗の回数が著しく多いものかどうかを調べる必要があります。

QJSTCmpComp はログに書き込まれる圧縮バイト数の合計で、QJSTCmpUncmp は圧縮前の合計バイト数です。どちらの合計にも、圧縮に適さなかったログ・レコードに関して書き込まれるバイト数は含まれません。数値が類似している場合は、圧縮の効果がほとんどなかったということです。

アクション

ログ圧縮をオフにします。SET LOG COMPLOG(NONE) コマンドを発行します。詳しくは、[SET LOG](#) コマンドを参照してください。

注: システムの始動後に最初に生成される統計のセットでは、未完了リカバリー単位の解決による大量のログ・アクティビティーが示される場合があります。

zHyper 書き込みの変更点

V 9.4.0

IBM MQ 9.3.5 以降、zHyper 書き込み処理は、QJSTHWC および QJSTHWE 統計の動作を変更します。

IBM MQ 9.3.5 より前:

- QJSTHWC は、SMF インターバルで使用されたログ・データ・セットのうち、zHyper 書き込み可能ボリューム上にあるログ・データ・セットの数です。ボリュームの zHyper 書き込み機能は、キュー・マネージャーの始動時に取得され、時間の経過とともに変化する可能性があるため、この情報は失効している可能性があります。
- QJSTHWE は、SMF インターバルで使用され、zHyperWrite を使用可能にして書き込まれたログ・データ・セットの数です。これは、ログ・データ・セットが zHyper 書き込み可能ボリューム上にあり、ZHYWRITE (YES) を設定することによってキュー・マネージャーが zHyper 書き込み可能になっている場合に起こります。

IBM MQ 9.3.5 以降:

- QJSTHWC は、SMF インターバルで使用されたログ・データ・セットのうち、zHyper 書き込み可能ボリューム上にあるログ・データ・セットの数です。ボリュームの zHyper 書き込み機能は、キュー・マネージャーの始動時に取得され、時間の経過とともに変化する可能性があるため、この情報は失効している可能性があります。
- QJSTHWE は、SMF インターバルで使用され、zHyperWrite を使用可能にして書き込まれたログ・データ・セットの数です。これは、ZHYWRITE (YES) を設定してキュー・マネージャーで zHyper 書き込みが有効になっている場合に発生します。



重要: ボリュームの zHyper 書き込み機能は、時間の経過とともに変化する可能性があります。IBM MQ 9.3.5 以降、QJSTHWE が QJSTHWC より大きいシナリオが発生する可能性があります。

zHyper リンク統計

V 9.4.0

IBM MQ 9.4.0 以降では、zHyper リンクのパフォーマンスを追跡するために、ログ・マネージャー統計が追加されました。

以下の統計が QJST に追加されました。

名前	タイプ	長さバイト	説明
QJSTHLSCIW	符号なし整数	4	zHyper リンクを要求した単一ページ書き込みの数。

名前	タイプ	長さバイト	説明
QJSTHLMCIW	符号なし整数	4	zHyper リンクを要求した複数ページ書き込みの数。
QJSTHLCICNTW	符号なし整数	4	書き込みが zHyper リンクを要求した場合に書き込まれたページの数。
QJSTHLIS (QJSTHLSCIS)	符号なし整数	4	zHyper リンクを正常に使用したページ書き込みの数。つまり、書き込みは同期的に行われました。
QJSTHLMCIS	符号なし整数	4	zHyper リンクを正常に使用した複数ページ書き込みの数。つまり、書き込みは同期的に行われました。
QJSTHLCICNTS	符号なし整数	4	zHyper リンクを使用して正常に書き込まれたページの数。つまり、書き込みは同期的に行われました。
QJSTHLIF (QJSTHLSCIF)	符号なし整数	4	zHyper リンクを使用しようとしたが、書き込みを同期的に行うことができなかった単一ページ書き込みの数。代わりに、書き込みは非同期で正常に行われました。
QJSTHLMCIF	符号なし整数	4	zHyper リンクを使用しようとしたが、書き込みを同期的に行うことができなかった複数ページ書き込みの数。代わりに、書き込みは非同期で正常に行われました。
QJSTHLCICNTF	符号なし整数	4	書き込みで zHyper リンクを使用しようとしたが、書き込みを同期的に行うことができなかった書き込みページの数。代わりに、書き込みは非同期で正常に行われました。
QJSTHLSCICONT	符号なし整数	4	zHyper リンク書き込みセッションが確立される前の単一ページ書き込みの数。 339 ページの『1』 を参照してください。
QJSTHLMCICONT	符号なし整数	4	zHyper リンク書き込みセッションが確立される前の複数ページ書き込みの数。 339 ページの『1』 を参照してください。
QJSTHLCONCICNT	符号なし整数	4	zHyper リンク接続が確立される前に書き込まれたページ数。 339 ページの『1』 を参照してください。
QJSTHLSCIWTM AX	符号なし整数	8	インターバルにおける最長の zHyper リンク書き込み時間 (単一ページ)。
QJSTHLSCIWTM IN	符号なし整数	8	間隔内の最短の zHyper リンク書き込み時間 (単一ページ)。
QJSTHLSCIWTT OT	符号なし整数	8	間隔内の zHyper リンク書き込み時間の合計 (単一ページ)。
QJSTHLMCIWT MAX	符号なし整数	8	インターバルにおける最長の zHyper リンク書き込み時間 (複数ページ)。
QJSTHLMCIWT MIN	符号なし整数	8	間隔内の最短の zHyper リンク書き込み時間 (複数ページ)。
QJSTHLMCIWTT OT	符号なし整数	8	間隔内の zHyper リンク書き込み時間の合計 (複数ページ)。

名前	タイプ	長さバイト	説明
QJSSTHREDIS	符号なし整数	16	単一ページ書き込みの場合の、zHyper リンク書き込み時間の二乗の合計。
QJSTHLC (言語)	符号なし整数	4	この SMF インターバルで使用された、zHyper リンク対応の新規ログの数。
QJSTHLE	符号なし整数	4	この SMF 間隔で使用され、zHyper リンクが使用可能になっている新規ログの数。

注:

1. 新しいアクティブ・ログ・コピーに切り替えるときに、DASD との zHyper リンク書き込みセッションが確立されるまでに時間がかかる場合があります。

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.

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.

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 331](#)

Use this topic as a reference for the format of the SMF header.

[“Self-defining sections” on page 332](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 333](#)

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 341](#).

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

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
16	10	Integer	4	CONNCNT	Number of connect requests
20	14	Integer	4	DISCCNT	Number of disconnect requests
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	UPDCNT	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

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
208	D0	Character	8	UPDTTMXW	Maximum STCK difference - Thread update
216	D8	Character	8	UPDTSCUW	Cumulative STCK difference - SQL update
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

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
480	1E0	Character	8	SSKDSMXW	Maximum STCK difference - SQL delete
488	1E8	Integer	4	LMSSEL	Number of Db2 BLOB read requests
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	LMSDTCUW	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	LMSDSCUW	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	LMSLTCUW	Total elapsed time for all thread list BLOB requests

Offset: Dec	Offset: Hex	Type	Len	Name	Description
644	284	64 bit integer	8	LMSLTMXW	Maximum elapsed time for a thread list BLOB request
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

Table 37. Coupling facility statistics record (QEST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
20	14	Integer	4	QESTSTRN	Structure number
24	18	Integer	4	QESTCSEC	Number of IXLLSTE calls
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.

Table 38. Topic manager statistics record (QTST)

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

Table 38. Topic manager statistics record (QTST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
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)
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 349 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)
4	4	Integer	2	QWSX0PSL	Length of the QWHS
6	6	Integer	2	QWSX0PSN	Number of instances of QWHS
8	8	Integer	4	QWSX0R1O	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	QWSX0R2O	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	QWSX0R3O	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	QWSX0R4O	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	QWSX0R5O	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

Count	First record	Last record
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
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. 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
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 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 351.
- 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 352
- [“Adapter tasks”](#) on page 353
- [“Domain Name Server \(DNS\) task”](#) on page 354
- [“SSL tasks”](#) on page 355

Each task includes:

- The elapsed time that the task spent processing requests in the interval (*qctel_{tm}*)
- 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 (*qctcpt_m*)
- The total wait time of this task in the interval (*qctwtt_m*)
- The number of requests in the interval (*qctreq_n*)

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 (*qctlg_{du}*, *qctls_{du}*) and the time of day when this occurred (*qctlg_{dm}*, *qctls_{dm}*) 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 (*qctcpt_m*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qctel_{tm}*) 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 `th1qua1.SCSQC370 (CSQDSMFC)`. Note that the field names in C are all in lowercase, for example, *qcct_{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 (*qcctj_{obn}*)
- QSG name if it is in a queue sharing group (*qcctq_{sgn}*)
- Peak number used of current channels (*qcctn_{occ}*)
- Peak number used of active channels (*qcctn_{oac}*)
- MAXCHL - maximum permitted current channels (*qcctm_{xcc}*)
- ACTCHL - maximum permitted active channels (*qcctm_{xac}*)
- TCPCHL - maximum permitted TCP/IP channels (*qcctm_{xtp}*)
- LU62CHL - maximum permitted LU62 channels (*qcctm_{xlu}*)
-  31-bit storage used by CHINIT in the extended private region (*qcctst_{us}*). This information is also provided by the CSQX004I message in the CHINIT job log.
-  64-bit storage limit available to the CHINIT (*qcctsl_{im}*)
-  64-bit storage used by CHINIT (*qcctst_{ab}*). 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 `th1qua1.SCSQC370 (CSQDSMFC)`, and in assembler macro `th1qua1.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:

V 9.4.0

```
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 trending towards the maximum number of channels, you might need to increase the maximum number of channels.

z/OS 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 `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.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 (*qctelmtm*) 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 %: *qctelmtm* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn*
- Average ET: *qctelmtm* 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 *thlqual.SCSQC370* (CSQDSMFC), and in assembler macro *thlqual.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 %: *qctelmtm* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn* average
- ET: *qctelmtm* 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 (*qctelmt*) 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 *thlqual1.SCSQC370* (CSQDSMFC), and in assembler macro *thlqual1.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 %: *qctelmt* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn*
- Average ET: *qctelmt* 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 `thlqual.SCSQC370` (CSQDSMFC), and in assembler macro `thlqual.SCSQMACS` (CSQDQCTA).

Example data

Task date,	Type,	Requests,	Busy %,	CPU used,	CPU %,	"avg CPU",	"avg ET",	longest,
		time		Seconds,		uSeconds,	uSeconds,	uSeconds,
0,	SSL,	3112,	1.2,	0.248538,	0.3,	80,	362,	8864,
12:46:40.237697								2014/03/18,
1,	SSL,	3070,	1.2,	0.245433,	0.3,	80,	359,	4714,
12:46:18.938022								2014/03/18,
2,	SSL,	3170,	1.2,	0.255557,	0.3,	81,	362,	7273,
12:46:35.358145								2014/03/18,
3,	SSL,	3060,	1.2,	0.246542,	0.3,	81,	365,	13164,
12:46:44.514045								2014/03/18,
4,	SSL,	3120,	1.3,	0.251927,	0.3,	81,	373,	22438,
12:46:22.134123								2014/03/18,
Summ,	SSL,	15532,	1.2,	1.247998,	0.3,	80,	364,	22438,
12:46:22.134123								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: *qctlsdu* 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 (*qctelmt*) value only includes time while processing requests. This means that the CPU time might be greater than the elapsed time.

Related reference

[ALTER QMGR](#)

V9.4.0 z/OS 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.

QQSTIPLO – 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 364 shows the format of SMF record header (SM116).

Offset : 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)

Table 42. SMF record header description (continued)

Offset : Dec	Offset: Hex	Type	Len	Name	Description	Example
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 366](#).

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 365](#) 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 367
0	Message manager	44	X'2C'	“Message manager data records” on page 369
1	Thread identification record	36	X'24'	“Thread-level and queue-level data records” on page 370
1	Thread-level accounting	44	X'2C'	“Thread-level and queue-level data records” on page 370
1	Queue-level accounting	52	X'34'	“Thread-level and queue-level data records” on page 370 . This section is present only if the WTASWQCT field in the task-related information (WTAS) structure is non-zero.

Table 43. Offsets to self-defining sections (continued)

Record subtype (SMF116STF)	Source of accounting data	Offset of self-defining section		See...
		Dec	Hex	
2	Thread identification record	36	X'24'	"Thread-level and queue-level data records" on page 370
2	Queue-level accounting	44	X'2C'	"Thread-level and queue-level data records" on page 370
10	Channel accounting			"Channel accounting data records" on page 373

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:

sssssss

Fullword containing the offset from start of the SMF record.

lll

Halfword giving the length of this data record.

nnnn

Halfword giving the number of data records in this SMF record.

Figure 22 on page 366 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 364 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 365) is shown in **bold**.

```
000000 01A40000 5E740035 61240100 223FD4E5 *...;.../.....MV*
000000 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600....*
000000 00700001 00000054 00B00001 00000104 *.....*
000000 00300001 00000000 00000000 00000000 *.....*
000000 00000000 00000000 00000000 00000000 *.....*
```

Figure 22. Part of an SMF record 116 showing the header and self-defining sections

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.

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 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 323](#).

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=IFASMFDL,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 368](#) 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, qwhnsda.

The QWHS data includes the subsystem name. For subtype 1 records, it also shows whether there are queue-level accounting records present. If the QWHNSDA 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.

Table 44. Structure of the common IBM MQ SMF header record QWHS

Offset: Dec	Offset : 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	QWHSSSID	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
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 369](#)
- IMS connections (QWHCATYP=3 or 4) - see [Table 46 on page 369](#)
- 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 370 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`, `wtasshex`.

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 371 and Figure 25 on page 372 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 372 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 372 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 371.

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 373 shows the format of the self-defining section.

Table 48. Structure of the channel accounting self-defining section

Offset: Dec	Offset : 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 `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.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
QWHSSTMFC	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 `th1qua1.SCSQMACS(CSQDQCST)`. The format is also described in the C programming language header file `th1qua1.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 (`qcstcbd1`)

- Channel batch interval (*qcstcbit*)
- Channel batch size (*qcstcbisz*)
- 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 (*qcstnpgm*)
- Number of put retries (*qcstprrc*)
- 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 ネットワークのチューニング

このセクションにあるチューニングのヒントを使用して、キュー・マネージャー・ネットワークのパフォーマンスを向上させます。

クライアントとサーバーの接続チャンネルのチューニング

SHARECNV のデフォルト設定は 10 です。これは、チャンネル・インスタンスごとに最大 10 個のクライアント会話を許可します。ただし、異なる数の共有会話を使用すると、パフォーマンスが向上する可能性があります。共有会話が必要ない場合、または分散サーバーを使用している場合は、**SHARECNV** を 1 に設定します。**SHARECNV** を 1 以上に設定したときに、正しく実行されない既存のクライアント・アプリケーションがある場合は、**SHARECNV** を 0 に設定します。

このタスクについて

一部の構成では、共有会話を使用することにより、非常に大きな利点がもたらされます。ただし、分散サーバーでは、デフォルト構成である 10 個の共有会話を使用するチャンネルでのメッセージの処理は、共有会話を使用しないチャンネルよりも平均で 15% 遅くなります。会話を共有している MQI チャンネル・インスタンスでは、1 つのソケット上の会話はすべて同一のスレッドによって受信されます。ソケットを共有する会話がすべてビジー状態の場合、各会話型スレッドは受信スレッドを使用するために互いに競合します。

この競合によって遅延が発生するため、このような状況では、共有会話の数を少なくした方がパフォーマンスが良くなります。

SHARECNV パラメーターを使用して、特定の TCP/IP クライアント・チャンネル・インスタンスで共有される会話の最大数を指定します。可能なすべての値について詳しくは、[サポートされる IBM MQ クライアント: クライアント接続チャンネルおよびサーバー接続チャンネルのデフォルトの動作を参照してください](#)。

SHARECNV を 1 以上に設定すると、以下のパフォーマンス拡張が有効になります。

- 双方向ハートビート
- 管理者の停止と静止
- 先読み
- クライアント・アプリケーションによる非同期コンシューム

共有会話が必要ない場合は、以下の 2 つの設定を使用するとパフォーマンスが最高になります。

- **SHARECNV(1)**.
- **SHARECNV(0)**.

注:

- クライアント接続の **SHARECNV** 値がサーバー接続の **SHARECNV** 値と一致しない場合、最も低い値が使用されます。
- アプリケーションが再入不可ライブラリーに対してリンクまたはコンパイルされると、**CLNTCONN** および **SVRCONN** により高い値が設定されている場合でも、**CURSHCNV(0)** 値がネゴシエーションされます。

特定のチャンネル・インスタンスのパフォーマンスを最適化するには、以下のいずれかのステップを実行します。

手順

- デフォルトの **SHARECNV** 値 **10** を使用するチャンネルをモニターします。

SHARECNV(10) のデフォルト設定は、多くのシナリオでうまく機能しますが、特定のチャンネル・インスタンスに最適な設定ではない可能性があります。例えば、分散サーバーの場合、この設定を使用するチャンネル上のメッセージの処理は、共有会話を使用しないチャンネル上の処理より平均 15% 遅くなります。

特定のチャンネル・インスタンスに対してデフォルト設定が適切であることを確認するには、この設定を使用してチャンネルのパフォーマンスをモニターします。

- 2 以上の **SHARECNV** 値を設定します。

SHARECNV(2) を **SHARECNV(999999999)** に設定できます。選択した設定が特定のチャンネル・インスタンスに対して適切であることを確認するには、新しい設定を使用してチャンネルのパフォーマンスをモニターします。

- 1 の **SHARECNV** 値を設定します。

共有会話が必要ない場合は、可能な限りこの設定を使用してください。これにより、受信スレッドを使用するための競合が解消され、クライアント・アプリケーションは、「このタスクについて」セクションで説明されているパフォーマンスの機能拡張を利用できます。

この設定により、分散サーバーのパフォーマンスが大幅に向上します。パフォーマンスの向上は、非先読み同期読み取り待機呼び出し (例えば、C クライアント **MQGET** 待機呼び出し) を発行するクライアント・アプリケーションに適用されます。これらのクライアント・アプリケーションに接続するとき、分散サーバーが使用するスレッドおよびメモリーの量は少なく済むため、スループットが向上します。

ソケットを介して会話を共有しているクライアントがサーバーに接続されている場合に、共有会話の設定を **SHARECNV(10)** から **SHARECNV(1)** に引き下げると、次のような影響があります。

- サーバー上でのソケット使用量が増加します。
- サーバー上でのチャンネル・インスタンスが増加します。

この場合は、**MaxChannels** および **MaxActiveChannels** の設定を引き上げることもできます。

注: また、MQCONNX オプション、MQCNO_NO_CONV_SHARING を設定し、SHARECNV が 1 より大きい値に設定されているチャンネルにアプリケーションを接続することもできます。結果は、SHARECNV が 1 に設定されているチャンネルにアプリケーションを接続する場合と同じです。

- SHARECNV 値を 0 に設定します。

チャンネル・インスタンスは、IBM WebSphere MQ 6.0 サーバーまたはクライアント接続チャンネルとまったく同じように動作します。共有会話や、SHARECNV を 1 以上に設定した場合に使用できるパフォーマンスの向上は得られません。SHARECNV を 1 以上に設定すると正常に実行されない既存のクライアント・アプリケーションが存在する場合のみ、値 0 を使用してください。

関連概念

[サポートされる IBM MQ クライアント: クライアント接続チャンネルとサーバー接続チャンネルのデフォルトの動作](#)

分散パブリッシュ/サブスクライブ・ネットワークのチューニング

このセクションにあるチューニングのヒントを利用して、IBM MQ 分散パブリッシュ/サブスクライブのクラスターおよび階層のパフォーマンスを向上させます。

関連概念

314 ページの『クラスターのモニター』

クラスター内で、アプリケーション・メッセージ、制御メッセージ、およびログをモニターできます。キューの複数のインスタンス間でクラスターのロード・バランシングが行われる場合は、モニターに関する特別な考慮事項があります。

直接ルーティング型パブリッシュ/サブスクライブ・クラスターのパフォーマンス

直接ルーティング型パブリッシュ/サブスクライブ・クラスターでは、すべてのクラスター・キュー・マネージャーがパブリッシュ/サブスクライブ・メッセージングにアクティブに関与しているかどうかにかかわらず、クラスター・トピックやプロキシ・サブスクリプションなどの情報がクラスターのすべてのメンバーにプッシュされます。このプロセスによって、システムに多大な負荷が生じる可能性があります。クラスター管理がパフォーマンスに及ぼす影響を軽減するには、オフピーク時に更新を実行したり、ずっと小規模な、パブリッシュ/サブスクライブに関与するキュー・マネージャーのサブセットを定義してそれを「オーバーラップ」クラスターにしたり、トピック・ホスト・ルーティング型の使用に切り替えたりすることができます。

パブリッシュ/サブスクライブ・クラスター内のキュー・マネージャーでのワークロードには、以下の 2 つのソースがあります。

- アプリケーション・プログラムのメッセージの直接的な処理。
- クラスターの管理に必要なメッセージおよびチャンネルの処理。

典型的な Point-to-Point クラスターでは、クラスター・システム・ワークロードは、主に、必要に応じてクラスターのメンバーによって明示的に要求される情報に限定されます。したがって、非常に大規模な Point-to-Point クラスター (何千ものキュー・マネージャーを含むクラスターなど) 以外の場合は、パフォーマンスに及ぼすクラスターの管理の影響の大部分を無視できます。しかし、直接ルーティング型パブリッシュ/サブスクライブ・クラスターでは、すべてのクラスター・キュー・マネージャーがパブリッシュ/サブスクライブ・メッセージングにアクティブに関与しているかどうかにかかわらず、クラスター・トピック、キュー・マネージャー・メンバーシップ、プロキシ・サブスクリプションなどの情報がクラスターのすべてのメンバーにプッシュされます。これにより、システムに多大な負荷がさらに生じる可能性があります。したがって、時間とサイズの両方の点で、キュー・マネージャーのパフォーマンスに及ぼすクラスター管理の影響を考慮する必要があります。

直接ルーティング型クラスターのパフォーマンス特性

中核的な管理作業の観点から、Point-to-Point クラスターと、直接ルーティング型パブリッシュ/サブスクライブ・クラスターを比較します。

最初に、Point-to-Point クラスターです。

1. 新しいクラスター・キューが定義されると、宛先情報が完全リポジトリ・キュー・マネージャーにプッシュされ、他のクラスター・メンバーには、それらのメンバーが最初にクラスター・キューを参照したとき (例えば、アプリケーションがそのクラスター・キューを開こうとしたときなど) にのみ送信されます。この情報は、キューにアクセスするたびに情報をリモートから取得する必要を取り除くために、キュー・マネージャーによってキャッシュに入れられます。
2. クラスターへのキュー・マネージャーの追加は、他のキュー・マネージャーの負荷には直接影響しません。新しいキュー・マネージャーに関する情報は完全リポジトリにプッシュされますが、クラスター内の他のキュー・マネージャーから新しいキュー・マネージャーへのチャンネルが作成され開始されるのは、新しいキュー・マネージャーとのトラフィックが流れ始めてからです。

要約すると、Point-to-Point クラスターでのキュー・マネージャーへの負荷は、アプリケーション・プログラムのために処理されるメッセージ・トラフィックと関係があり、クラスターのサイズとは直接的な関係はありません。

次に、直接ルーティング型パブリッシュ/サブスクライブ・クラスターです。

1. 新しいクラスター・トピックが定義されると、その情報は完全リポジトリ・キュー・マネージャーにプッシュされ、そこからクラスターのすべてのメンバーに直接プッシュされます。その結果、完全リポジトリからクラスターの各メンバーへのチャンネルが開始されます (まだ開始されていない場合)。これが最初の直接クラスター・トピックである場合には、各キュー・マネージャー・メンバーに、クラスター内にある他のすべてのキュー・マネージャー・メンバーに関する情報が送信されます。
2. 新しいトピック・ストリングに関するクラスター・トピックに対するサブスクリプションが作成されると、その情報は、即座にキュー・マネージャーからクラスターの他のすべてのメンバーへと直接プッシュされます。その結果、そのキュー・マネージャーからクラスターの各メンバーへのチャンネルが開始されます (まだ開始されていない場合)。
3. 新しいキュー・マネージャーが既存のクラスターに加わると、すべてのクラスター・トピックに関する情報 (および直接クラスター・トピックが定義されている場合には、すべてのキュー・マネージャー・メンバーに関する情報) が完全リポジトリ・キュー・マネージャーからその新しいキュー・マネージャーにプッシュされます。すると、その新しいキュー・マネージャーは、クラスター内のクラスター・トピックに対するすべてのサブスクリプションに関する情報を、クラスターのすべてのメンバーのものと同期させます。

要約すると、直接ルーティング型パブリッシュ/サブスクライブ・クラスター内のいずれのキュー・マネージャーでも、クラスター管理の負荷は、そのクラスター内のキュー・マネージャーの数、クラスター・トピックの数、および各種トピック・ストリングに関するサブスクリプションの変更の数に応じて増加します。各キュー・マネージャーでそれらのクラスター・トピックがローカルに使用されるかどうかは関係ありません。

大規模なクラスター、またはサブスクリプションの変更のペースが速いクラスターでは、このレベルのクラスター管理は、すべてのキュー・マネージャーにわたって相当なオーバーヘッドとなる可能性があります。

直接ルーティング型パブリッシュ/サブスクライブのパフォーマンスへの影響の軽減

直接ルーティング型パブリッシュ/サブスクライブ・クラスターのパフォーマンスに及ぼすクラスター管理の影響を軽減するには、以下のオプションを考慮してください。

- クラスター、トピック、およびサブスクリプションの更新を、一日の内のオフピーク時に行う。
- ずっと小規模な、パブリッシュ/サブスクライブに関与するキュー・マネージャーのサブセットを定義して、それを「オーバーラップ」クラスターにする。そのようにすると、このクラスターはクラスター・トピックが定義されるクラスターになります。いくつかのキュー・マネージャーが2つのクラスターに置かれることとなりますが、パブリッシュ/サブスクライブの全体的な影響は以下のように軽減されます。
 - パブリッシュ/サブスクライブ・クラスターのサイズが小さくなります。
 - パブリッシュ/サブスクライブ・クラスターにないキュー・マネージャーが受ける、クラスター管理トラフィックの影響がかなり小さくなります。

上記のオプションによってパフォーマンスの問題が十分に解決されない場合には、代わりにトピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターの使用を検討してください。パブリッ

キュー/サブスクライブ・クラスターにおける直接ルーティング型とトピック・ホスト・ルーティング型の比較について詳しくは、[パブリッシュ/サブスクライブ・クラスターの設計](#)を参照してください。

関連概念

[トピック・ホストでルーティングされるパブリッシュ/サブスクライブ・クラスターのパフォーマンス](#)
トピック・ホストでルーティングされるパブリッシュ/サブスクライブ・クラスターを使用すると、各トピックをどのキュー・マネージャーがホストするかを正確に制御できます。これらのトピック・ホストが、そのトピック・ツリー・ブランチのルーティング・キュー・マネージャーになります。さらに、サブスクリプションもパブリッシャーもないキュー・マネージャーは、トピック・ホストに接続する必要はありません。この構成では、クラスター内のキュー・マネージャー間の接続の数、およびキュー・マネージャー間で渡される情報の量がかなり少なくなる可能性があります。

[パブリッシュ/サブスクライブ・ネットワークにおけるプロデューサーとコンシューマーのバランシング](#)
非同期メッセージング・パフォーマンスの重要な概念はバランスを取ることです。メッセージ・コンシューマーとメッセージ・プロデューサーとのバランスが取れていない場合、生じうる危険として、コンシュームされていないメッセージのバックログが増大し、複数のアプリケーションのパフォーマンスに深刻な影響を及ぼす場合があります。

[パブリッシュ/サブスクライブ・ネットワークでのサブスクリプションのパフォーマンス](#)
IBM MQ での分散パブリッシュ/サブスクライブは、さまざまなトピック・ストリングのサブスクリプションがキュー・マネージャー・ネットワーク内のどこで作成されたかに関する情報を伝搬させることで機能します。これにより、メッセージのパブリッシュが行われたキュー・マネージャーが、パブリッシュされたメッセージのコピーを必要とする他のキュー・マネージャーを識別して、一致するサブスクリプションに対応させることが可能になります。

トピック・ホストでルーティングされるパブリッシュ/サブスクライブ・クラスターのパフォーマンス

トピック・ホストでルーティングされるパブリッシュ/サブスクライブ・クラスターを使用すると、各トピックをどのキュー・マネージャーがホストするかを正確に制御できます。これらのトピック・ホストが、そのトピック・ツリー・ブランチのルーティング・キュー・マネージャーになります。さらに、サブスクリプションもパブリッシャーもないキュー・マネージャーは、トピック・ホストに接続する必要はありません。この構成では、クラスター内のキュー・マネージャー間の接続の数、およびキュー・マネージャー間で渡される情報の量がかなり少なくなる可能性があります。

トピック・ホスト・ルーティングされるパブリッシュ/サブスクライブ・クラスターは、次のように動作します。

- トピックは、クラスター内の個々のトピック・ホスト・キュー・マネージャーで手動で定義されます。
- クラスター・キュー・マネージャーでサブスクリプションが行われた場合、プロキシ・サブスクリプションはトピック・ホストでのみ作成されます。
- アプリケーションがトピックに対して情報をパブリッシュすると、受信キュー・マネージャーが、トピックをホストするキュー・マネージャーにパブリケーションを転送します。その後、トピック・ホストはパブリケーションを、そのトピックに対する有効なサブスクリプションを持つクラスター内のすべてのキュー・マネージャーに送信します。

トピック・ホスト・ルーティングのさらに詳しい紹介については、[クラスターでのトピック・ホスト・ルーティング](#)を参照してください。

トピック・ホスト・ルーティングには次の利点があることから、多くの構成でこれは直接ルーティングよりも適切なトポロジーです。

- 大規模クラスターのスケラビリティが向上します。クラスター内の他のすべてのキュー・マネージャーに接続できないのは、トピック・ホスト・キュー・マネージャーだけです。このため、直接ルーティングの場合に比べてキュー・マネージャー間のチャネル数が少なくなり、キュー・マネージャー間のパブリッシュ/サブスクライブ管理トラフィックが減ります。キュー・マネージャーでサブスクリプションが変更されるときに通知しなければならないのは、トピック・ホスト・キュー・マネージャーだけです。
- 物理構成を制御できる程度が大きくなります。直接ルーティングでは、すべてのキュー・マネージャーがすべての役割を担うため、すべてを同等の能力にする必要があります。トピック・ホスト・ルーティ

ングでは、トピック・ホスト・キュー・マネージャーを明示的に選択します。そのため、それらのキュー・マネージャーは十分能力のある装置で実行するようにし、他のキュー・マネージャーにはそれほど強力でないシステムを使用できます。

しかし、トピック・ホスト・ルーティングでは、システムにいくつかの点で制約が課せられることにもなります。

- 直接ルーティングの場合よりも、システム構成および保守を十分に計画する必要があります。トピック・ツリーでクラスター化を実行するポイントや、クラスター内のトピック定義の場所を決定する必要があります。
- 直接ルーティング型トピックとちょうど同じように、新しいトピック・ホスト・ルーティング型トピックが定義されると、情報がフル・リポジトリ・キュー・マネージャーにプッシュされ、そこからクラスター内のすべてのメンバーに送信されます。フル・リポジトリからクラスターの各メンバーへのチャンネルがまだ開始されていないのであれば、このイベントによりそれが開始されることとなります。
- クラスター内にサブスクリプションがない場合でも、非ホスト・キュー・マネージャーからホスト・キュー・マネージャーにパブリケーションが常に送信されます。そのため、通常はサブスクリプションが存在することが予期される場合、あるいはグローバルな接続とナレッジによるオーバーヘッドが余分のパブリケーション・トラフィックのリスクより大きい場合は、ルーティング型トピックを使用してください。
- 非ホスト・キュー・マネージャー上でパブリッシュされるメッセージは、サブスクリプションをホストするキュー・マネージャーに直接には到達しません。それらは、常にトピック・ホスト・キュー・マネージャーを経由してルーティングされます。このアプローチにより、クラスターの総オーバーヘッドが増加し、メッセージの遅延が大きくなり、パフォーマンスが低下することがあります。

注: 構成によっては、集中型パブリッシャーまたはサブスクライバーを使用したトピック・ホスト・ルーティングの説明に従ってこの制約を除去すると便利な場合があります。

- 単一のトピック・ホスト・キュー・マネージャーを使用すると、トピックに対してパブリッシュされるすべてのメッセージに対して単一障害点が存在することとなります。その単一障害点は、複数のトピック・ホストを定義すると除去できます。しかし、ホストが複数になると、サブスクリプションの受け取るパブリッシュ・メッセージの順序に影響します。
- トピック・ホスト・キュー・マネージャーを使用すると、複数のキュー・マネージャーからのパブリケーション・トラフィックを処理する必要があるため、追加のメッセージ負荷が発生します。この負荷を軽減するには、単一トピックに対して複数のトピック・ホストを使用するか（この場合、メッセージの順序は維持されない）、またはトピック・ツリーの複数の異なるブランチに対して、ルーティングされるトピックをホストするために複数の異なるキュー・マネージャーを使用することができます。

集中型パブリッシャーまたはサブスクライバーを使用したトピック・ホスト・ルーティング

パブリケーションが常にトピック・ホスト・キュー・マネージャー経由でサブスクリプションにルーティングされる場合に、余分な「ホップ」が発生しないようにするには、トピックをホストするのと同じキュー・マネージャー上にパブリッシャーまたはサブスクリプションを構成します。この方法を使用すると、次の2つの状況で最大限のパフォーマンスが得られます。

- 多数のパブリッシャーと少数のサブスクリプションがあるトピック。この場合は、トピック・ホスト・キュー・マネージャーでサブスクリプションをホストします。
- 少数のパブリッシャーと多数のサブスクリプションがあるトピック。この場合は、トピック・ホスト・キュー・マネージャーでパブリッシャーをホストします。

次の図は、サブスクリプションもホストするトピック・ホスト・キュー・マネージャーを示しています。このアプローチによって、パブリッシャーとサブスクライバーの間の余分な「ホップ」がなくなり、クラスターのすべてのメンバー間でサブスクリプション情報が不必要に共有されることが少なくなります。

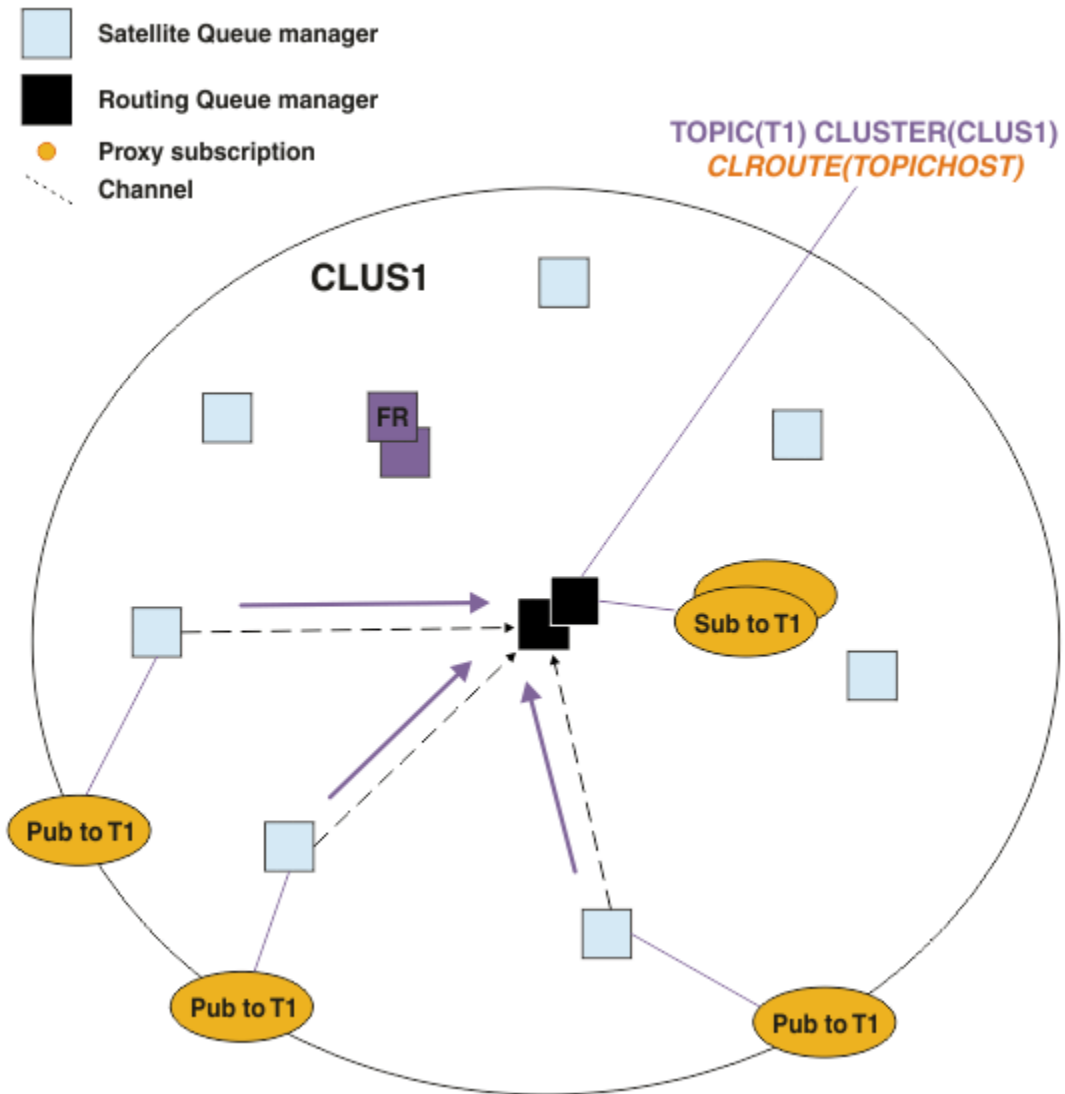


図 27. トピック・ホスト・キュー・マネージャーでのサブスクリプションのホスト

次の図は、パブリッシャーもホストするトピック・ホスト・キュー・マネージャーを示しています。このアプローチによって、パブリッシャーとサブスクライバーの間の余分な「ホップ」がなくなり、クラスターのすべてのメンバー間でサブスクリプション情報が不必要に共有されることが少なくなります。

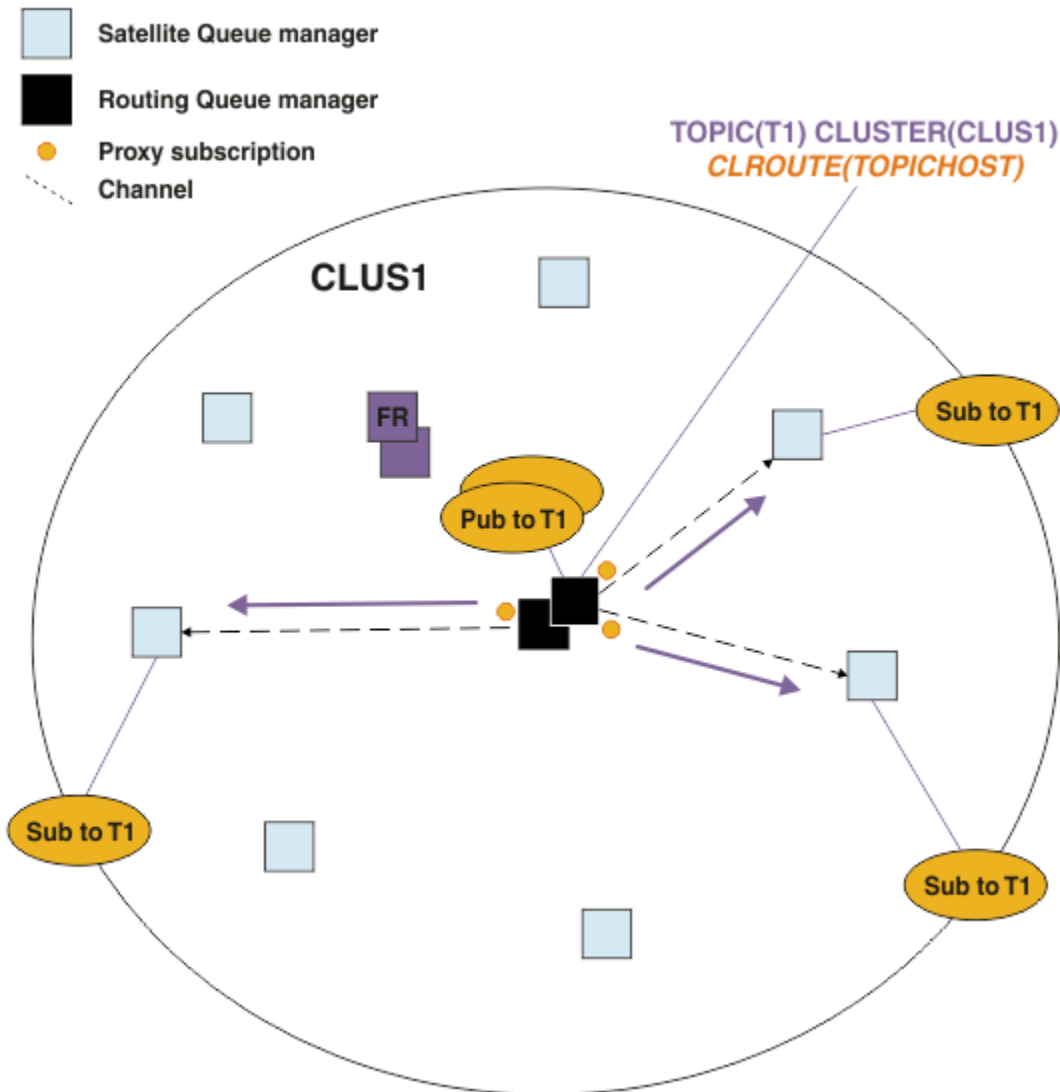


図 28. トピック・ホスト・キュー・マネージャーでのパブリケーションのホスト

関連概念

直接ルーティング型パブリッシュ/サブスクライブ・クラスターのパフォーマンス

直接ルーティング型パブリッシュ/サブスクライブ・クラスターでは、すべてのクラスター・キュー・マネージャーがパブリッシュ/サブスクライブ・メッセージングにアクティブに関与しているかどうかにかかわらず、クラスター・トピックやプロキシ・サブスクリプションなどの情報がクラスターのすべてのメンバーにプッシュされます。このプロセスによって、システムに多大な負荷が生じる可能性があります。クラスター管理がパフォーマンスに及ぼす影響を軽減するには、オフピーク時に更新を実行したり、ずっと小規模な、パブリッシュ/サブスクライブに関与するキュー・マネージャーのサブセットを定義してそれを「オーバーラップ」クラスターにしたり、トピック・ホスト・ルーティング型の使用に切り替えたりすることができます。

パブリッシュ/サブスクライブ・ネットワークにおけるプロデューサーとコンシューマーのバランシング

非同期メッセージング・パフォーマンスの重要な概念はバランスを取ることです。メッセージ・コンシューマーとメッセージ・プロデューサーとのバランスが取れていない場合、生じる危険として、コンシュームされていないメッセージのバックログが増大し、複数のアプリケーションのパフォーマンスに深刻な影響を及ぼす場合があります。

パブリッシュ/サブスクライブ・ネットワークでのサブスクリプションのパフォーマンス

IBM MQでの分散パブリッシュ/サブスクライブは、さまざまなトピック・ストリングのサブスクリプションがキュー・マネージャー・ネットワーク内のどこで作成されたかに関する情報を伝搬させることで機能します。これにより、メッセージのパブリッシュが行われたキュー・マネージャーが、パブリッシュされ

たメッセージのコピーを必要とする他のキュー・マネージャーを識別して、一致するサブスクリプションに対応させることが可能になります。

パブリッシュ/サブスクライブ・ネットワークにおけるプロデューサーとコンシューマーのバランス

非同期メッセージング・パフォーマンスの重要な概念はバランスを取ることです。メッセージ・コンシューマーとメッセージ・プロデューサーとのバランスが取れていない場合、生じうる危険として、コンシュームされていないメッセージのバックログが増大し、複数のアプリケーションのパフォーマンスに深刻な影響を及ぼす場合があります。

Point-to-Point メッセージング・トポロジーでは、メッセージ・コンシューマーとメッセージ・プロデューサーの間の関係は理解しやすいといえます。メッセージのプロデュースとコンシュームの見積もりは、キューごと、およびチャンネルごとに取得できます。バランスが取れていない場合、ボトルネックを容易に識別して修正できます。

パブリッシュ/サブスクライブ・トポロジーで、パブリッシャーとサブスクライバーのバランスが取れているかどうかを分析することは、それよりも難しくなります。各サブスクリプションから始めて、そのトピックに対するパブリッシャーを持つキュー・マネージャーにまでさかのぼります。各キュー・マネージャーから各サブスクライバーに流れるパブリケーションの数を計算します。

リモート・キュー・マネージャーでのサブスクリプションと(プロキシー・サブスクリプションに基づき)一致する各パブリケーションは、伝送キューに書き込まれます。そのパブリケーションのプロキシー・サブスクリプションが複数のリモート・キュー・マネージャーにある場合、複数のメッセージのコピーが、それぞれ異なる送信側チャンネルに向けて伝送キューに書き込まれます。

パブリッシュ/サブスクライブ・クラスターでは、それらのパブリケーションは、サブスクリプションをホストするリモート・キュー・マネージャー上の SYSTEM.INTER.QMGR.PUBS キューに向けられます。階層では、各パブリケーションは、SYSTEM.BROKER.DEFAULT.STREAM キューか、またはリモート・キュー・マネージャー上にある SYSTEM.QPUBSUB.QUEUE.NAMELIST 内にリストされた他のすべてのストリーム・キューに向けられます。各キュー・マネージャーは、そのキューに到着したメッセージを処理し、そのキュー・マネージャーにある正しいサブスクリプションにメッセージを配信します。

そのため、ボトルネックが発生する可能性のある以下のポイントでの負荷をモニターする必要があります。

- 個々のサブスクリプション・キューでの負荷をモニターします。
 - このボトルネックは、サブスクライブ側アプリケーションによるパブリケーションのコンシュームが、パブリケーションのパブリッシュより遅いペースで行われていることを意味します。
- SYSTEM.INTER.QMGR.PUBS キューまたはストリーム・キューでの負荷をモニターします。
 - このボトルネックは、キュー・マネージャーが、パブリケーションをローカル・サブスクリプションに配信できるペースよりも速く、1つ以上のリモート・キュー・マネージャーからパブリケーションを受信していることを意味します。
 - クラスター内でトピック・ホスト・ルーティング型を使用している際に、あるトピック・ホスト・キュー・マネージャーで認識される場合には、追加のキュー・マネージャーをトピック・ホストにして、それらの間でパブリケーションのワークロードのバランスを取れるようにすることを検討してください。ただし、これを行うと、パブリケーション間のメッセージの順序に影響が及びます。単一のトピックに複数のトピック・ホストを使用するトピック・ホスト・ルーティングを参照してください。
- パブリッシュ側キュー・マネージャー上の伝送キューによる送り先となる、サブスクライブ側キュー・マネージャーとパブリッシュ側キュー・マネージャーの間のチャンネルでの負荷をモニターします。
 - このボトルネックは、1つ以上のチャンネルが稼働していないか、または、チャンネルがリモート・キュー・マネージャーに配信できるよりも速いペースでメッセージがローカル・キュー・マネージャーにパブリッシュされていることを意味します。
 - パブリッシュ/サブスクライブ・クラスターを使用する場合には、ターゲット・キュー・マネージャーで追加のクラスター受信側チャンネルを定義することを検討してください。これにより、それらの間でパブリケーションのワークロードのバランスを取ることが可能になります。ただし、これを行うと、パブリケーション間のメッセージの順序に影響を与えます。また、複数のクラスター伝送キューの構

成に移行することも検討してください。特定の環境では、これによりパフォーマンスが向上する可能性があるためです。

- パブリッシュ・アプリケーションがキューに入れられたパブリッシュ/サブスクライブ・インターフェースを使用している場合は、(a) `SYSTEM.BROKER.DEFAULT.STREAM` キュー、および `SYSTEM.QPUBSUB.QUEUE.NAMELIST` にリストされているその他のストリーム・キューの負荷をモニターします。(b) `SYSTEM.BROKER.DEFAULT.SUBPOINT` キュー、および `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST` にリストされている他のすべてのサブポイント・キュー。
 - このボトルネックは、ローカル・キュー・マネージャーが処理できるよりも速いペースでメッセージがローカルのパブリッシュ側アプリケーションによって書き込まれていることを意味します。

関連概念

直接ルーティング型パブリッシュ/サブスクライブ・クラスターのパフォーマンス

直接ルーティング型パブリッシュ/サブスクライブ・クラスターでは、すべてのクラスター・キュー・マネージャーがパブリッシュ/サブスクライブ・メッセージングにアクティブに関与しているかどうかにかかわらず、クラスター・トピックやブローカー・サブスクリプションなどの情報がクラスターのすべてのメンバーにプッシュされます。このプロセスによって、システムに多大な負荷が生じる可能性があります。クラスター管理がパフォーマンスに及ぼす影響を軽減するには、オフピーク時に更新を実行したり、ずっと小規模な、パブリッシュ/サブスクライブに関与するキュー・マネージャーのサブセットを定義してそれを「オーバーラップ」クラスターにしたり、トピック・ホスト・ルーティング型の使用に切り替えたりすることができます。

トピック・ホストでルーティングされるパブリッシュ/サブスクライブ・クラスターのパフォーマンス

トピック・ホストでルーティングされるパブリッシュ/サブスクライブ・クラスターを使用すると、各トピックをどのキュー・マネージャーがホストするかを正確に制御できます。これらのトピック・ホストが、そのトピック・ツリー・ブランチのルーティング・キュー・マネージャーになります。さらに、サブスクリプションもパブリッシャーもないキュー・マネージャーは、トピック・ホストに接続する必要はありません。この構成では、クラスター内のキュー・マネージャー間の接続の数、およびキュー・マネージャー間で渡される情報の量がかなり少なくなる可能性があります。

パブリッシュ/サブスクライブ・ネットワークでのサブスクリプションのパフォーマンス

IBM MQ での分散パブリッシュ/サブスクライブは、さまざまなトピック・ストリングのサブスクリプションがキュー・マネージャー・ネットワーク内のどこで作成されたかに関する情報を伝搬させることで機能します。これにより、メッセージのパブリッシュが行われたキュー・マネージャーが、パブリッシュされたメッセージのコピーを必要とする他のキュー・マネージャーを識別して、一致するサブスクリプションに対応させることが可能になります。

314 ページの『クラスターのモニター』

クラスター内で、アプリケーション・メッセージ、制御メッセージ、およびログをモニターできます。キューの複数のインスタンス間でクラスターのロード・バランシングが行われる場合は、モニターに関する特別な考慮事項があります。

パブリッシュ/サブスクライブ・ネットワークでのサブスクリプションのパフォーマンス

IBM MQ での分散パブリッシュ/サブスクライブは、さまざまなトピック・ストリングのサブスクリプションがキュー・マネージャー・ネットワーク内のどこで作成されたかに関する情報を伝搬させることで機能します。これにより、メッセージのパブリッシュが行われたキュー・マネージャーが、パブリッシュされたメッセージのコピーを必要とする他のキュー・マネージャーを識別して、一致するサブスクリプションに対応させることが可能になります。

この方式により、パブリッシュされたメッセージが、一致するサブスクリプションが存在しないキュー・マネージャーに送信されることを最小限に抑えることができます。ただし、サブスクライブされているトピック・ストリングの数が多く、サブスクリプションの作成と削除が頻繁に行われて絶えず変化している場合には、サブスクリプションに関する情報の伝搬はかなり大きなオーバーヘッドとなる場合があります。

パブリッシュ/サブスクライブ・ネットワーク内でパブリケーションとサブスクリプションが流れる方法を調整することにより、パフォーマンスを変化させることができます。ネットワーク・トラフィックで扱うパブリケーションが少なく、サブスクリプションの作成、削除、または変更が速いペースで行われる場合には、サブスクリプション情報がすべてのキュー・マネージャーに流れないようにし、その代わりにすべ

てのパブリケーションをネットワーク内のすべてのキュー・マネージャーに転送することができます。また、接続されているキュー・マネージャー間で特定のトピックに関するプロキシ・サブスクリプションおよびパブリケーションのフローを制限したり、ワイルドカードを含むプロキシ・サブスクリプションのフローを制限したり、トピック・ストリングの数および一時的な性質を低減したりすることもできます。

個別サブスクリプション伝搬と全対象パブリッシュ

全対象パブリッシュは、個別サブスクリプション伝搬の代わりとなります。個別伝搬の場合、パブリケーションは、それと一致するサブスクリプションがあるキュー・マネージャーにのみ転送されます。全対象パブリッシュの場合、すべてのパブリケーションが、ネットワーク内のすべてのキュー・マネージャーに転送されます。その後、受信側のキュー・マネージャーは、ローカル・サブスクリプションと一致するパブリケーションを送信します。

個別サブスクリプション伝搬

このメカニズムの場合、キュー・マネージャーのサブスクリプションと一致するパブリケーションのみが送信されるため、キュー・マネージャー間のパブリケーション・トラフィックが最小になります。

ただし、以下の制限があります。

- サブスクライブされる個々のトピック・ストリングごとに、プロキシ・サブスクリプションが、パブリッシュ/サブスクライブ・トポロジー内の他のキュー・マネージャーに送信されます。キュー・マネージャーのセットは、分散パブリッシュ/サブスクライブ・ネットワークの計画に示されているように、使用されるルーティング・モデルに応じて異なります。
 - このメッセージングのオーバーヘッドは、次のような場合にかなり大きくなる場合があります。それは、何千ものサブスクリプションの作成または削除(例えば、キュー・マネージャーの再始動後に行われるすべての非永続サブスクリプションの再作成)が行われる場合や、サブスクリプションのセットが早いペースで変更され、それぞれが別のトピック・ストリングに対するものである場合です。
 - プロキシ・サブスクリプションの伝搬先となるキュー・マネージャーの数も、オーバーヘッドのスケールに影響を及ぼします。
- プロキシ・サブスクリプションは、非同期メッセージングを使用して他のキュー・マネージャーに流されます。これにより、以下の影響が生じます。
 - サブスクリプションの作成から、プロキシ・サブスクリプションの作成、送信、およびその処理が他のキュー・マネージャーによって行われるまでの間で、遅延が生じます。
 - この間隔の中でこれらのキュー・マネージャーでパブリッシュされるメッセージは、リモート・サブスクリプションには送信されません。

全対象パブリッシュ

このメカニズムを使用した場合、システムにおけるトピック・ストリングごとのプロキシ・サブスクリプションによるオーバーヘッドはありません。つまり、早いペースでサブスクリプションの作成、削除、または変更が行われても、ネットワーク・ロードおよび処理が増えることはありません。

また、すべてのパブリケーションがすべてのキュー・マネージャーに流れるため、サブスクリプションの作成から、パブリケーションがキュー・マネージャーに流れるまでの間に遅延はありません。したがって、新しく作成されたリモート・サブスクリプションにパブリケーションが送信されない時間帯はありません。

ただし、以下の制限があります。

- すべてのパブリケーションをパブリッシュ/サブスクライブ・トポロジー内のすべてのキュー・マネージャーに送信することにより、各キュー・マネージャーでパブリケーションと一致するサブスクリプションが存在しないときに、ネットワーク・トラフィックが過多になる可能性があります。
 - トポロジー内のキュー・マネージャー数が多いほど、オーバーヘッドが大きくなります。

全対象パブリッシュ・メカニズムの使用を検討できるケースとしては、かなりの割合のキュー・マネージャーからパブリケーションのサブスクライブが出されることが想定される場合、またはサブスクリプションの変更の頻度が原因でプロキシ・サブスクリプションのオーバーヘッドが大きくなりすぎる場合があります。個別プロキシ・サブスクリプション転送を使用すべきケースとしては、パブリケーションが、一致するサブスクリプションがあるキュー・マネージャーに対して送信されるときよりも、すべてのキュー・マネージャーに対して送信されるときにメッセージング・トラフィックが増加するような場合です。

全対象パブリッシュの動作は、トピック・ツリー内の任意のレベルで設定することができます。全対象パブリッシュを使用可能にするには、上位トピック・オブジェクトの **PROXYSUB** パラメーターを **FORCE** に設定します。この結果、トピック・ツリー内のこのトピック・オブジェクトの下のすべてのトピックに一致する単一ワイルドカードのプロキシ・サブスクリプションが得られます。クラスター・トピック・オブジェクトに設定すると、**PROXYSUB(FORCE)** 属性は、トピックが定義されたキュー・マネージャーだけでなく、ネットワーク内のすべてのキュー・マネージャーに伝搬されます。

注：階層で使用する場合は、各キュー・マネージャーで **PROXYSUB(FORCE)** を個別に設定するため、トポロジー・メカニズムによってチャンネルの数が自然に制限されます。一方、クラスターで使用される場合には、さらに多くのチャンネルが開始される可能性があります。

- トピック・ホスト・ルーティング型クラスターでは、各キュー・マネージャーから各トピック・ホスト・キュー・マネージャーにチャンネルが開始されます。
- 直接ルーティング型クラスターでは、各キュー・マネージャーから他のすべてのキュー・マネージャーにチャンネルが開始されます。

多数のチャンネルの開始によるオーバーヘッドは、直接ルーティング型クラスターにおいて最も顕著であり、パフォーマンス上の問題となる可能性があります。377 ページの『[直接ルーティング型パブリッシュ/サブスクライブ・クラスターのパフォーマンス](#)』を参照。

接続されているキュー・マネージャー間でのプロキシ・サブスクリプションおよびパブリケーションのフローを制限する他の方法

トピック・ストリングを統合する

多数の別個の一時的トピック・ストリングを使用すると、パブリッシャーまたはサブスクリプションが接続しているシステム内の各キュー・マネージャーで何らかのレベルの管理オーバーヘッドが発生します。トピック・ストリングの使用について定期的に検討して、それらを統合できないか判断してください。トピック・ストリングの数を減らし、一時的な性質をもつトピック・ストリングの使用を避けることで、トピック・ストリングに対するパブリッシャーおよびサブスクリプションが削減され、システムへの影響が軽減されます。

パブリケーションおよびサブスクリプションの有効範囲を制限する

特定のトピックについては、[パブリケーション・スコープ](#)および[サブスクリプション・スコープ](#)の設定を使用して、パブリケーションおよびサブスクリプションを、それらが定義されているキュー・マネージャーに対してローカルに保つことができます。

ワイルドカード付きのトピックに対するサブスクリプションをブロックする

Topic 属性 **WILDCARD** を **BLOCK** に設定することにより、ワイルドカードを含むプロキシ・サブスクリプションのフローを制限できます。[プロキシ・サブスクリプション内のワイルドカード](#)を参照してください。

383 ページの『[パブリッシュ/サブスクライブ・ネットワークにおけるプロデューサーとコンシューマーのバランシング](#)』も参照してください。

クラスターにおけるプロキシ・サブスクリプションのトラフィックをモニターする

プロキシ・サブスクリプションのトラフィックによるシステムの負荷を考慮する際には、383 ページの『[パブリッシュ/サブスクライブ・ネットワークにおけるプロデューサーとコンシューマーのバランシング](#)』にリストされているキューをモニターすることに加えて、以下のクラスター・キューもモニターしてください。

- サブスクライバー・キュー・マネージャー上の **SYSTEM.INTER.QMGR.FANREQ** キュー
- クラスター内の他のすべてのキュー・マネージャー上の **SYSTEM.INTER.QMGR.CONTROL** キュー

上記のキューに大量のメッセージ・バックログがある場合、サブスクリプションの変更率がシステムにとって大きすぎるか、またはキュー・マネージャーがクラスター内で正常に機能していないことを意味します。特定のキュー・マネージャーに問題があると考えられる場合には、そのキュー・マネージャーに対するパブリッシュ/サブスクライブ・サポートが無効でないか確認してください。[ALTER QMGR](#) の **PSMODE** を参照してください。

関連概念

直接ルーティング型パブリッシュ/サブスクライブ・クラスターのパフォーマンス

直接ルーティング型パブリッシュ/サブスクライブ・クラスターでは、すべてのクラスター・キュー・マネージャーがパブリッシュ/サブスクライブ・メッセージングにアクティブに関与しているかどうかにかかわらず、クラスター・トピックやプロキシ・サブスクリプションなどの情報がクラスターのすべてのメンバーにプッシュされます。このプロセスによって、システムに多大な負荷が生じる可能性があります。クラスター管理がパフォーマンスに及ぼす影響を軽減するには、オフピーク時に更新を実行したり、ずっと小規模な、パブリッシュ/サブスクライブに関与するキュー・マネージャーのサブセットを定義してそれを「オーバーラップ」クラスターにしたり、トピック・ホスト・ルーティング型の使用に切り替えたりすることができます。

トピック・ホストでルーティングされるパブリッシュ/サブスクライブ・クラスターのパフォーマンス

トピック・ホストでルーティングされるパブリッシュ/サブスクライブ・クラスターを使用すると、各トピックをどのキュー・マネージャーがホストするかを正確に制御できます。これらのトピック・ホストが、そのトピック・ツリー・ブランチのルーティング・キュー・マネージャーになります。さらに、サブスクリプションもパブリッシャーもないキュー・マネージャーは、トピック・ホストに接続する必要はありません。この構成では、クラスター内のキュー・マネージャー間の接続の数、およびキュー・マネージャー間で渡される情報の量がかなり少なくなる可能性があります。

パブリッシュ/サブスクライブ・ネットワークにおけるプロデューサーとコンシューマーのバランス
非同期メッセージング・パフォーマンスの重要な概念はバランスを取ることです。メッセージ・コンシューマーとメッセージ・プロデューサーとのバランスが取れていない場合、生じうる危険として、コンシュームされていないメッセージのバックログが増大し、複数のアプリケーションのパフォーマンスに深刻な影響を及ぼす場合があります。

パブリッシュ/サブスクライブ・ネットワークでのプロキシ・サブスクリプション

トピック・ツリー内の不要なトピック数の削減

トピック・ツリー内の不要なトピックの数を減らすことによって、パブリッシュ/サブスクライブ・システムのパフォーマンスは向上します。どのようなトピックが不要なトピックであるか、および不要なトピックを削除する方法について説明します。

パフォーマンスに悪影響を与えることなく、多数のトピックを作成することができます。ただし、パブリッシュ/サブスクライブを使用する方法によっては、トピック・ツリーが継続的に拡張されるものがあります。非常に多くのトピックが、一度作成されたきり、二度と使用されることがありません。トピック数が増加していくと、パフォーマンス上の問題が発生する場合があります。

多数の不要なトピックが作成され続けるような設計を避けるには、どうすればよいでしょうか。キュー・マネージャーがトピック・ツリーから不要なトピックを削除できるようにするには、どうすればよいでしょうか。

キュー・マネージャーは、トピックが 30 分間使用されなければ、それを不要なトピックと見なします。キュー・マネージャーは、使用されないトピックをトピック・ツリーから削除します。30 分の期間は、キュー・マネージャー属性 **TREELIFE** を変更することによって変更できます。キュー・マネージャーに対して、トピックが使用されていないことが示されていることを確認し、キュー・マネージャーが不要なトピックを削除できるようにします。387 ページの『使用されないトピックとは何か』のセクションでは、使用されないトピックとは何かが説明されています。

アプリケーション (特に、実行時間の長いアプリケーション) を設計するプログラマーは、リソースの使用 (プログラムが必要とするリソース量、無制限の要求の有無、リソース・リークの有無) について検討します。トピックは、パブリッシュ/サブスクライブ・プログラムが使用するリソースです。プログラムが使用するその他のリソースと同様に、トピックの使用法を綿密に調べてください。

使用されないトピックとは何か

使用されないトピックとは何かを定義する前に、具体的に何がトピックと見なされるのかについて説明します。

トピック・ストリング (USA/Alabama/Auburn など) がトピックに変換されると、トピック・ツリーにトピックが追加されます。追加のトピック・ノードと、それに対応するトピックが、必要に応じてツリー内

に作成されます。トピック・ストリング USA/Alabama/Auburn は、3 つのトピックを持つツリーに変換されます。

- USA
- USA/Alabama
- USA/Alabama/Auburn

トピック・ツリー内のすべてのトピックを表示するには、**runmqsc** コマンド `DISPLAY TPSTATUS('#') TYPE(TOPIC)` を使用します。

トピック・ツリー内の使用されていないトピックには以下の特性があります。

トピック・オブジェクトと関連付けられていない

管理トピック・オブジェクトには、それをトピックと関連付けるトピック・ストリングがあります。トピック・オブジェクト Alabama を定義すると、関連付けるべきトピック USA/Alabama が存在しない場合は、トピック・ストリングからトピックが作成されます。トピックが存在する場合、トピック・オブジェクトとトピックは、トピック・ストリングを使用して互いに関連付けられます。

保存パブリケーションを持たない

保存パブリケーションを持つトピックは、パブリッシャーがオプション MQPMO_RETAIN を指定してトピックにメッセージを入力した結果です。

runmqsc コマンド `DISPLAY TPSTATUS('USA/Alabama') RETAINED` を使用して、USA/Alabama に保存パブリケーションがあるかどうかを調べます。応答は YES または NO です。

runmqsc コマンド `CLEAR TOPICSTR('USA/Alabama') CLTRTYPE(RETAINED)` を使用して、保存パブリケーションを USA/Alabama から削除します。

子トピックを持たない

USA/Alabama/Auburn は、子トピックのないトピックです。USA/Alabama/Auburn は、USA/Alabama の直系の子トピックです。

runmqsc コマンド `DISPLAY TPSTATUS('USA/Alabama/+')` を使用して USA/Alabama の直接の子を表示します。

ノードに対するアクティブなパブリッシャーがない

ノードに対するアクティブなパブリッシャーは、出力用にトピックをオープンにしているアプリケーションです。

例えば、アプリケーションは、オープン・オプション MQOO_OUTPUT を指定して、**Alabama** という名前のトピック・オブジェクトを開きます。

USA/Alabama に対するアクティブ・パブリッシャーとそのすべての子を表示するには、**runmqsc** コマンド `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACTCONN` を使用します。

ノードに対するアクティブなサブスクライバーがない

アクティブなサブスクライバーは、永続サブスクリプション、または MQSUB を使用してサブスクリプションをトピックに登録し、まだそのトピックをクローズしていないアプリケーションのいずれかです。

USA/Alabama に対するアクティブなサブスクリプションを表示するには、**runmqsc** コマンド `DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACTCONN` を使用します。

USA/Alabama に対するアクティブ・サブスクリプションとそのすべての子を表示するには、**runmqsc** コマンド `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACTCONN` を使用します。

トピック・ツリー内のトピック数の管理

要約すると、トピック・ツリー内のトピック数を管理する方法は数多くあります。

TPCOUNT を表示する

runmqsc コマンドの DISPLAY PUBSUB ALL を定期的を使用して、**TPCOUNT** プロパティを表示します。これは、トピック・ツリー内のトピック・ノードの数です。数が増えている場合、**TREELIFE** を短くするか、トピック自体を再設計する必要がある可能性があります。

TREELIFE の変更

使用されないトピックの存続時間は、デフォルトで 30 分間です。使用されないトピックの存続時間を短くすることができます。

例えば、**runmqsc** コマンド ALTER QMGR TREELIFE(900) は、使用されないトピックの存続時間を 30 分間から 15 分間に短縮します。

キュー・マネージャーの再始動 (例外的)

キュー・マネージャーを再始動すると、トピック・ツリーは、トピック・オブジェクト、保存パブリケーションを含むノード、および永続サブスクリプションから再初期設定されます。パブリッシャーおよびサブスクライバー・プログラムの操作によって作成されたトピックは除去されます。

不要なトピックの増加がパフォーマンス上の問題の原因となっていた場合、最後の手段としてキュー・マネージャーを再始動します。

関連概念

[トピック・ツリー](#)

Windows Linux MQ Adv. MQ Adv.VUE Aspera gateway を使用して、待ち時間の長いネットワークでのパフォーマンスの向上を図る

IBM Aspera® faspio Gateway は、IBM MQ のネットワーク・スループットを大幅に向上させることができる高速 TCP/IP トンネルを提供します。

Aspera gateway を使用すると、キュー・マネージャーのチャンネルのパフォーマンスを向上させることができます。ネットワークの待ち時間が長い場合やパケットを損失する傾向がある場合に特に効果的です。一般には、別々のデータ・センターにあるキュー・マネージャー間の接続を高速化するために使用されます。

ただし、パケットが失われない高速ネットワークでは、Aspera gateway を使用するとパフォーマンスが低下するため、Aspera gateway 接続を定義する前後のネットワーク・パフォーマンスを確認することが重要です。

資格のあるプラットフォームで実行されているキュー・マネージャーは、Aspera gateway を介して接続できます。ゲートウェイ自体は、Red Hat®、Ubuntu Linux、または Windows にデプロイされます。

詳しくは、[Linux または Windows での Aspera gateway 接続の定義](#)を参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

IBM 本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

日本アイ・ビー・エム株式会社

法務・知的財産

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

〒 103-8510

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 INTERNATIONAL BUSINESS MACHINES CORPORATION は、法律上の瑕疵担保責任、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。"" 国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N

Rochester, MN 55901

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っていません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報 (提供されている場合) は、このプログラムで使用するアプリケーション・ソフトウェアの作成を支援することを目的としています。

本書には、プログラムを作成するユーザーが IBM MQ のサービスを使用できるようにするためのプログラミング・インターフェースに関する情報が記載されています。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

重要: この診断、修正、およびチューニング情報は、変更される可能性があるため、プログラミング・インターフェースとして使用しないでください。

商標

IBM、IBM ロゴ、ibm.com[®]は、世界の多くの国で登録された IBM Corporation の商標です。現時点での IBM の商標リストについては、"Copyright and trademark information" www.ibm.com/legal/copytrade.shtml をご覧ください。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標です。

この製品には、Eclipse Project (<https://www.eclipse.org/>) により開発されたソフトウェアが含まれています。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。



部品番号:

(1P) P/N: