

7.5

*IBM WebSphere 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、[285 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® WebSphere® MQ バージョン 7 リリース 5、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

<b>モニターおよびパフォーマンス.....</b>	<b>5</b>
イベント・モニター.....	5
観測イベント.....	6
パフォーマンス・イベント.....	20
構成イベント.....	37
コマンド・イベント.....	41
ロガー・イベント.....	43
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.....	49
メッセージ・モニター.....	55
アクティビティーと操作.....	56
メッセージ経路の手法.....	57
アクティビティー記録.....	59
経路トレース・メッセージング.....	64
IBM WebSphere MQ 経路表示アプリケーション.....	78
アクティビティー・レポート参照.....	95
経路トレース・メッセージ参照.....	122
経路トレース応答メッセージ参照.....	132
アカウントティング・メッセージと統計メッセージ.....	134
アカウントティング・メッセージ.....	135
統計メッセージ.....	138
アカウントティング情報と統計情報の表示.....	143
アカウントティング・メッセージ参照と統計メッセージ参照.....	149
アプリケーション・アクティビティー・トレース.....	197
アプリケーションのアクティビティー・トレース情報の収集.....	197
amqsact サンプル・プログラム.....	206
アプリケーション・アクティビティー・トレース・メッセージ参照.....	207
リアルタイム・モニター.....	271
リアルタイム・モニターを制御する属性.....	272
キューおよびチャネルのモニター・データの表示.....	273
キューのモニター.....	275
チャネルのモニター.....	278
Windows パフォーマンス・モニター.....	284
<b>特記事項.....</b>	<b>285</b>
プログラミング・インターフェース情報.....	286
商標.....	286



## モニターおよびパフォーマンス

---

IBM WebSphere MQ では、いくつかのモニター手法を使用して、キュー・マネージャー・ネットワークの実行方法に関する統計やその他の具体的な情報を取得することができます。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。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。

### 関連タスク

#### 構成

#### [WebSphere MQ の管理](#)

## 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 WebSphere 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

#### [イベント・メッセージ参照](#)

#### 8 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・メッセージの形式](#)

## 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 WebSphere 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 (Tivoli® NetView® for z/OS®などの他の方法で) モニターすることができます。

[7 ページの図 1](#) に、観測イベントの概念を図示します。

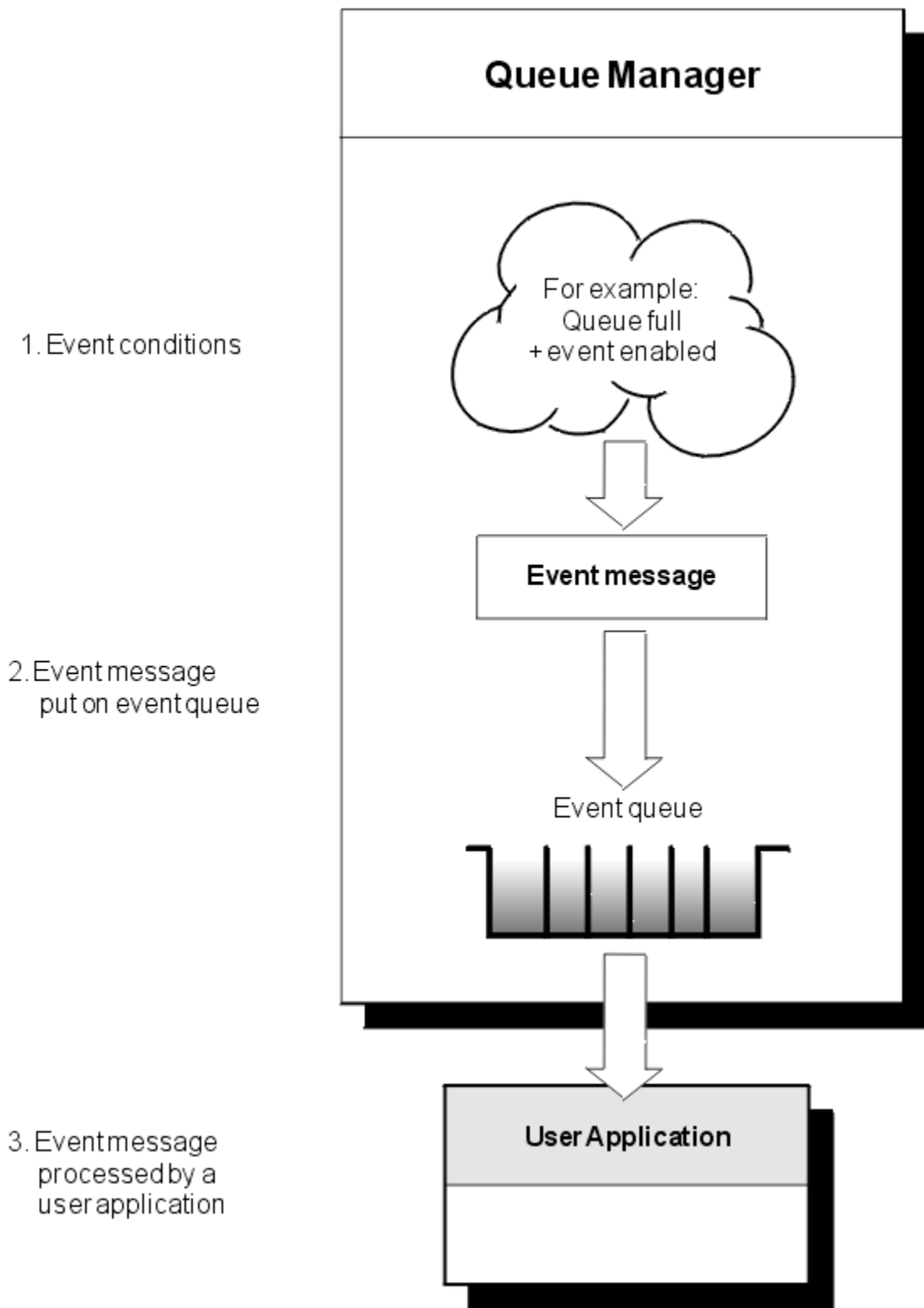


図 1. 観測イベントについて

### 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、z/OS キュー・マネージャーから UNIX システム・キュー・マネージャー上のイベントをモニターする場合は、必ず EBCDIC を ASCII に変換してください。

## 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。
- アプリケーションが、IBM WebSphere MQ for IBM i、Windows、UNIX and Linux® システムで許可されていないユーザー ID を指定して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

#### [20 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。

#### [49 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』](#)

このページでは、観測イベントをモニターするための C プログラム例を示します。

## 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 WebSphere MQ 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。

観測イベントを独自のシステム管理アプリケーションに組み込むことにより、多数のキュー・マネージャー、多数の異なるノード、および複数の IBM WebSphere MQ アプリケーションのアクティビティをモニターすることができます。特に、[10 ページの図 2](#) に示すように、単一ノード (IBM WebSphere 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。

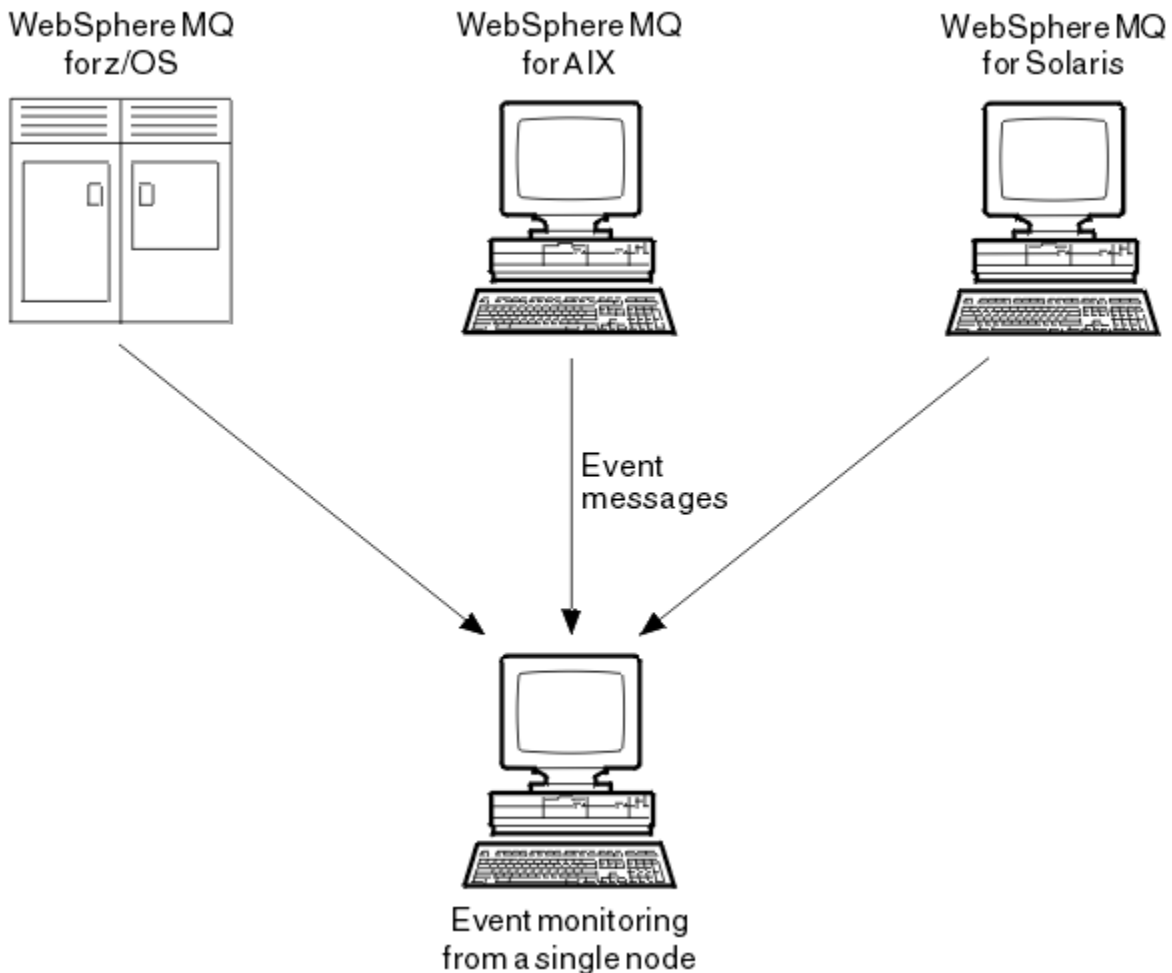


図 2. 単一のノードから異なるプラットフォームにあるキュー・マネージャーをモニターする

インストールメンテーション・イベントにより、他の管理ネットワーク (Tivoli NetView for z/OS など) のエージェントとして機能するアプリケーションは、レポートをモニターし、適切なアラートを作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。

WebSphere 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。

#### 権限 (Windows、および UNIX システムのみ)

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、Windows および UNIX システムでのみ有効です。

#### 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

ローカル・イベントは、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・キューまたは他のローカル・オブジェクトにアクセスできなかったことを示します。例えば、定義されていないオブジェクトにアプリケーションがアクセスしようとする場合です。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

リモート・イベントは、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ことを示します。例えば、使用される伝送キューが正しく定義されていないような場合です。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](#)

## SSL イベント

Secure Sockets Layer (SSL または 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、SSL または TLS を使用するチャンネルが SSL 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。

### 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。イベント・メッセージには、イベント・データ [Create object](#) が含まれています。

#### 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。イベント・メッセージには、イベント・データ [Change object](#) が含まれています。

#### 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。イベント・メッセージには、イベント・データ [Delete object](#) が含まれています。

#### 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。イベント・メッセージには、イベント・データ [Refresh object](#) が含まれています。

## 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・データが含まれます: [Command](#)。

## ロガー・イベント

ロガー・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・エクステンツに対してログ・レコードの書き込みを開始したときに報告されます。

ロガー・イベント・メッセージには、ログ・エクステンツを指定する情報が含まれています。ログ・エクステンツはキュー・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。

ロガー・イベント・メッセージには、イベント・データ [Logger](#) が含まれています。

## 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	<a href="#">非許可 (タイプ 1)</a>
	<a href="#">非許可 (タイプ 2)</a>
	<a href="#">非許可 (タイプ 3)</a>
	<a href="#">非許可 (タイプ 4)</a>
	<a href="#">不許可 (タイプ 5)</a>
	<a href="#">不許可 (タイプ 6)</a>
チャンネル・イベント	<a href="#">チャンネルアクティブ化</a>
	<a href="#">チャンネル自動定義エラー</a>
	<a href="#">チャンネル自動定義 OK</a>
	<a href="#">チャンネルのブロック</a>
	<a href="#">チャンネル変換エラー</a>
	<a href="#">チャンネル非アクティブ化</a>
	<a href="#">チャンネル開始</a>
	<a href="#">チャンネル停止</a>
	<a href="#">ユーザーによるチャンネル停止</a>
コマンド・イベント	<a href="#">コマンド</a>

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

## 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 WebSphere MQ スクリプト・コマンド (MQSC)。
- 対応する IBM WebSphere MQ PCF コマンド。
- IBM WebSphere MQ エクスプローラー。

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

### 6 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

### 8 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。16 ページの表 1 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
Authority 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。17 ページの表 2 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、該当するパラメーターを **DISABLED** に設定してください。



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す。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。

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のいずれかとして、イベント・キューを定義できます。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。

## 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、あるカテゴリーのイベント用にイベント・キューを定義しないと、そのカテゴリーのイベント・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せん。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

#### [15 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。

#### [19 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 WebSphere 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。

#### [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 WebSphere 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

[98 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。

[102 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。

[103 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 WebSphere 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。では、キュー・マネージャーの操作および制御パネルを使用してキュー属性を設定することもできます。

#### 関連資料

[8 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。21 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。

#### 関連概念

##### 20 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。

##### 23 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。

##### 23 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

##### 24 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](#)

## 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。

22 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。

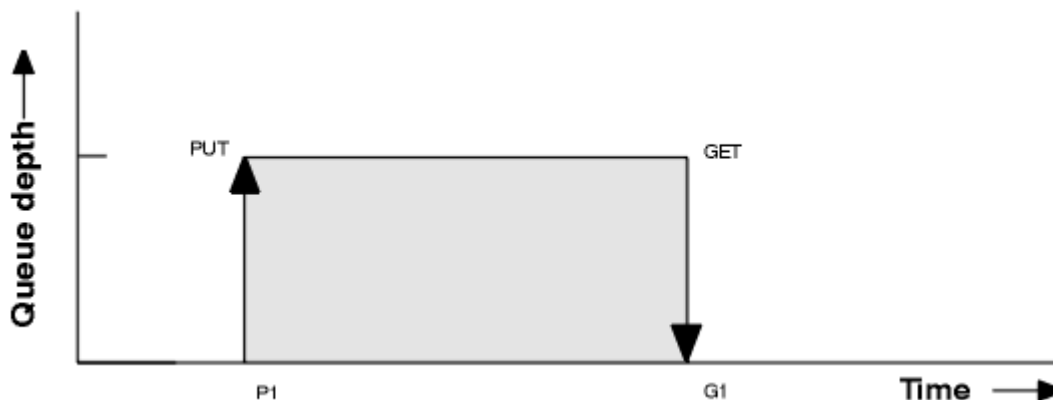


図 3. キュー・サービス間隔イベントについて

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、[23 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 \(MQLONG\)](#)

[ServiceIntervalEvent プロパティ](#)

## 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

24 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 PERFM EV 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 QSVCIINT 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') QSVCIINT(10000) QSVCIIEV(HIGH)
```

## キュー・サービス間隔イベントの例

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

次の 3 つの例では、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。

例に示されている図の構成は同じです。

- 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

### [21 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。

### [23 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。

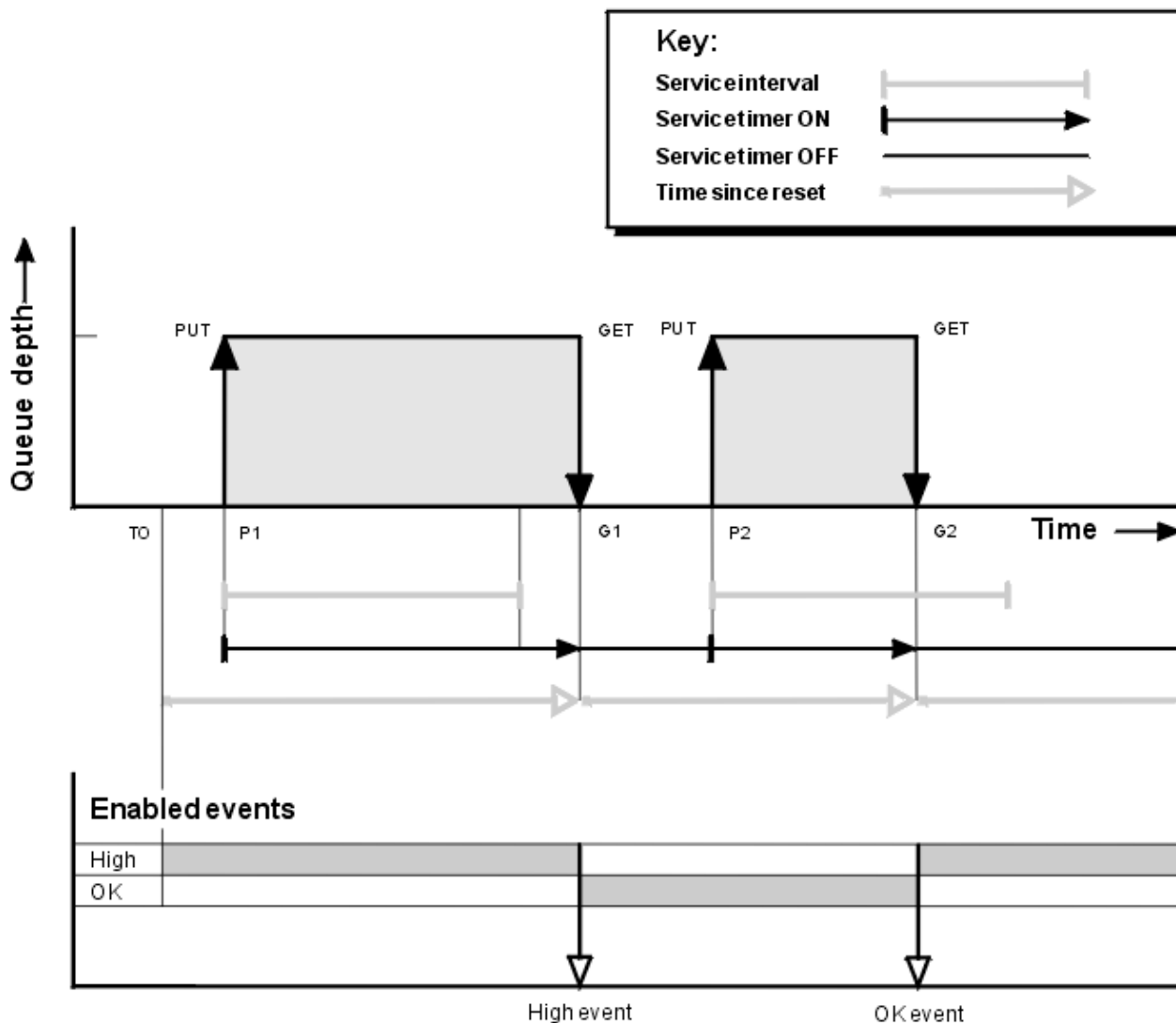


図 4. キュー・サービス間隔イベント - 例 1

### 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

27 ページの表 5 は、この例のイベント統計の要約です。

表 5. 例 1 のイベント統計の要約		
	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

26 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。

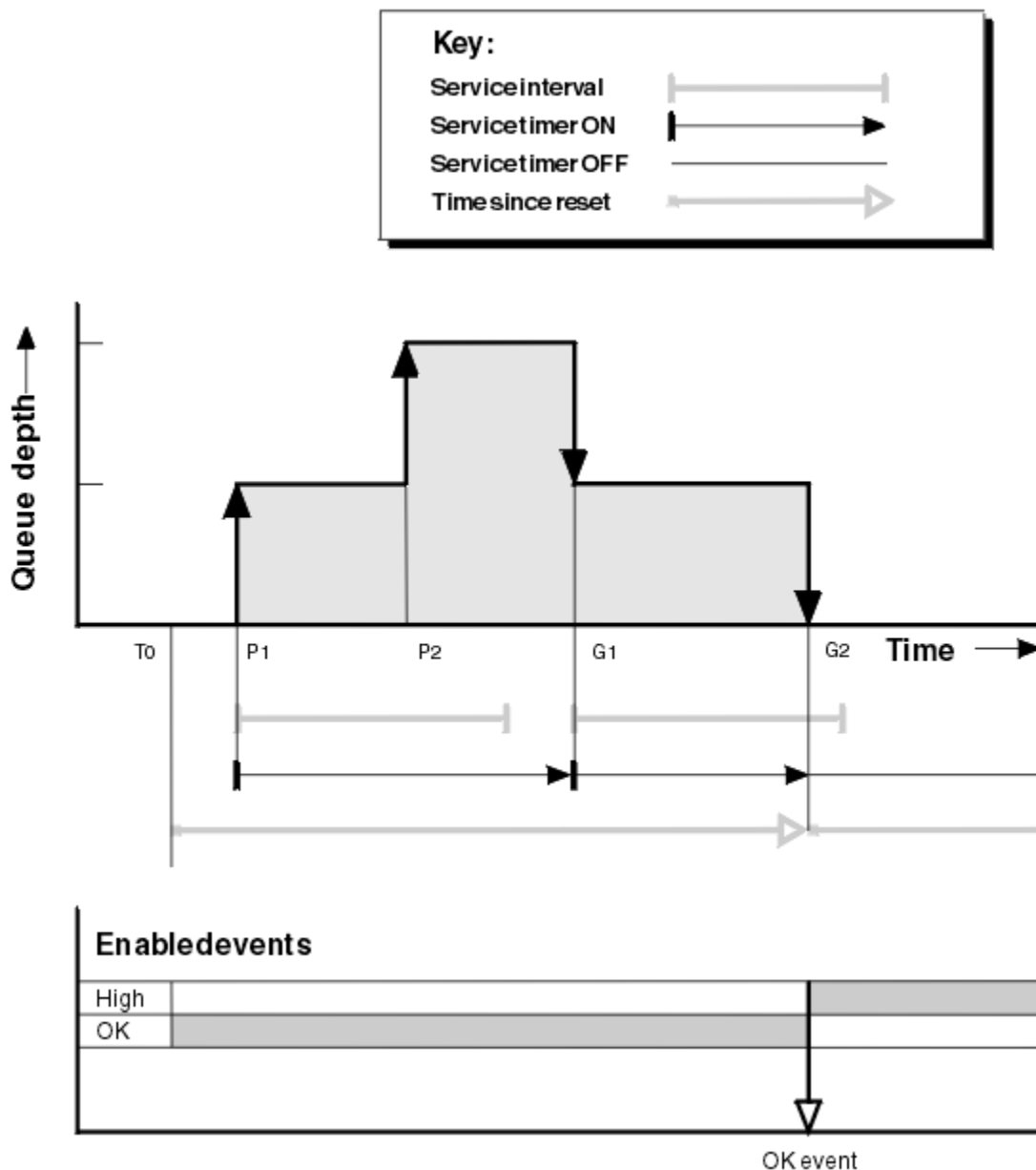


図 5. キュー・サービス間隔イベント - 例 2

### 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

29 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	
	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。

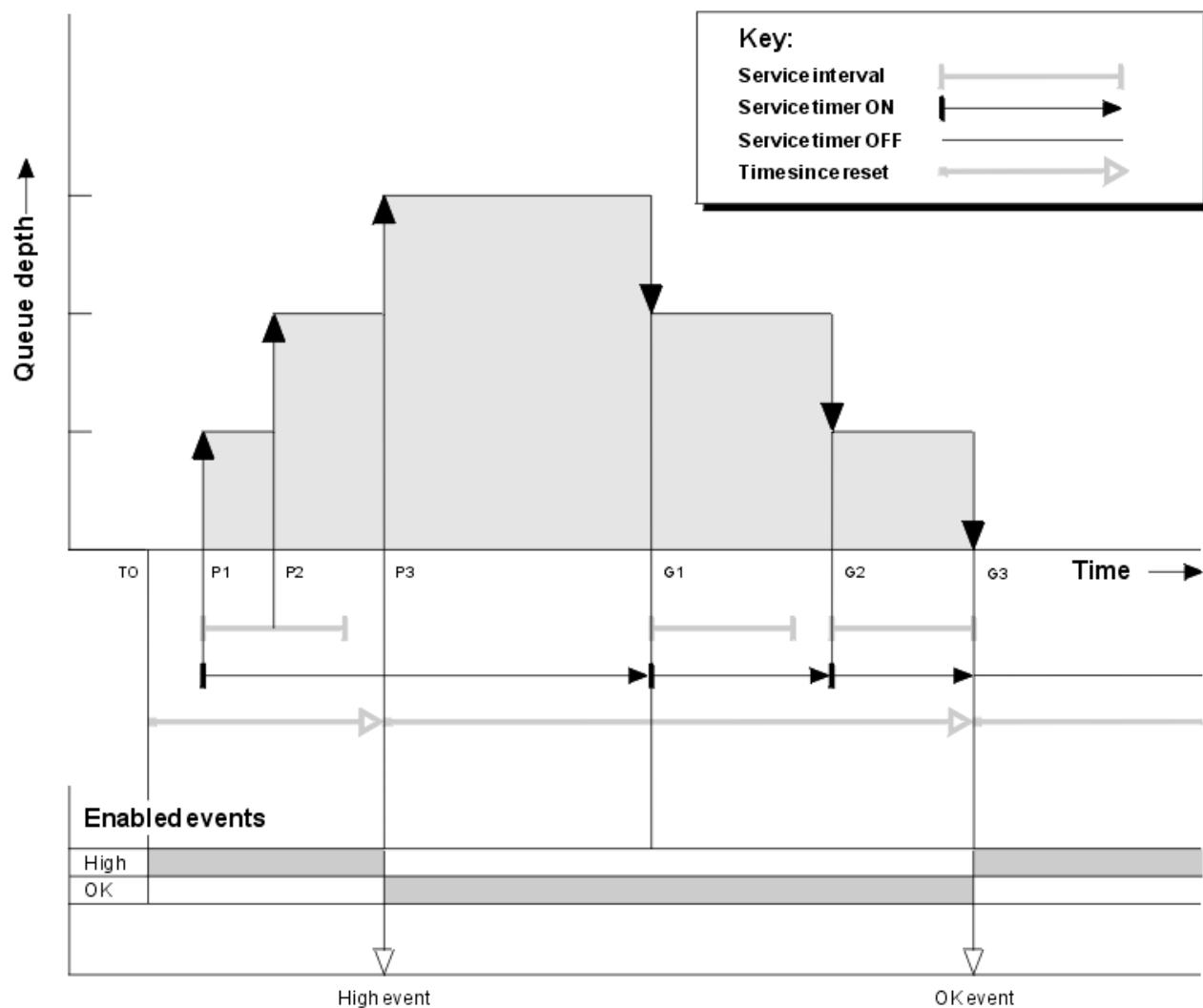


図 6. キュー・サービス間隔イベント - 例 3

## 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

30 ページの表 7 は、この例のイベント統計の要約です。

	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。

WebSphere 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身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、WebSphere 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

[32 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、キュー・サイズ上限で決められている値以上のキュー・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。
- 取得操作でメッセージがキューから取り出された結果、キュー・サイズ下限で決められている値以下のキュー・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。
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:

キュー・サイズ下限で決められている値以下のキュー・サイズになる取得操作で、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時、キュー・サイズ下限イベントは生成されません。

IBM WebSphere MQ は、正常な取得操作の間のみ、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時、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、キュー・サイズ上限イベントおよびキュー・サイズ下限イベントはリセットされません。

### 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)
```

## 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 WebSphere MQ コマンド (MQSC) は次のとおりです。

```
ALTER QMGR PERFMEV(ENABLED)

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

## 関連概念

[30 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

[31 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

[34 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。

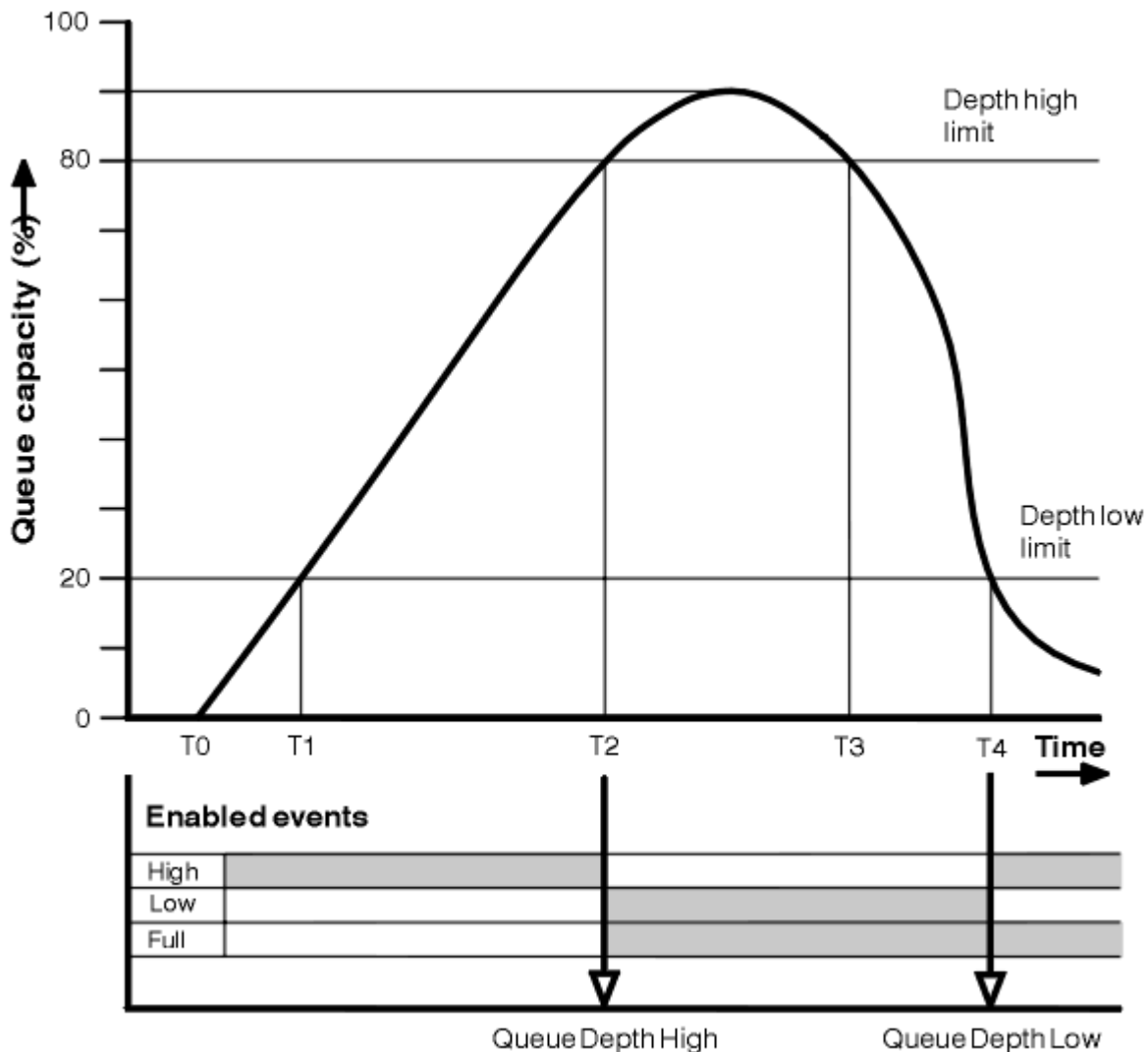


図 7. キュー・サイズ・イベント (1)

## 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)、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

35 ページの表 8 は、キュー・イベント統計の要約であり、35 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 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

36 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。

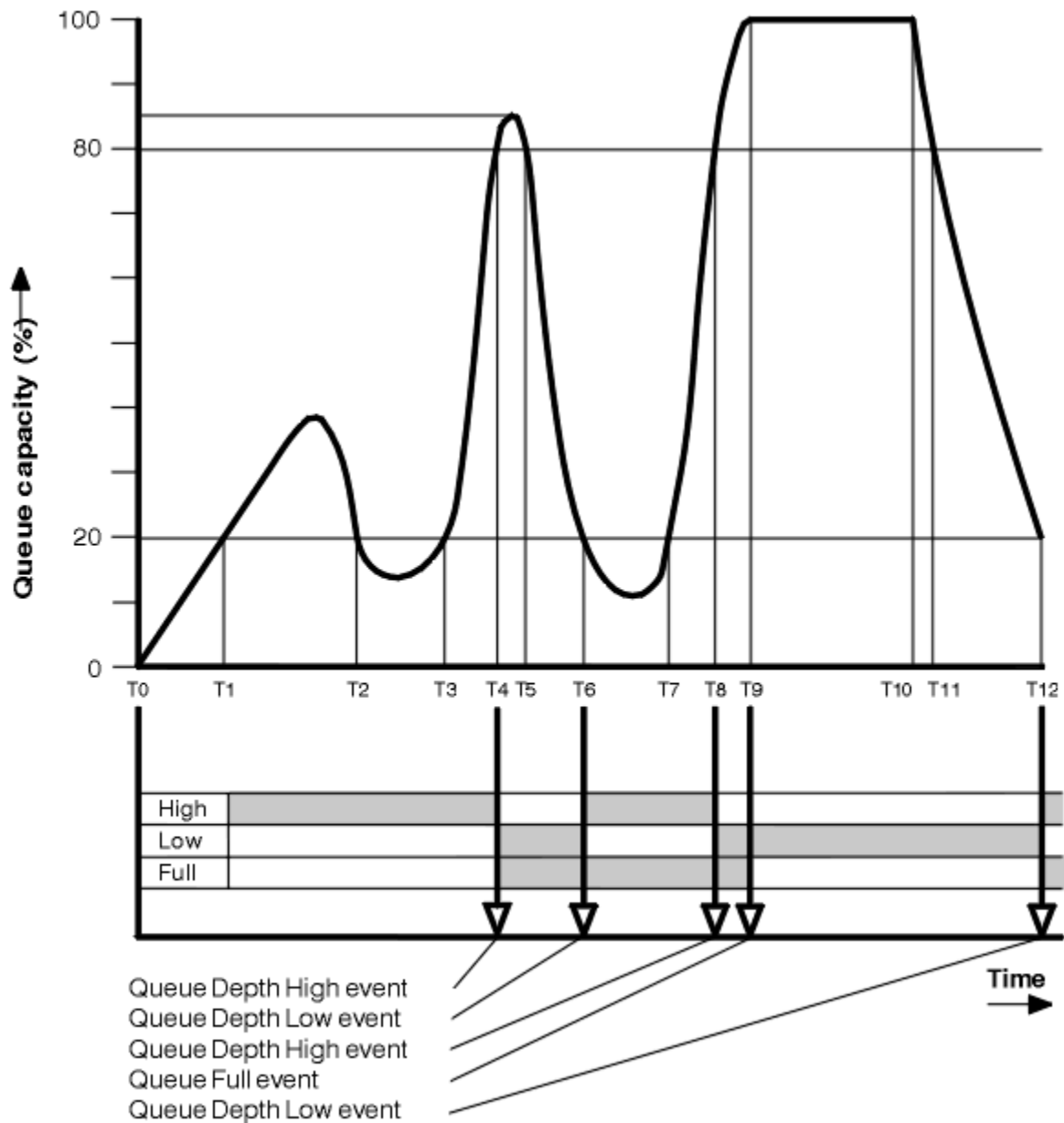


図 8. キュー・サイズ・イベント (2)

### 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

37 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、37 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 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。

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・データ内にも含まれます。

### 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。

オブジェクトの変更イベントには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

#### [13 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](#)

#### [8 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](#)

37 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](#)

## 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](#)

37 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](#)

## 「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

[37 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・データ内にのみ含まれます。

#### 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、分散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` に入れられます。

#### 関連資料

[コマンド](#)

[8 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

[41 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](#)

[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

### [41 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。

### [41 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](#)

## 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・エクステントへの書き込みを開始したという通知のことです。

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。

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 に入れます。

## 関連資料

### [ロガー](#)

### [8 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。

## 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・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。

**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、イベントを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

[43 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・エクステントへの書き込みを開始したという通知のことです。

## 関連資料

[LoggerEvent \(MQLONG\)](#)

[RESET QMGR](#)

## 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

[43 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・エクステントへの書き込みを開始したという通知のことです。

## 関連資料

[LoggerEvent \(MQLONG\)](#)

[43 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・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。

```
/*
/*****
*/
/* Program name: AMQSLOG0.C
*/
/*
*/
/* Description: Sample C program to monitor the logger event queue and output*/
/* a message to stdout when a logger event occurs
*/
/* <N_OCO_COPYRIGHT>
*/
/* Licensed Materials - Property of IBM
*/
*/
/* 63H9336
*/
/* (c) Copyright IBM Corp. 2005, 2024. All Rights Reserved.
*/
/*
*/
/* US Government Users Restricted Rights - Use, duplication or
*/
/* disclosure restricted by GSA ADP Schedule Contract with
*/
/* IBM Corp.
*/
/* <NOC_COPYRIGHT>
*/
/*****
*/
/*
*/
/* Function: AMQSLOG is a sample program which monitors the logger event
*/
/* queue for new event messages, reads those messages, and puts the contents
*/
/* of the message to stdout.
*/
*/
/*****
*/
/*
*/
/* AMQSLOG has 1 parameter - the queue manager name (optional, if not
*/
/* specified then the default queue manager is implied)
*/
*/
/*****
*/
/*****
*/
```

```

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

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

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

#define MAX_MESSAGE_LENGTH 8000

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

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

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

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

static PMQCHAR ParmToString(MQLONG Parameter);

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

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

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

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

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

    /*****/
    /* Connect to the specified (or default) queue manager
    /*****/

    MQCONN(QMName,
           &hConn,
           &CompCode,
           &Reason);

```

```

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

/* Open the logger event queue for input */

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)
{
    printf("MQOPEN failed for queue manager %.48s Queue %.48s Reason: %d\n",
        ObjDesc.ObjectQMgrName,
        ObjDesc.ObjectName,
        Reason);
    goto MOD_EXIT;
}
else
{
    ProcessPCF(hConn, hEventQueue, pBuffer);
}

MOD_EXIT:

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

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

return 0;
}

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

static void ProcessPCF(MQHCONN hConn,
                     MQHOBJ hEventQueue,
                     PMQCHAR pBuffer)
{
    MQCFH * pCfh;
    MQCFST * pCfst;
    MQGMO Gmo = { MQGMO_DEFAULT };
    MQMD Mqmd = { MQMD_DEFAULT };
    PMQCHAR pPCFCmd;
    MQLONG Reason = 0;
    MQLONG CompCode;
    MQLONG MsgLen;
    PMQCHAR Parm = NULL;

    /* Set timeout value */

    Gmo.Options |= MQGMO_WAIT;
    Gmo.Options |= MQGMO_CONVERT;
    Gmo.WaitInterval = MQWI_UNLIMITED;
    /*****
    /* Process response Queue */
    *****/

```

```

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

    MQGET( hConn,
           hEventQueue,
           &Mqmd,
           &Gmo,
           MAX_MESSAGE_LENGTH,
           pBuffer,
           &MsgLen,
           &CompCode,
           &Reason);
    if (Reason != MQCC_OK)
    {
        switch(Reason)
        {
            case MQRC_NO_MSG_AVAILABLE:
                printf("Timed out");
                break;

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

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

    /*****/
    /* Build the output by parsing the received PCF message, first the */
    /* header, then each of the parameters */
    /*****/

    pCfh = (MQCFH *)pBuffer;

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

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

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

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

    pPCFCmd = (char *) (pCfh+1);
    printf("-----\n");
    while(pCfh -> ParameterCount-- )
    {
        pCfst = (MQCFST *) pPCFCmd;
        switch(pCfst -> Type)
        {
            case MQCFT_STRING:
                Parm = ParmToString(pCfst -> Parameter);

```

```

        if (Parm != NULL) {
            printf("%-32s",Parm);
        }
        else
        {
            printf("%-32d",pCfst -> Parameter);
        }

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

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

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

```

## 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

### [44 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。

### [42 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

### 43 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。

## 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

このページでは、観測イベントをモニターするためのCプログラム例を示します。

このサンプル・プログラムは、IBM WebSphere MQ 製品の一部ではないため、実際の物理項目として提供されていません。指定した処置に対して考えられるすべての結果を列挙していないために、この例は完全なものではありません。ただし、特にイベント・メッセージに使用される PCF 形式を使用するユーザー固有のプログラムの基礎として、このサンプルを使用できます。ただし、このプログラムをご使用のシステム上で実行するには、事前に変更しておく必要があります。

```
/******  
/*  
/* Program name: EVMON  
/*  
/* Description: C program that acts as an event monitor  
/*  
/*  
/******  
/*  
/* Function:  
/*  
/* EVMON is a C program that acts as an event monitor - reads an  
/* event queue and tells you if anything appears on it  
/*  
/* Its first parameter is the queue manager name, the second is  
/* the event queue name. If these are not supplied it uses the  
/* defaults.  
/*  
/******  
#include <time.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#ifndef min  
#define min(a,b) ((a) < (b)) ? (a) : (b)  
#endif  
  
/******  
/* includes for MQI  
/******  
#include <cmqc.h>  
#include <cmqfc.h>  
void printfmqcfst(MQCFST* pmqcfst);  
void printfmqcfst(MQCFIN* pmqcfst);  
void printreas(MQLONG reason);  
  
#define PRINTREAS(param) \  
case param: \  
printf("Reason = %s\n",#param); \  
break;  
  
/******  
/* global variable  
/******  
MQCFH *evtmsg; /* evtmsg message buffer  
  
int main(int argc, char **argv)  
{  
/******  
/* declare variables  
/******  
int i; /* auxiliary counter  
/******  
/* Declare MQI structures needed  
/*
```

```

/*****
MQOD      od = {MQOD_DEFAULT};          /* Object Descriptor          */
MQMD      md = {MQMD_DEFAULT};          /* Message Descriptor         */
MQGMO     gmo = {MQGMO_DEFAULT};        /* get message options       */
/*****
/* note, uses defaults where it can
/*****

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

/*****
/* Connect to queue manager
/*****
QMName[0] = 0;                            /* default queue manager    */
if (argc > 1)
    strcpy(QMName, argv[1]);
MQCONN(QMName,                            /* queue manager            */
        &Hcon,                            /* connection handle        */
        &CompCode,                       /* completion code          */
        &CReason);                       /* reason code              */

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

/*****
/* Open the event queue for input; exclusive or shared. Use of
/* the queue is controlled by the queue definition here
/*****
O_options = MQOO_INPUT_AS_Q_DEF          /* open queue for input     */
            + MQOO_FAIL_IF_QUIESCING     /* but not if qmgr stopping */
            + MQOO_BROWSE;
MQOPEN(Hcon,                            /* connection handle        */
        &od,                            /* object descriptor for queue*/
        O_options,                       /* open options             */
        &Hobj,                          /* object handle            */
        &CompCode,                      /* completion code          */
        &Reason);                       /* reason code              */

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

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

```

```

memcpy(md.CorrelId, MQCI_NONE, sizeof(md.CorrelId));

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

/*****
/* report reason, if any */
/*****
if (Reason != MQRC_NONE && Reason != MQRC_TRUNCATED_MSG_ACCEPTED)
{
    printf("MQGET ==> %ld\n", Reason);
}
else
{
    gmo.Options = MQGMO_NO_WAIT + MQGMO_MSG_UNDER_CURSOR;
    buflen = evtmsglen; /* amount of message to get */
    evtmsg = malloc(buflen);
    if (evtmsg != NULL)
    {
        /*****
        /* clear selectors to get messages in sequence */
        /*****
        memcpy(md.MsgId, MQMI_NONE, sizeof(md.MsgId));
        memcpy(md.CorrelId, MQCI_NONE, sizeof(md.CorrelId));

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

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

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

    if (evtmsglen != buflen)
        printf("DataLength = %ld?\n", evtmsglen);
}

```

```

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

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

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

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

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

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

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

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

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

```

```

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

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

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

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

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

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

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

```

```

void printfmqcfst(MQCFIN* pmqcfst)
{
    switch (pmqcfst->Parameter)
    {
        case MQIA_APPL_TYPE:
            switch (pmqcfst->Value)
            {
                PRINTAT(MQAT_UNKNOWN)
                PRINTAT(MQAT_OS2)
                PRINTAT(MQAT_DOS)
                PRINTAT(MQAT_UNIX)
                PRINTAT(MQAT_QMGR)
                PRINTAT(MQAT_OS400)
                PRINTAT(MQAT_WINDOWS)
                PRINTAT(MQAT_CICS_VSE)
                PRINTAT(MQAT_VMS)
                PRINTAT(MQAT_GUARDIAN)
                PRINTAT(MQAT_VOS)
            }
            break;
        case MQIA_Q_TYPE:
            if (pmqcfst->Value == MQQT_ALIAS)
            {
                printf("MQIA_Q_TYPE is MQQT_ALIAS\n");
            }
            else
            {
                if (pmqcfst->Value == MQQT_REMOTE)
                {
                    printf("MQIA_Q_TYPE is MQQT_REMOTE\n");
                    if (evtmsg->Reason == MQRC_ALIAS_BASE_Q_TYPE_ERROR)
                    {
                        printf("but remote is not valid here\n");
                    }
                }
                else
                {
                    printf("MQIA_Q_TYPE is wrong, %ld\n",pmqcfst->Value);
                }
            }
            break;

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

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

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

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

void printreas(MQLONG reason)
{
    switch (reason)
    {
        PRINTREAS(MQRCFC_CFH_TYPE_ERROR)
        PRINTREAS(MQRCFC_CFH_LENGTH_ERROR)
        PRINTREAS(MQRCFC_CFH_VERSION_ERROR)
        PRINTREAS(MQRCFC_CFH_MSG_SEQ_NUMBER_ERR)
        :
        PRINTREAS(MQRC_NO_MSG_LOCKED)
        PRINTREAS(MQRC_CONNECTION_NOT_AUTHORIZED)
    }
}

```

```
PRINTREAS(MQRC_MSG_TOO_BIG_FOR_CHANNEL)
PRINTREAS(MQRC_CALL_IN_PROGRESS)
default:
    printf("It's an unknown reason, %ld\n",
           reason);
    break;
}
}
```

## 関連概念

### [6 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。

### [5 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。

## 関連資料

### [C プログラミング](#)

#### [44 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・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 WebSphere MQ 経路表示アプリケーション (dspmqrte)
- アクティビティ記録
- 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。
- WebSphere 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、WebSphere 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、WebSphere 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

#### 55 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。

#### 57 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。

#### 64 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ーを実行すると、アクティビティー・レポートが生成され、適切な場所に送信されます。アクティビティー・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ー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、アクティビティー・レポートを使用して収集されたアクティビティー情報を、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ーを実行するときに、アクティビティー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ー・レポートを生成することができます。アクティビティー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。

メリット	アクティビティ ー記録	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	Yes	Yes
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	Yes	Yes
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)	Yes	No
メッセージ・データは未変更のままになる	Yes	No
メッセージは通常どおり処理される	Yes	No
アクティビティ情報をメッセージ・データに累積可能	No	Yes
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	No	Yes
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	No	Yes
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	No	Yes
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	No	Yes

## 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。メッセージ経路の完全性は、以下のよう  
に、キュー・マネージャー・ネットワーク内のキュー・マネージャーのレベルによって異なります。

### WebSphere MQ バージョン 6.0 以降のリリースのキュー・マネージャー

WebSphere MQ バージョン 6.0 以降のリリースのキュー・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、メッセージに対して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・ネットワークが WebSphere MQ バージョン 6.0 以降のリリースのキュー・マネージャーでのみ構成されている場合は、完全なメッセージ経路を判別できます。

### WebSphere MQ バージョン 6.0 より前のキュー・マネージャー

WebSphere MQ バージョン 6.0 より前のキュー・マネージャーに接続されたアプリケーションは、メッセージに対して実行されたアクティビティを記録し**ません**。キュー・マネージャー・ネットワークにバージョン 6.0 以前の WebSphere MQ キュー・マネージャーが含まれている場合、メッセージ経路は一部のみ判別できます。

## アクティビティ情報の保管方法

WebSphere MQ では、アクティビティ情報は、アクティビティ・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・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ごとに、個別にアクティビティ・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

### 59 ページの『アクティビティ記録』

アクティビティ・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が記録されます。

### 64 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

### 70 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。

### 105 ページの『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。

## アクティビティ記録

アクティビティ・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が記録されます。

アクティビティ・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をそれぞれアクティビティ・レポートに記録することができます。アクティビティ・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・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記録されます。通常、アクティビティ・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・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・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・レポートを生成できます。アクティビティ・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・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・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で、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: WebSphere 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 WebSphere 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は生成された場合は 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。これはデフォルト値です。

## 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。

## 無効化

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相関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 WebSphere 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必要があります。
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ことにに関する詳細が示されているとします。アクティビティ・レポートに詳述されている最後の操作は、*OperationType* が *send* であり、メッセージがチャンネル CH1 を使用して宛先キュー・マネージャー QM1 に送信されたことの詳細を示しています。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、ります。

#### **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必要があります。

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もあります。

## 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。

- バージョン 6.0 より前の WebSphere MQ 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。



WebSphere 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

### 122 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、65 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、WebSphere 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。WebSphere 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、60 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。  
69 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

69 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 WebSphere MQ 経路表示アプリケーションを使用します。

## 関連タスク

### 60 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。値は次のいずれかです。

## 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。

## 無効化

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、[122 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 WebSphere 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。

## WebSphere MQ 経路表示アプリケーション

WebSphere 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。WebSphere 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、WebSphere MQ バージョン 6.0 以前のキュー・マネージャー、または WebSphere 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、WebSphere MQ バージョン 6.0 以降の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。

### Priority

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: WebSphere 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: WebSphere 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。75 ページの『追加のアクティビティ情報』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
Detail	MQCFIN
RecordedActivities	MQCFIN
UnrecordedActivities	MQCFIN
DiscontinuityCount	MQCFIN
MaxActivities	MQCFIN
累積	MQCFIN
Forward	MQCFIN
送信	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情報の詳細レベルを指定します。値は次のいずれかです。

#### 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の数を示します。アクティビティは、そのアクティビティについての情報が経路トレース・メッセージに書き込まれた場合、またはアクティビティ・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。値は次のいずれかです。

### 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 WebSphere MQ バージョン 6 (またはそれ以降の) 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 WebSphere MQ バージョン 6 (またはそれ以降の) 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 72 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 72 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 72 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 72 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 73 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。値は次のいずれかです。

#### **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. WebSphere 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. WebSphere 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る、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 WebSphere 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 WebSphere 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 WebSphere 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 WebSphere 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せん。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>

表 14. パラメーター名 (続き)	
説明	IBM WebSphere 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、 <b>ParameterValue</b> の値が取り込まれます。
値:	表示される名前。

### ParameterValue

表 15. パラメーター値	
説明	IBM WebSphere MQ 経路表示アプリケーションによって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:	<i>GroupName</i> .
値:	表示される値。

### 追加アクティビティ情報の記録例

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 WebSphere 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情報を表示します。

#### 例 1

追加のアクティビティ情報は、パラメーター ID が WebSphere 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。

- WebSphere 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、WebSphere 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。

- 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を実行します。標準的な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に関する詳細情報を提供しますが、パラメーター ID が WebSphere 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 WebSphere 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。

```
65536: 'Red'
```

WebSphere 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。

WebSphere 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、77 ページの『例 2』を参照してください。

## 例 2

追加のアクティビティ情報は、パラメーター ID が IBM WebSphere 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、ユーザー・アプリケーションによって記録されます。

1. IBM WebSphere MQ 経路表示アプリケーションを使用して、76 ページの『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を実行します。標準的な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 WebSphere MQ 経路表示アプリケーションによって表示される名前が含まれます。このアプリケーションは、 <i>ColorValue</i> の値をコンテキストに入れます。
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 WebSphere MQ 経路表示アプリケーションによって表示される値が含まれます。
ID:	65536

表 18. 色の値 (続き)	
説明	IBM WebSphere MQ 経路表示アプリケーションによって表示される値が含まれます。
データ型:	MQCFST
組み込まれる PCF グループ:	ColorInfo.
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 WebSphere 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作成されます。

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 WebSphere 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。

```
Color: 'Red'
```

IBM WebSphere MQ 経路表示アプリケーションは、追加アクティビティー情報の値を含む 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。

## WebSphere 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、WebSphere 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ー情報を処理します。

**注:** キュー・マネージャーに対してクライアント・アプリケーションを実行するには、クライアント接続機構をインストールする必要があります。

WebSphere MQ 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、アクティビティー情報を収集し、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。
- アクティビティー情報の記録にアクティビティー・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ー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ー情報が返された後で WebSphere 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、WebSphere 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ー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、WebSphere 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、WebSphere 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**

WebSphere 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、WebSphere 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**

WebSphere 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。名前は 48 文字以内で指定します。

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 WebSphere 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* で指定されません。

このパラメーターを指定しない場合は、WebSphere 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、WebSphere 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**

WebSphere 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、WebSphere 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 WebSphere 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)。

#### いいえ

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値を共有します。

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

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 WebSphere 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情報

このページを使用して、記録されたアクティビティ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・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。

### activity レポート・オプション。 -ro を使用して指定されます。

アクティビティ情報をアクティビティ・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・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 WebSphere 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 WebSphere 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が記録されます。さらに、パブリッシュ・アクティビティおよび MCA で実行されたアクティビティも記録されます。

##### **high**

**low** および **medium** で指定されているアクティビティが記録されます。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が記録されます。

デフォルトでは、IBM WebSphere 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 WebSphere 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 WebSphere 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。名前は 48 文字以内で指定します。

このパラメーターを指定しない場合は、IBM WebSphere 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。

<b>yes</b>	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。
<b>no</b>	経路トレース・メッセージは、到着時にターゲット・キューに書き込まれません。

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、[70 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。

#### **all**

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:** バージョン 6.0 より前の IBM WebSphere MQ 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・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。

これはデフォルト値です。

#### **いいえ**

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。

**discard レポート・オプション。** **-ro** を使用して指定されます。

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 WebSphere 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 WebSphere 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 WebSphere 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、アクティビティー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ー情報を表示するには、以下のパラメーターを指定する必要があります。

### **-q TargetQName**

IBM WebSphere 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ー情報を表示する場合は、アクティビティー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 WebSphere 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ー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ー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ー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ー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ー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 WebSphere 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 WebSphere 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。

### all

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。

#### 識別子

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 WebSphere MQ 経路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、75 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。

## WebSphere MQ 経路表示アプリケーションの例

次の例は、WebSphere MQ 経路表示アプリケーションの使用方法を示しています。それぞれの例には、2つのチャンネル(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、WebSphere 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 WebSphere 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。

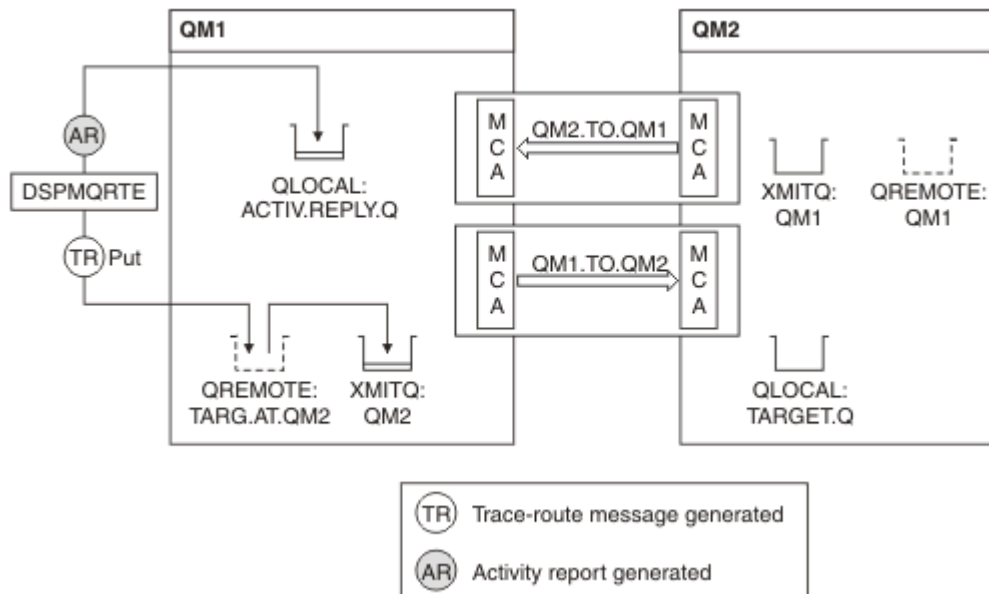


図 9. アクティビティ・レポートの要求、図 1

- 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 WebSphere 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。

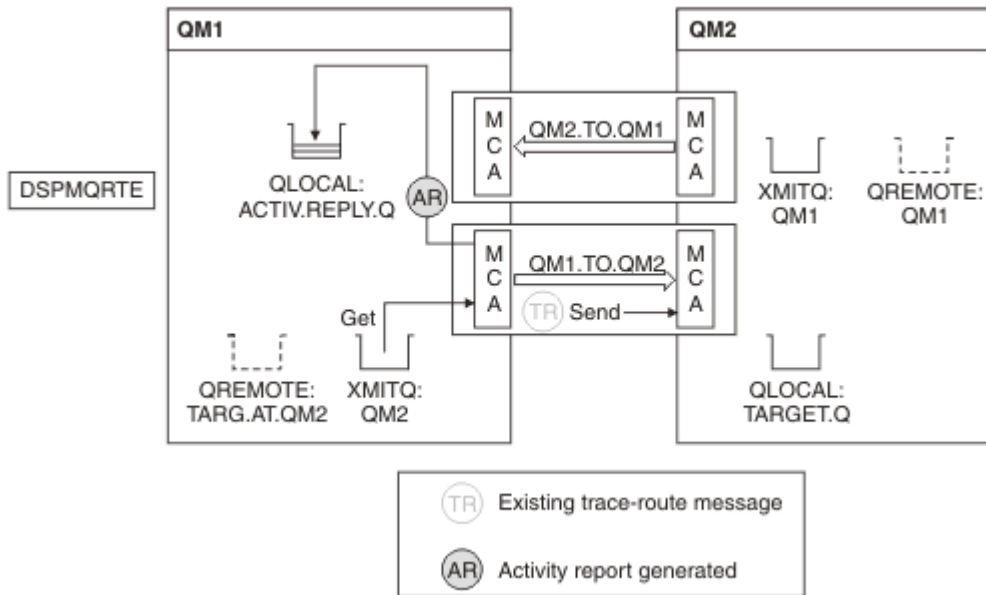


図 10. アクティビティ・レポートの要求、図 2

- 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 72 ページの『1』、72 ページの『4』、および 73 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。

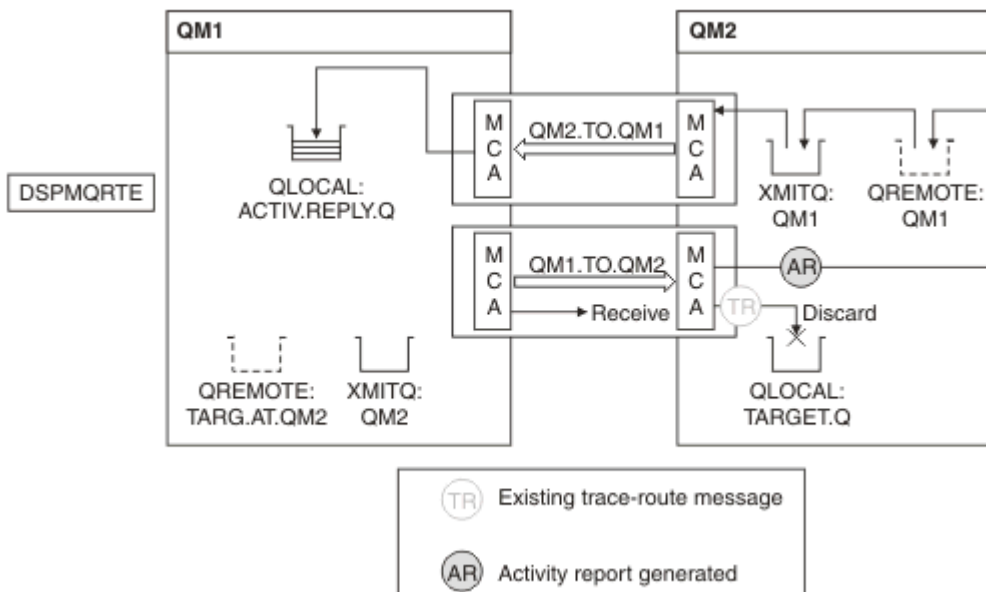


図 11. アクティビティ・レポートの要求、図 3

- 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 はアクティビティー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・マネージャーがバージョン 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 はアクティビティー・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ー・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、アクティビティー・レポートがキュー・マネージャー QM1 (ACTIV.REPLY.Q) に返されます。

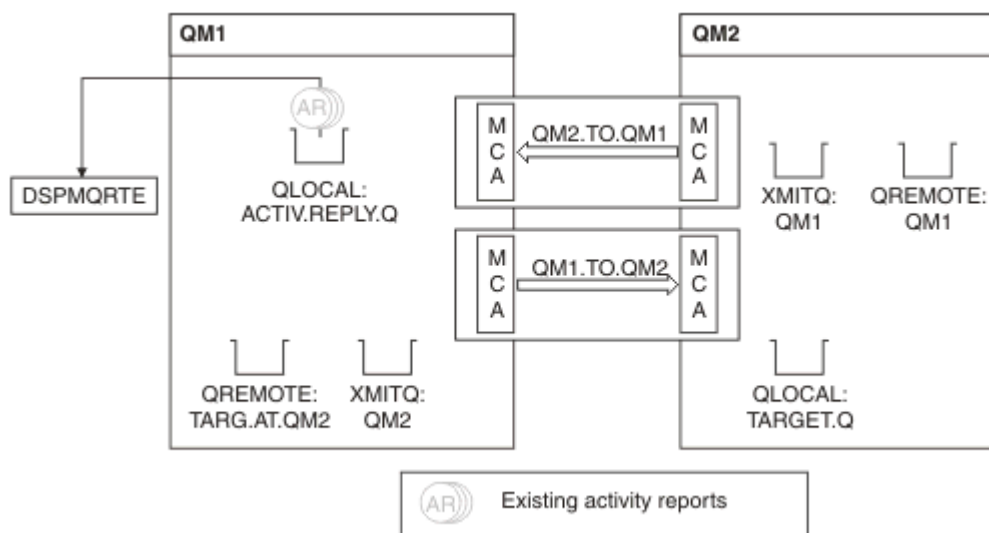


図 12. アクティビティー・レポートの要求、図 4

- 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、WebSphere 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。

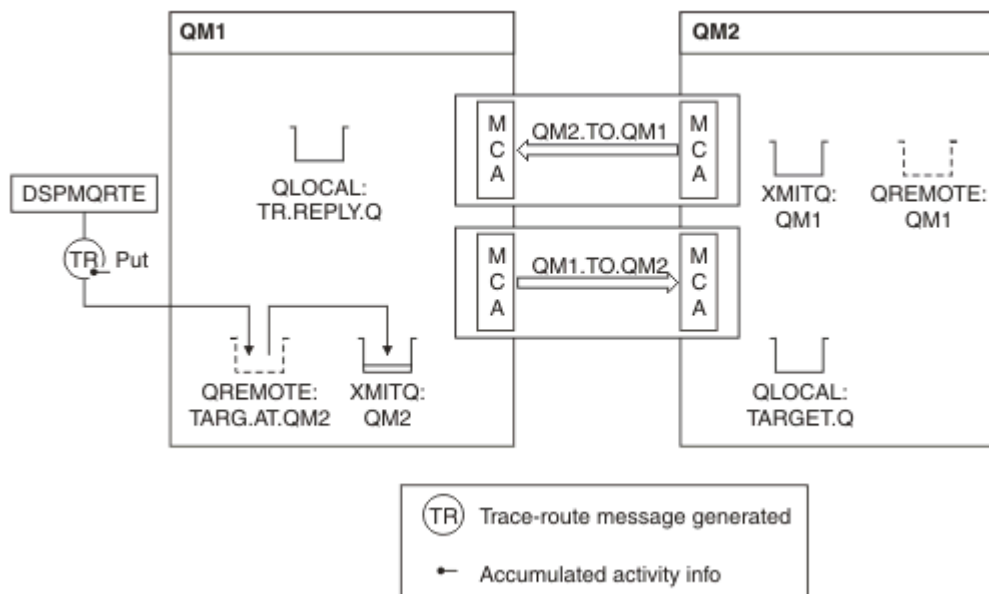


図 13. 経路トレース応答メッセージの要求、図 1

- 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 WebSphere 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・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。

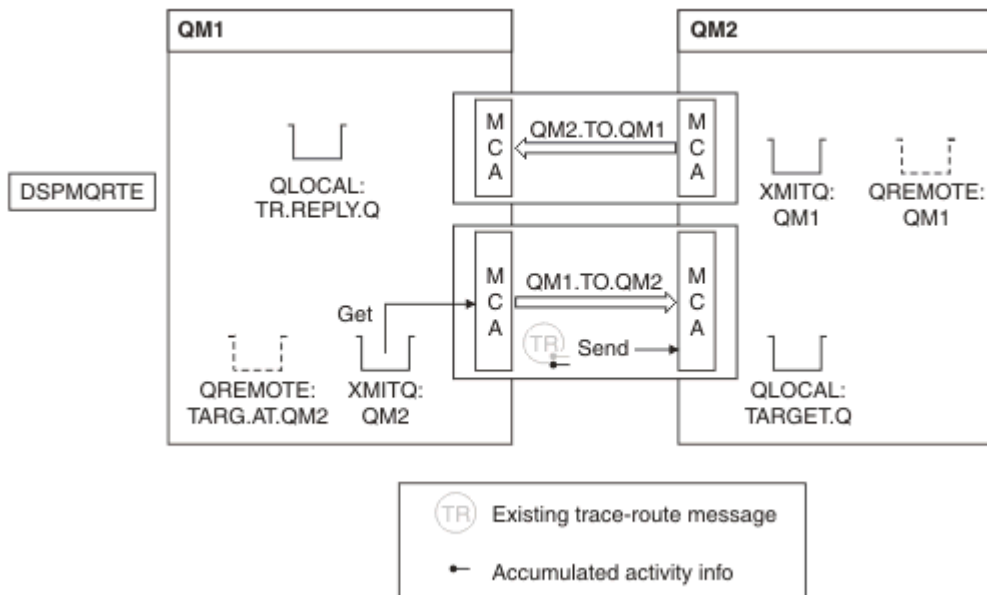


図 14. 経路トレース応答メッセージの要求、図 2

- 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。

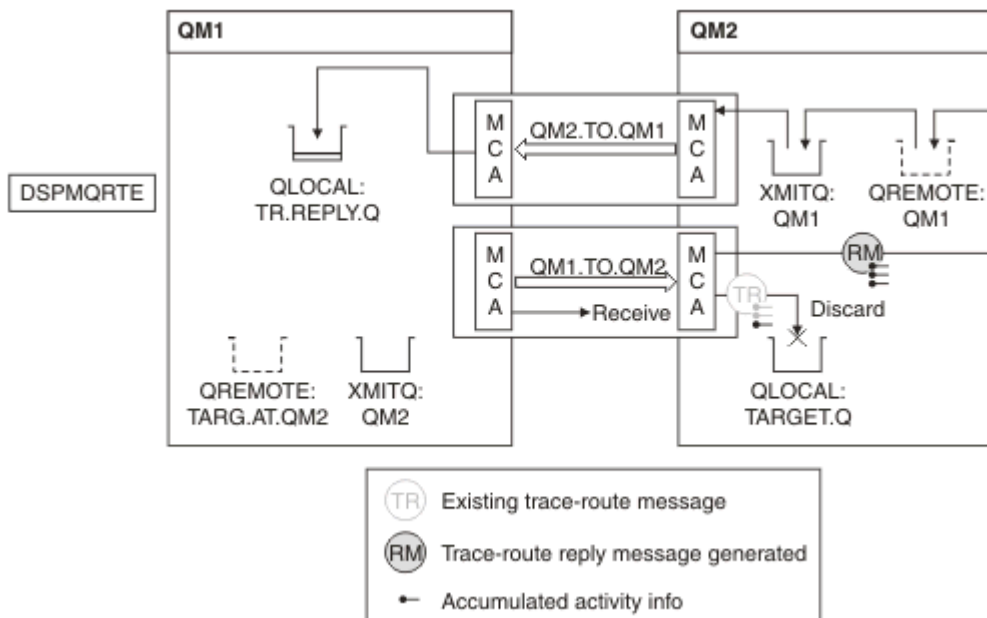


図 15. 経路トレース応答メッセージの要求、図 3

- 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。

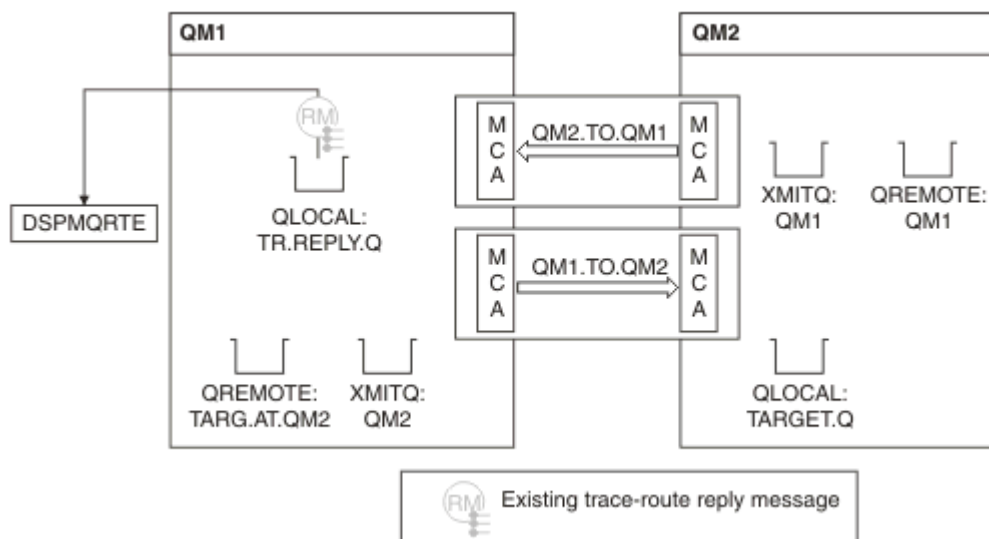


図 16. 経路トレース応答メッセージの要求、図 4

- 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ーを解析し、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ー・レポートの送信

アクティビティー・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、WebSphere 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ー・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、86 ページの『例 1 - アクティビティー・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ー・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、アクティビティー・レポートを別のキューから読み取るために WebSphere MQ 経路表示アプリケーションが使用されます。

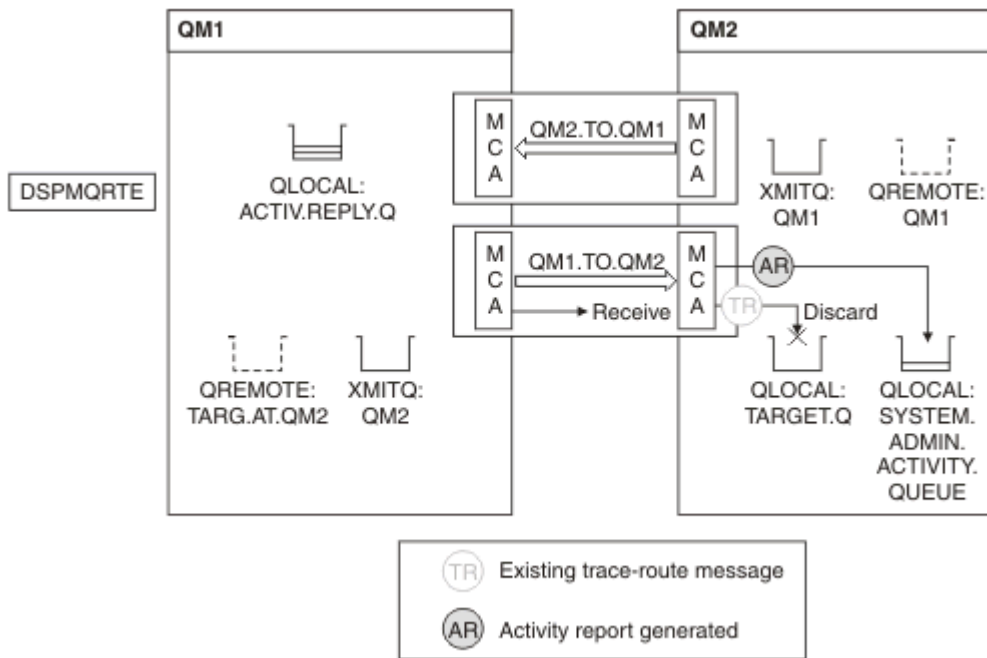


図 17. システム・キューへのアクティビティ・レポートの送信、図 1

- 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。

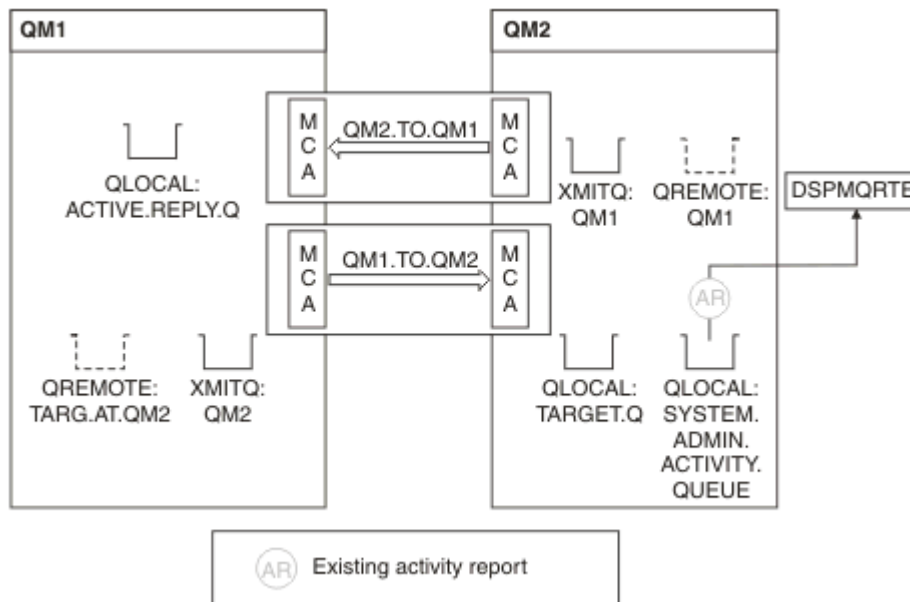


図 18. システム・キューへのアクティビティ・レポートの送信、図 2

- 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・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・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・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、WebSphere 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。

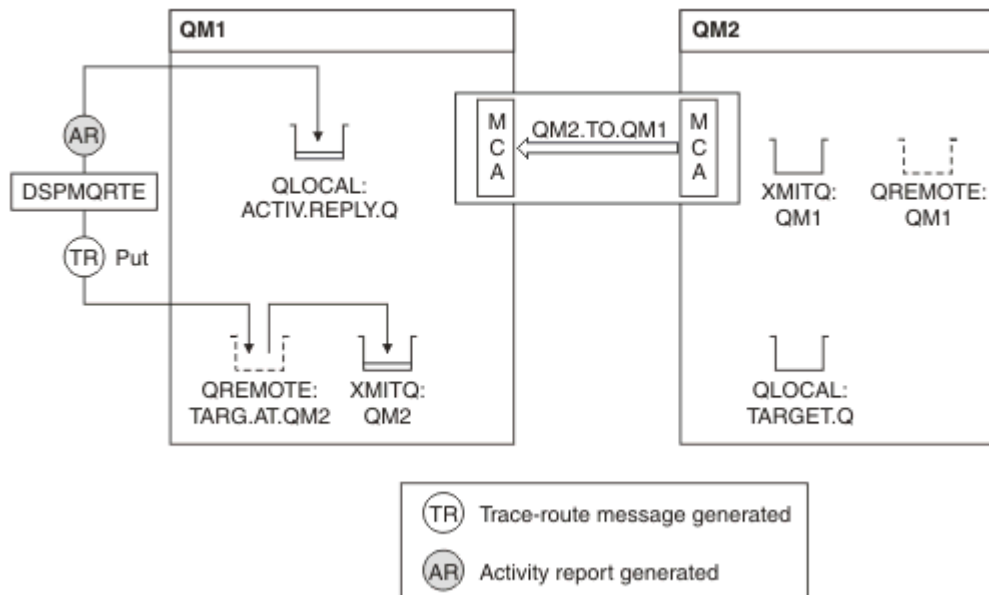


図 19. チャンネル問題の診断

- 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 WebSphere 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生成された場合は、*TraceRoute PCF* グループから構成されます。

97 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 Priority 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 <sup>8</sup> 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 <sup>1</sup> <sup>2 3</sup> <sup>7</sup> 解決されたキュー名 <sup>1</sup> <sup>3</sup> <sup>7</sup> リモート・キュー名 <sup>3</sup> <sup>7</sup> リモート・キュー・マネージャー名 <sup>2</sup> <sup>3 4</sup> <sup>5</sup> <sup>7</sup> サブスクリプション・レベル <sup>9</sup> サブスクリプション ID <sup>9</sup> フィードバック <sup>2</sup> <sup>10</sup> チャンネル名 <sup>4</sup> <sup>5 つの</sup> チャンネル・タイプ <sup>4</sup> <sup>5 つの</sup> 伝送キュー名 <sup>5</sup> トレース経路 <sup>6</sup> Detail 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 送信

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 のメッセージ記述子構造体。AIX®、HP-UX、z/OS、IBM i、Solaris、Linux、Windows、およびこれらのシステムに接続されているすべての WebSphere MQ MQI クライアントでサポートされます。

### **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ング・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\_CC SID\_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: [103 ページの『アクティビティ・レポート MQCFH \(PCF ヘッダー\)』](#)を参照してください。

## **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.

### **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: *ActivityApplName*  
*ActivityApplType*  
*ActivityDescription*  
*Operation*  
*TraceRoute*

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: *Activity*

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: *Activity*

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\_QUEUE  
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 グループ: MQMD または *EmbeddedMQMD*  
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 グループ: MQMD または *EmbeddedMQMD*  
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 グループ: MQMD または *EmbeddedMQMD*  
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 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。

### **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 グループ: MQMD または *EmbeddedMQMD*

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 WebSphere 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 *Operation*  
グループ:  
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 *Operation*  
グループ:  
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 *Operation*  
グループ:  
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 *Operation*  
グループ:  
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・アクティビティー内に含まれる。

### **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・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 グループ: *Operation*  
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 グループ: *Operation*  
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 グループ: *Operation*  
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 WebSphere 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)
- *Format* が MQFMT\_EMBEDDED\_PCF に設定されている場合は、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。

WebSphere 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。

123 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。

表 20. 経路トレース・メッセージ形式		
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 Priority 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 Detail 記録されたアクティビティー 記録されていないアクティビティー 不連続カウント 最大アクティビティー数 累積 送信

## 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。 *TraceRoute* 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。 *TraceRoute* 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なりません。

### **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リング・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: [128 ページの『経路トレース・メッセージ MQCFH \(PCF ヘッダー\)』を参照してください。](#)

## **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: **MQCFH\_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.

### **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 グループの部分を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  
                  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 WebSphere 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。

[133 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 Priority 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	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 <sup>1 2 3</sup> 解決されたキュー名 <sup>1 3</sup> リモート・キュー名 <sup>3</sup> リモート・キュー・マネージャー 2 3 4 5 フィードバック <sup>2</sup> チャンネル名 <sup>4 5</sup> チャンネル・タイプ <sup>4 5</sup> 伝送キュー名 <sup>5</sup> TraceRoute Detail 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 ー 不連続カウント 最大アクティビティ数 累積 送信

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ります。

### **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、105 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。

## **アカウントティング・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トティング・メッセージと統計メッセージを生成して、IBM WebSphere MQ アプリケーションによって実行される MQI 操作に関する情報を記録したり、IBM WebSphere MQ システムで発生するアクティビティに関する情報を記録したりします。

## アカウントティング・メッセージ

アカウントティング・メッセージは、IBM WebSphere 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。135 ページの『アカウントティング・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 WebSphere MQ システムで発生するアクティビティーに関する情報を記録するために使用されます。138 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。

アカウントティング・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。

## アカウントティング・メッセージ

アカウントティング・メッセージは、WebSphere 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。アカウントティング・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、アカウントティング・メッセージが生成され、システム・アカウントティング・キュー (SYSTEM.ADMIN.ACCOUNTING.QUEUE) に送信されます。長期間実行される WebSphere MQ アプリケーションでは、以下の場合に中間アカウントティング・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トティング・メッセージ以降の時間が、構成された間隔を超えるとき。

アカウントティング・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・アカウントティング・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。

アカウントティング・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、アカウントティング・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、アカウントティング・データは伝送キューに対して記録されます。

## 関連資料

### 152 ページの『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。

### 163 ページの『キュー・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、キュー・アカウントティング・メッセージの構造を確認します。

## アカウントティング・メッセージの形式

アカウントティング・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

- ・ アカウントティング・メッセージ MQMD (メッセージ記述子)

### アカウントティング・メッセージ・データ

- ・ アカウントティング・メッセージ MQCFH (PCF ヘッダー)
- ・ 常に返されるアカウントティング・メッセージ・データ
- ・ 可能な場合に返されるアカウントティング・メッセージ・データ

アカウントティング・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、アカウントティング・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。

アカウントティング・メッセージ・データは、アカウント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。アカウントティング・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。

アカウントティング・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、複数のアカウントティング・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。

## アカウント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、アカウント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 アカウントティング情報

キュー・マネージャー属性 ACCTMQI を使用して、MQI アカウントティング情報の収集を制御します。

この属性の値を変更するには、MQSC コマンド ALTER QMGR を使用して、パラメーター ACCTMQI を指定します。アカウントティング・メッセージは、アカウント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 アカウントティング情報が収集されます。

#### OFF

MQI アカウントティング情報は収集されません。これはデフォルト値です。

例えば、MQI アカウント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・アカウントティング情報

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・アカウント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。アカウントリング・メッセージは、アカウントリ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。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・アカウントリング情報が収集されます。

#### OFF

このキューのキュー・アカウントリング情報は収集されません。

#### QMGR

このキューのキュー・アカウントリング情報の収集は、キュー・マネージャー属性 ACCTQ の値に従って制御されます。これはデフォルト値です。

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・アカウントリ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・アカウントリング情報は収集されません。これはデフォルト値です。

#### NONE

キュー属性 ACCTQ とは無関係に、すべてのキューのキュー・アカウントリ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・アカウントリング情報の収集が無効になります。

例えば、キュー Q1 のアカウントリ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トリ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)
```

## 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 アカウントリング情報とキュー・アカウントリ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 アカウントリ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 アカウントリ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・アカウントリ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・アカウントリ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・アカウントリ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トリング・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)
```

## アカウントリング・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、アカウントリング・メッセージが生成されます。また、長期間実行されている WebSphere MQ アプリケーションについても、中間アカウントリング・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、アカウントリング・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トリング・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 WebSphere MQ アプリケーションについて中間アカウントリング・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、中間アカウントリング・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。アカウントリング・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に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、アカウントリ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トリング・メッセージを生成しません。

デフォルトのアカウントリング間隔は 1800 秒 (30 分) です。例えば、アカウントリング間隔を 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、WebSphere 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含まれています。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含まれています。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。

### 関連資料

#### 140 ページの『MQI 統計情報』

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。

#### 140 ページの『キュー統計情報』

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。

#### 141 ページの『チャンネル統計情報』

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。これらの属性は分散プラットフォームおよび IBM i で使用可能です。

## 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

- 統計メッセージング 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ー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) には *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 統計情報

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。

この属性の値を変更するには、MQSC コマンド *ALTER QMGR* を使用し、パラメーター *STATMQI* を指定します。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。これはデフォルト値です。

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)
```

### キュー統計情報

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 WebSphere 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。これはデフォルト値です。

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。これはデフォルト値です。

#### 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)
```

### **distributed** チャンネル統計情報

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。これらの属性は分散プラットフォームおよび IBM i で使用可能です。

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 WebSphere 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 STACLAS 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。[141 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。

レベル	説明	用途
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。

表 22. チャネル統計情報収集の詳細レベル (続き)

レベル	説明	用途
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。

**終**

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。

これはデフォルト値です。

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。

**終**

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。

これはデフォルト値です。

**NONE**

チャネル属性 `STATCHL` とは無関係に、すべてのチャネルのチャネル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 `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ネルの統計情報が、中程度の詳細レベルで収集されます。

終

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。

これはデフォルト値です。

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` を使用して、`STATINT` パラメーターを指定します。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。

## アカウントング情報と統計情報の表示

アカウントング・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、`amqsmn` 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。

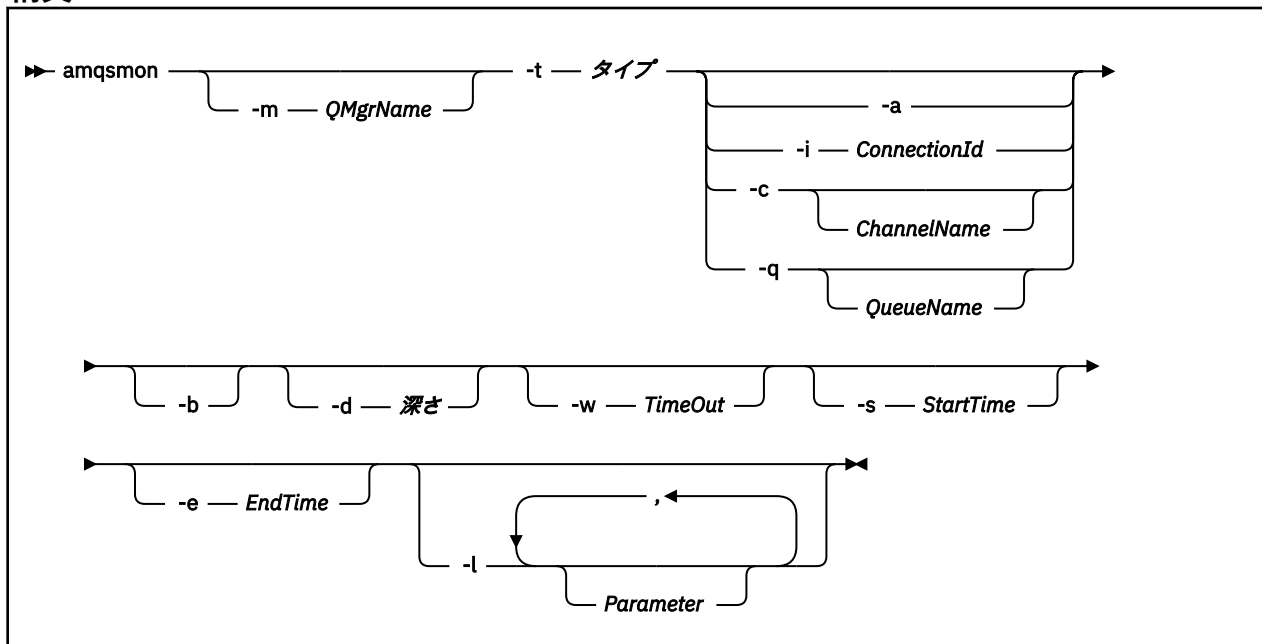
アカウントング・メッセージと統計メッセージは、システムのアカウントング・キューと統計キューに書き込まれます。`amqsmn` は、WebSphere MQ に用意されたサンプル・プログラムです。このサンプル・プログラムは、アカウントング・キューおよび統計キューのメッセージを処理し、読み取り可能な形式で情報を画面に表示します。

`amqsmn` 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、アカウントング・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、`amqsmn` 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、アカウンティング・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、アカウンティング・メッセージをアカウンティング・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。

##### accounting

アカウンティング・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 から読み取られます。

##### statistics

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 から読み取られます。

### オプション・パラメーター

#### -m QMgrName

アカウンティング・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  
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  
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*)。

### **-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、アカウントィング・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。

### **-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、amqsmn 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 の形式で指定されます。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 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. 次のコマンドは、キュー・マネージャー saturn.queue.manager からすべての 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, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 73, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqCount: [4, 2102, 0, 0, 0, 46, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [26, 1]
PutFailCount: 0
Put1Count: [40, 0]
Put1FailCount: 0
PutBytes: [57064, 12320]
GetCount: [18, 1]
GetBytes: [52, 12320]
GetFailCount: 2254
BrowseCount: [18, 60]
BrowseBytes: [23784, 30760]
BrowseFailCount: 9
CommitCount: 0
CommitFailCount: 0
BackCount: 0
ExpiredMsgCount: 0
PurgeCount: 0
```

2. 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 の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。

```
amqsmson -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. 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、アカウンティング・キューを参照し、MQI アカウンティ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 -a -l 7006,3024
```

このコマンドの出力は以下のとおりです。

```
ConnectionId: x'414d514354524556312020202020208d0b374203090020'
ApplicationName: 'runmqsc'

ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'
ApplicationName: 'amqsput'
```

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

## アカウントティング・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、アカウントティング・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。

アカウントティング・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 WebSphere MQ メッセージです。メッセージ・データには、WebSphere 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 WebSphere 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ティング・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ティング・メッセージ・データまたは統計メッセージ・データ

## アカウントティング・メッセージと統計メッセージの形式

このページを MQI アカウントティング・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 <sup>1</sup>
<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>

注:

- 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。

## 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:	<b>MQFMT_ADMIN</b> 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:	<b>MQCMD_ACCOUNTING_MQI</b> MQI アカウントティング・メッセージ。 <b>MQCMD_ACCOUNTING_Q</b> キュー・アカウントティング・メッセージ。 <b>MQCMD_STATISTICS_MQI</b> MQI 統計メッセージ。 <b>MQCMD_STATISTICS_Q</b> キュー統計メッセージ。 <b>MQCMD_STATISTICS_CHANNEL</b> 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:	<b>MQCFH_VERSION_3</b> 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 *ChlStatisticsData* 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 WebSphere 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: WebSphere 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 WebSphere 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。

### **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: 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\_CTL  
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\_CTLS\_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 WebSphere 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: WebSphere 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 WebSphere 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

### **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: キューのアカウント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: 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。

### **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。

## **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。

## **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 WebSphere 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: 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: 成功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リライブ要求の数。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: 失敗したサブスクリライブ要求の数。

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リライブ要求の数。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リライブ要求の数。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リライブ要求の数。

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\_CTL  
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\_CTL\_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 - 永続プロキシー・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 - 永続プロキシー・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 - 非永続プロキシー・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 - 非永続プロキシー・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\_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 WebSphere 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:  
*QName*  
*CreateDate*  
*CreateTime*  
*QType*  
*QDefinitionType*  
*QMinDepth*  
*QMaxDepth*  
*AvgTimeOnQ*  
*PutCount*  
*PutFailCount*  
*Put1Count*  
*Put1FailCount*  
*PutBytes*  
*GetCount*  
*GetFailCount*  
*GetBytes*  
*BrowseCount*  
*BrowseFailCount*  
*BrowseBytes*  
*NonQueuedMsgCount*  
*ExpiredMsgCount*  
*PurgeCount*  
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、WebSphere MQ がキューを迂回できた回数を表示します。

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。

### **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。

## **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 WebSphere 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 ChIStatisticsData 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: チャンネル属性の値 BATCHSZ または BATCHLIM に達したために送信された、チャンネルで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。

### **IncplBatchCount**

説明: チャンネル属性の値 BATCHSZ に達することなく送信された、チャンネルで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。

- [152 ページの『MQI アカウントティング・メッセージ・データ』](#)
- [163 ページの『キュー・アカウントティング・メッセージ・データ』](#)
- [173 ページの『MQI 統計メッセージ・データ』](#)
- [185 ページの『キュー統計メッセージ・データ』](#)
- [191 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、WebSphere 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。

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_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。
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、WebSphere 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 WebSphere 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・アクティビティー・トレースを使用します。

## アプリケーションのアクティビティー・トレース情報の収集

アプリケーションのアクティビティー・トレース・メッセージは PCF メッセージです。アクティビティー・トレースは、構成ファイルを使用して構成します。アプリケーションのアクティビティー・トレース情報を収集するには、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ー・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

アクティビティー・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。アクティビティー・トレースは、mqat.ini という構成ファイルを使用して構成します。

アプリケーションのアクティビティー・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. アプリケーションに対応するスタンザ (アクティビティー・トレース構成ファイル mqat.ini 内)。

前述の項目の順序は重要です。ACTVTRC 属性は ACTVCONO 設定によって指定変更され、ACTVCONO 設定は mqat.ini ファイル内の設定によって指定変更されます。

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。これらのパラメーターは両方とも、アクティビティ・トレース構成ファイル mqat.ini 内で指定します。

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。205 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、206 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. 198 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. 199 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. 200 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. 205 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・トレースの収集がオンになります

#### OFF

API アクティビティ・トレースの収集がオフになります

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。199 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ー・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ー・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、[206 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・アクティビティー・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。[205 ページの『アプリケーション・アクティビティー・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。

## アクティビティー・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・アクティビティー・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ー・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、アクティビティー・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ティ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、アクティビティー・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ー・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ー・トレースをオフに切り替えます。

### **MQCNO\_ ACTIVITY\_ TRACE\_ENABLED**

接続のアクティビティー・トレースをオンに切り替えます。

**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ー・トレース設定が、アクティビティー・トレース構成ファイル `mqat.ini` の設定によって指定変更されていないことを確認してください。

[200 ページの『mqat.ini を使用したアクティビティー・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ー・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、[206 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。205 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、mqs.ini ファイルおよび qm.ini ファイルと同じスタンザ・キー形式とパラメーター/値ペア形式に従います。

### このタスクについて

**Linux** **UNIX** UNIX and Linux システムの場合、mqat.ini はキュー・マネージャー・データ・ディレクトリーにあります。これは 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 にあります。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 Version 7.1 以前から移行されたキュー・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、mqat.ini ファイルを手動で作成し、ファイルにパーミッション 660 を設定する必要があります。

このファイルの形式の構文規則は次のとおりです。

- ・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 (**App1Name** と **DebugPath**) には大/小文字の区別があります。

#### スタンザ・キー

構成ファイルには、AllActivityTrace スタンザと ApplicationTrace スタンザの2つのタイプのスタンザ・キーがあります。

#### AllActivityTrace スタンザ

AllActivityTrace スタンザは、オーバーライドされない限りすべての IBM WebSphere MQ 接続に適用されるアクティビティ・トレースの設定を定義します。

AllActivityTrace スタンザの個々の値は、ApplicationTrace 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。

#### ApplicationTrace スタンザ

ApplicationTrace スタンザは、IBM WebSphere MQ 接続の特定の名前またはタイプ (あるいはその両方) に適用される可能性のある設定値を定義します。



このスタンザには、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る ApplName と ApplClass 値が含まれています。

### パラメーター/値ペア

次の表に、アクティビティー・トレースの構成ファイルで使用できるパラメーター/値のペアをリストします。

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	スタンザ・タイプ	値 (太字がデフォルト値)	説明
トレース	ApplicationTrace	<b>ON/OFF</b>	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。
ActivityInterval	AllActivityTrace ApplicationTrace	<b>0</b> から 99999999 ( <b>0 = オフ</b> )	トレース・メッセージ間の時間間隔 (秒単位)。アクティビティー・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・メッセージは時間が経過しても特定の瞬間に書き込まれることはなく、この時間間隔の経過後に最初の MQI 操作が実行されると書き込まれます。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・カウントに達したとき) に書き込まれます。
ActivityCount	AllActivityTrace ApplicationTrace	<b>0</b> から 99999999 ( <b>0 = オフ</b> )	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。

表 26. アクティビティ・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 (続き)

名前	スタンザ・タイプ	値 (太字がデフォルト値)	説明
TraceLevel	AllActivityTrace ApplicationTrace	LOW/ <b>MEDIUM</b> /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	AllActivityTrace ApplicationTrace	<b>0</b> から 104 857 600 (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)。
ApplName	ApplicationTrace	文字ストリング (必須パラメーター - デフォルトなし)	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。分散プラットフォームの場合、MQAXC.ApplName のファイル名部分だけが、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。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。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なります。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。

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 (続き)

名前	スタンザ・タイプ	値 (太字がデフォルト値)	説明
ApplClass	ApplicationTrace	<b>USER</b> /MCA/INTERNAL/ALL	アプリケーションのクラス。AppType 値と IBM WebSphere MQ 接続の対応関係の説明については、以下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 27. 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ネル (amqrmppa)
内部	MQXACT_EXTERNAL	MQXE_COMMAND_SERVER MQXE_MQSC	「runmqsc」およびコマンド・サーバー
内部	MQXACT_INTERNAL	(任意の値)	「信頼された」内部アプリケーションおよびプロセス。amqzdmaa など
ALL	(任意の値)	(任意の値)	すべてのユーザーと内部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、**amqspu**tc 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=amqspu      # Application name (may be wildcard)
                    # (matched to app name without path)
                    # Default: *
Trace=ON           # Activity trace switch for application
                  # Values: ( ON | OFF )
                  # Default: OFF
ActivityInterval=30 # Time interval between trace messages
                  # Values: 0-99999999 (0=off)
                  # Default: 0
ActivityCount=1    # Number of operations between trace msgs
                  # Values: 0-99999999 (0=off)
                  # Default: 0
TraceLevel=MEDIUM # Amount of data traced for each operation
                  # Values: LOW | MEDIUM | HIGH
                  # Default: MEDIUM
TraceMessageData=1000 # Amount of message data traced
                    # Values: 0-100000000
                    # Default: 0
    
```

### 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 WebSphere MQ 接続のタイプと突き合わされます。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「&#xa5;」) 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 ApplName からの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&#xa5;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 ApplName と一致していない場合、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 ApplName および ApplClass と一致している場合、最も具体的な一致である ApplName のスタンザが使用されます。最も具体的な一致の ApplName とは、接続の ApplName との一致に最も多くの文字が使用されるものと定義されています。例えば、ini ファイルに ApplName="FRE\*" というスタンザと ApplName="FREE\*" という別のスタンザが含まれる場合、ApplName="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 ApplName="FREE\*" になります (一方 ApplName="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 ApplName および ApplClass 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からの値が使用され、他のすべてのスタンザは無視されます。

## 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。この例は、C サンプル・ディレクトリー (amqsact.c file と同じディレクトリー) に mqat.ini というサンプルとして出荷されています。

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

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

## 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。205 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プを作成するか、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。200 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 199 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るか確認します。
- 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。200 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

## amqsact サンプル・プログラム

アプリケーション・アクティビティー・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、WebSphere MQ で提供されています。

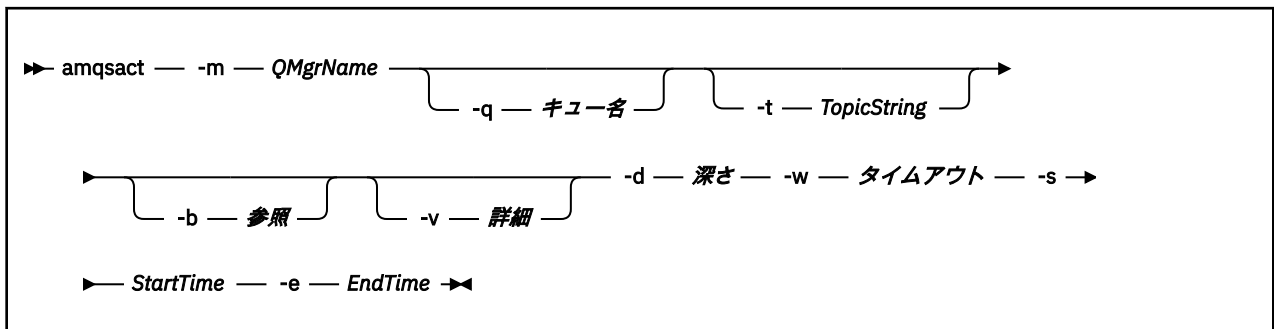
コンパイル済みプログラムが、以下のサンプル・ディレクトリーにあります。

- UNIX 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ー・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: 'MQ_1\bin\amqsput.exe'
Application Type: MQAT_WINDOWS_7
ApplicationPid: 14076
UserId: 'Emma_Bushby'
API Caller Type: MQXACT_EXTERNAL
API Environment: MQXE_OTHER
Application Function: ''
Appl Function Type: MQFUN_TYPE_UNKNOWN
Trace Detail Level: 2
Trace Data Length: 0
Pointer size: 4
Platform: MQPL_WINDOWS_7
MQI Operation: 0
Operation Id: MQXF_CONN
ApplicationTid: 1
OperationDate: '2014-03-15'
OperationTime: '12:08:10'
ConnectionId:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM '
00000010: FFFFFFFB5FFFFFFF6 4251 2000 FFFFFFFE601 '
QueueManager: 'TESTQM'
Completion Code: MQCC_OK
Reason Code: 0
```

## 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 WebSphere MQ メッセージです。メッセージ・データには、IBM WebSphere 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 WebSphere 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。  
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ーを識別します。  
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**

#### **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 WebSphere 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・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\_OTHER、MQXE\_MCA、MQXE\_MCA\_SVRCONN、MQXE\_COMMAND\_SERVER、MQXE\_MQSC

ID: MQIACF\_API\_ENVIRONMENT

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

### **Pointer Size**

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。値は MQPL\_\* 値の 1 つです。

ID: MQIA\_PLATFORM

データ型: MQCFIN

## アプリケーション・アクティビティ MQI 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 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"中"または"高"アクティビティ・トレースが構成されている場合のみ、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。

## 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

## **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)。

## **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\_STRUCT  
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\_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 (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)。

### **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\_STRUCT

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)

### **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

## **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

## **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)

### **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

### **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、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。(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)

## ***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

## **IntAttrs**

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

## **CharAttrs**

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

## **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

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<sup>12</sup>**

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<sup>12</sup>**

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<sup>12</sup>**

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<sup>12</sup>**

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 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
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 VLength 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、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
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

### 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

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

<sup>1</sup> 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。

<sup>2</sup> 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。

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。

表 28. AppActivityDistList グループ MQCFGR 構造体		
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。

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、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。

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

### **ResolvedLocalQName<sup>3</sup>**

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

### **ResolvedLocalQMgrName<sup>3</sup>**

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\_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)

## 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 AppCompatActivityDistList 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、237 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。AppCompatActivityDistList 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、AppCompatActivityDistList 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

### 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のみ組み込まれます。

PCF パラメーター: MQBACF\_GROUP\_ID

<sup>3</sup> 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。

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、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。  
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

#### **ResolvedLocalQName<sup>4</sup>**

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

#### **ResolvedLocalQMgrName<sup>4</sup>**

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: 代替セキュリティー 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 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

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、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

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)

## 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、[237 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。

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: 関連 ID。MQPMR 構造体が提供され、PutMsgRecFields に MQPMRF\_CORREL\_ID が含まれている場合のみ組み込まれます。

---

<sup>4</sup> 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: 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タ。  
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。

### **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。

### **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

### **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)

## 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 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"中"または"高"アクティビティ・トレースが構成されている場合のみ、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

## **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 WebSphere MQ システムの定常状態を理解するのを支援します。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。
- チャン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、チャンネルをモニターします。チャン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。

キューおよびチャン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ンネル・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ン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ン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ンネル、あるいは複数のキューまたはチャン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ンネルを制御するには、キュー属性 MONQ またはチャンネル属性 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ン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ンネル・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ンネルは WebSphere 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 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ンネルのチャンネル・モニターを有効にするか無効にするかが決まります。

チャン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。272 ページの表 29 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	用途
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。

```
ALTER QMGR MONACLS(QMGR)
```

## 関連概念

### [271 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。

### [キュー・マネージャーの処理](#)

## 関連タスク

### [273 ページの『キューおよびチャンネルのモニター・データの表示』](#)

キューまたはチャンネルのリアルタイム・モニター情報を表示するには、IBM WebSphere MQ エクスプローラーまたは該当の 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。

### [モニター \(MONCHL\)](#)

## キューおよびチャンネルのモニター・データの表示

キューまたはチャンネルのリアルタイム・モニター情報を表示するには、IBM WebSphere MQ エクスプローラーまたは該当の 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必要があります。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 WebSphere MQ エクスプローラーを使用するか、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. チャンネルのリアルタイム・モニター情報を表示するには、IBM WebSphere MQ エクスプローラーを使用するか、またはオプション・パラメーター MONITOR を指定して 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・マネージャーの属性 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・マネージャーの属性 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

[271 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

[DISPLAY QSTATUS](#)

# 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。

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 または相関 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。

## チャンネルのモニター

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ン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ンネル・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ン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ンネル・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 というチャンネルに関する問題を調査します。このチャンネルは、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 は送信側チャンネルまたはサーバー・チャンネルとして定義され、キュー・マネージャー QM2 は受信側チャンネルまたは要求側チャンネルとして定義されます。

### チャン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、チャン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ン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ンネルを確認します。

```
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)
```

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) 279 ページの表 30 を確認して、次に行うアクションを決定します。

開始側 MCA の副状態 <sup>1</sup>	応答側 MCA の副状態 <sup>2</sup>	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 <i>CONNAME</i> 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、281 ページの『 <a href="#">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</a> 』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。詳細については、281 ページの『 <a href="#">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</a> 』を参照してください。
RCVEXIT SENDEXIT MSGEXIT MREXIT	RCVEXIT SENDEXIT MSGEXIT MREXIT	出口は <i>MQXR_INIT</i> 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。詳細については、281 ページの『 <a href="#">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</a> 』を参照してください。
SERIALIZE	SERIALIZE	この副状態は、属性指定が <i>SHARED</i> 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のみ示されます。

表 30. 状況がバイディングまたは要求の場合に見られる副状態 (続き)		
開始側 MCA の副状態 <sup>1</sup>	応答側 MCA の副状態 <sup>2</sup>	注
SSLHANDSHAKE	SSLHANDSHAKE	SSL ハンドシェークは、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、これはハンドシェークにかかる時間に影響します。

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 276 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。

#### CURMSGs

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。



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場合 (CURMSGs の値がゼロ以外)、チャンネルはもう一方のチャンネルでバッチが確認されるのを待っている可能性があります。出力の SUBSTATE フィールドを調べて、[281 ページの表 31](#) を参照してください。

表 31. 送信側および受信側 MCA の副状態		
送信側 SUBSTATE	受信側 SUBSTATE	注
MQGET	受信	静止しているチャンネルの通常の状態。
送信	受信	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。
受信		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

このページを使用して、バッチの完了に時間がかかることがある理由を確認します。

### 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。以下の要因が、送信側チャンネルの待機時間に影響を与える可能性があります。

### 手順

- ネットワーク速度が低下しているかどうかの確認  
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。
- 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 (276 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

### 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 フィールドを調べてください。

#### 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。278 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。

## 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ン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、多数のチャン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。

## チャンネルの切り替えプロセス

チャン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. チャンネルは、入力用に新しい伝送キューを開き、(相関 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 WebSphere MQ では、**runswchl** コマンドを使用して実行されていない 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
- 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。

- **runswch1** コマンドを照会モードで実行して、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。

## Windows パフォーマンス・モニター

WebSphere MQ バージョン 7.0 以前では、Windows パフォーマンス・モニターを使用して、Windows システム上のローカル・キューのパフォーマンスをモニターすることができました。WebSphere MQ バージョン 7.1 からは、パフォーマンス・モニターのこのメソッドは使用可能ではありません。

[271 ページの『リアルタイム・モニター』](#)で説明されているメソッドを使用して、すべてのサポートされているプラットフォーム上のキューをモニターできます。

## 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。

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

103-8510

東京 103-8510、日本

**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。これらの名前はすべて架空のものであり、名前や住所が類似する個人や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 WebSphere MQ のサービスを使用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](http://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 の米国およびその他の国における商標です。

この製品には、Eclipse Project (<http://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: