

9.0

*IBM MQ* の構成

**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、[799 ページの『特記事項』](#)に記載されている情報をお読みください。

本書は、IBM® MQ バージョン 9 リリース 0、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, 2023.

# 目次

<b>構成</b> .....	<b>5</b>
マルチプラットフォームでのキュー・マネージャーの作成と管理.....	5
デフォルト・キュー・マネージャーの作成.....	8
既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする方法.....	10
キュー・マネージャーの作成後の構成ファイルのバックアップ.....	11
キュー・マネージャーの開始.....	12
キュー・マネージャーの停止.....	12
キュー・マネージャーの再始動.....	14
キュー・マネージャーの削除.....	15
サーバーとクライアント間の接続の構成.....	16
使用する通信タイプ.....	17
拡張トランザクション・クライアントの構成.....	19
MQI チャンネルの定義.....	30
AMQP チャンネルの作成および使用.....	31
異なるプラットフォーム上でのサーバー接続およびクライアント接続の定義の作成.....	36
サーバー側でのサーバー接続およびクライアント接続の定義の作成.....	39
MQI チャンネル用のチャンネル出口プログラム.....	48
キュー共有グループへのクライアントの接続.....	51
構成ファイルを使用したクライアントの構成.....	52
IBM MQ 環境変数の使用.....	74
IBM MQ およびキュー・マネージャー構成情報の変更.....	85
UNIX, Linux, and Windows での構成情報の変更.....	86
IBM i での構成情報の変更.....	94
IBM MQ 構成情報を変更するための属性.....	106
キュー・マネージャー構成情報の変更.....	112
分散キューイングの構成.....	140
IBM MQ 分散キューイング技法.....	140
分散キュー管理の概要.....	161
UNIX, Linux, and Windows 上のチャンネルのモニターおよび制御.....	191
IBM i 上のチャンネルのモニターおよび制御.....	214
キュー・マネージャー・クラスターの構成.....	236
パブリッシュ/サブスクライブ・メッセージングの構成.....	357
キューに入れられたパブリッシュ/サブスクライブのメッセージ属性の設定.....	358
キューに入れられたパブリッシュ/サブスクライブの開始.....	359
キューに入れられたパブリッシュ/サブスクライブの停止.....	360
ストリームの追加.....	360
ストリームの削除.....	361
サブスクリプション・ポイントの追加.....	362
分散パブリッシュ/サブスクライブ・ネットワークの構成.....	363
複数のインストールの構成.....	383
複数のインストール環境でのアプリケーションの接続.....	383
プライマリー・インストールの変更.....	393
キュー・マネージャーとインストールの関連付け.....	395
システム上の IBM MQ のインストールの見つけ方.....	396
高可用性、リカバリー、および再始動の構成.....	397
クライアントの自動再接続.....	398
コンソール・メッセージのモニター.....	404
高可用性の構成.....	408
ロギング: メッセージが失われないようにするための機能.....	528
IBM MQ キュー・マネージャー・データのバックアップと復元.....	555
クラスター・エラー・リカバリーの変更 (z/OS 以外のサーバー).....	563
JMS リソースの構成.....	565

JNDI ネーム・スペースでの接続ファクトリーおよび宛先の構成.....	566
IBM MQ Explorer を使用した JMS オブジェクトの構成.....	570
管理ツールを使用した JMS オブジェクトの構成.....	571
WebSphere Application Server での JMS リソースの構成.....	580
最新のリソース・アダプター保守レベルを使用するためのアプリケーション・サーバーの構成.....	592
JMS <b>PROVIDERVERSION</b> プロパティの構成.....	595
WebSphere Application Server 永続サブスクリプションの削除.....	603
IBM MQ Console および REST API の構成.....	605
セキュリティの構成.....	606
CSRF 保護の構成.....	606
HTTP ホスト名の構成.....	607
HTTP および HTTPS ポートの構成.....	609
応答タイムアウトの構成.....	610
自動始動の構成.....	611
ログインの構成.....	612
LTPA トークンの有効期間の構成.....	615
administrative REST API ゲートウェイの構成.....	616
messaging REST API の構成.....	617
REST API for MFT の構成.....	618
mqweb サーバーの JVM のチューニング.....	620
IBM MQ Console および REST API のインストール・コンポーネントのファイル構造.....	620
IBM MQ を Docker で構成.....	622
Linux システムでの Docker サポート.....	623
Docker を使用した独自の IBM MQ キュー・マネージャー・イメージの計画.....	623
Docker を使用したサンプル IBM MQ キュー・マネージャー・イメージの作成.....	624
別々のコンテナでのローカル・バインディング・アプリケーションの実行.....	627
Salesforce プッシュ・トピックおよびプラットフォーム・イベントで使用する IBM MQ の構成.....	630
IBM MQ Bridge to Salesforce の構成.....	632
Salesforce プラットフォーム・イベントのイベント・メッセージの作成.....	637
IBM MQ Bridge to Salesforce の実行.....	643
ブロックチェーンで使用するための IBM MQ の構成.....	644
IBM MQ Bridge to blockchain の構成ファイルの作成.....	646
IBM MQ Bridge to blockchain の実行.....	651
IBM MQ Bridge to blockchain クライアント・サンプルの実行.....	655
z/OS でのキュー・マネージャーの構成.....	657
z/OS でのキュー・マネージャーのカスタマイズ準備.....	657
IBM MQ for z/OS のセットアップ.....	661
z/OS でのキュー・マネージャーのテスト.....	719
他のキュー・マネージャーとの通信のセットアップ.....	728
IMS での IBM MQ の使用.....	759
CICS での IBM MQ の使用.....	767
言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用.....	768
IMS における OTMA 出口の使用.....	770
IBM z/OSMF を使用した IBM MQ の自動化.....	774
IBM MQ Advanced for z/OS VUE の構成.....	785
MFT エージェントからリモート z/OS キュー・マネージャーへの接続.....	786
ブロックチェーンで使用するための IBM MQ Advanced for z/OS VUE の構成.....	786
<b>特記事項.....</b>	<b>799</b>
プログラミング・インターフェース情報.....	800
商標.....	800

# IBM MQ の構成

1 台以上のコンピューターにキュー・マネージャーを 1 つ以上作成し、それらをビジネス・データが含まれるメッセージを処理するように開発システム、テスト・システム、実動システム上で構成します。

IBM MQ を構成する前に、「[IBM MQ 技術概要](#)」の IBM MQ の概念についてお読みください。計画で、IBM MQ 環境の計画方法についてお読みください。

IBM MQ でキュー・マネージャーとその関連リソースを作成、構成、管理するには、いくつかの異なる方式を使用できます。これらの方式には、コマンド行インターフェース、グラフィカル・ユーザー・インターフェース、および管理 API があります。これらのインターフェースについて詳しくは、[IBM MQ の管理](#)を参照してください。

キュー・マネージャーを作成、開始、停止、および削除する方法については、[5 ページの『マルチプラットフォームでのキュー・マネージャーの作成と管理』](#)を参照してください。

IBM MQ インストール済み環境とアプリケーションを結び付けるのに必要なコンポーネントを作成する方法については、[140 ページの『分散キューイングの構成』](#)を参照してください。

さまざまな方式でクライアントを IBM MQ サーバーに接続する方法については、[16 ページの『サーバーとクライアント間の接続の構成』](#)を参照してください。

キュー・マネージャー・クラスターを構成する方法については、[236 ページの『キュー・マネージャー・クラスターの構成』](#)を参照してください。

構成情報を変更することによって、IBM MQ またはキュー・マネージャーの動作を変更できます。詳しくは、[85 ページの『IBM MQ およびキュー・マネージャー構成情報の変更』](#)を参照してください。一般に、この製品資料で示されている場合を除いて、構成変更を有効にするためにキュー・マネージャーを再始動する必要はありません。

 [IBM MQ for z/OS® の構成方法については、657 ページの『z/OS でのキュー・マネージャーの構成』](#)を参照してください。

## 関連タスク

[657 ページの『z/OS でのキュー・マネージャーの構成』](#)

IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

## 関連情報

[IBM MQ の技術概要](#)

[ローカル IBM MQ オブジェクトの管理](#)

[リモート IBM MQ オブジェクトの管理](#)

 [IBM i の管理](#)

 [管理 IBM MQ for z/OS](#)

[計画](#)

 [z/OS での IBM MQ 環境の計画](#)

## **マルチプラットフォームでのキュー・マネージャーの作成と管理**

メッセージおよびキューを使用するには、事前に少なくとも 1 つのキュー・マネージャーとそれに関連するオブジェクトを作成し、開始しておく必要があります。キュー・マネージャーは、関連したリソース、特にキュー・マネージャー自体が所有しているキューを管理します。キュー・マネージャーは、メッセージ・キューイング・インターフェース (MQI) 呼び出しおよびコマンドで IBM MQ オブジェクトを作成、修正、表示、および削除できるように、アプリケーションに対してキューイング・サービスを提供します。

## 始める前に

**重要:** IBM MQ は、スペースを含むマシン名をサポートしていません。スペースが使用されたマシン名を持つコンピューター上に IBM MQ をインストールすると、キュー・マネージャーを作成することはできません。

キュー・マネージャーを作成する前に、特に実稼働環境で考慮が必要な点がいくつかあります。以下のチェックリストを使用して調べてください。

### キュー・マネージャーとインストール済み環境の関連付け

キュー・マネージャーを作成するには、IBM MQ 制御コマンド `crtmqm` を使用します。`crtmqm` コマンドは、キュー・マネージャーを、その `crtmqm` コマンドを発行したインストール済み環境と自動的に関連付けます。キュー・マネージャー上で作動するコマンドの場合は、そのキュー・マネージャーと関連付けられたインストール済み環境から、そのコマンドを発行する必要があります。キュー・マネージャーに関連付けられたインストール済み環境は、`setmqm` コマンドを使用して変更できます。

Windows インストーラーは、インストールを実行するユーザーを `mqm` グループに追加しないことに注意してください。詳しくは、[UNIX, Linux®, and Windows 上の IBM MQ を管理する権限](#)を参照してください。

### 命名規則

すべてのプラットフォーム上のキュー・マネージャーと通信できるようにするため、大文字名を使用します。入力どおりに名前が割り当てられることに注意してください。あまりタイプ入力しなくても済むように、不必要に長い名前を使用しないでください。

### 固有のキュー・マネージャー名の指定

キュー・マネージャーを作成するときには、ネットワーク内のどこにも同じ名前のキュー・マネージャーがないことを確かめる必要があります。キュー・マネージャー名は、作成時には検査されません。固有の名前でないと、分散キューイング用のチャンネルを作成することができなくなります。また、パブリッシュ/サブスクライブ・メッセージング用にネットワークを使用する場合、サブスクリプションは、それを作成したキュー・マネージャー名と関連付けられています。このため、クラスターまたは階層内に同じ名前のキュー・マネージャーが存在する場合、結果としてパブリケーションがそれらに到達しない可能性があります。

確実に固有の名前になるようにする方法の 1 つは、各キュー・マネージャー名に、接頭部として固有のノード名を付けることです。例えば、ノードの名前が `ACCOUNTS` の場合は、キュー・マネージャーの名前を `ACCOUNTS.SATURN.QUEUE.MANAGER` にすることもできます。ここで、`SATURN` は特定のキュー・マネージャーを識別し、`QUEUE.MANAGER` はすべてのキュー・マネージャーに与えることができる拡張子です。この拡張子は省略することもできますが、その場合、`ACCOUNTS.SATURN` と `ACCOUNTS.SATURN.QUEUE.MANAGER` とは異なるキュー・マネージャー名になることに注意してください。

他社との通信のために IBM MQ を使用する場合には、自社名を接頭部として組み込むこともできます。これは、例には示されていません。この方法を一層理解しにくくしてしまうためです。

**注:** 制御コマンドでは、キュー・マネージャー名は大文字小文字の区別がされます。つまり、`jupiter.queue.manager` という名前と `JUPITER.queue.manager` という名前の 2 つのキュー・マネージャーを作成できるということです。しかし、複雑な問題点もあります。

### キュー・マネージャー数の制限

リソースに余裕がある限り、いくつでもキュー・マネージャーを作成できます。ただし、各キュー・マネージャーは、独自のリソースを必要とするため、それぞれが 10 個のキューを持つ 10 個のキュー・マネージャーを作成するよりも、ノードに 100 個のキューを持つ 1 個のキュー・マネージャーを作成する方が一般的には有効です。

実動システムでは、1 つのキュー・マネージャーで多数のプロセッサを活用できますが、大規模なサーバー・マシンでは複数のキュー・マネージャーを利用する方が効果的に運用できます。

### デフォルト・キュー・マネージャーの指定

各ノードにデフォルト・キュー・マネージャーを 1 つずつ設ける必要があります。ただし、デフォルト・キュー・マネージャーを持たない IBM MQ をノード上に構成することは可能です。デフォルト・キュー・マネージャーとは、アプリケーションが `MQCONN` 呼び出しでキュー・マネージャー名を指定しなかった場合に、そのアプリケーションが接続されるキュー・マネージャーのことです。また、これ

は、キュー・マネージャー名を指定せずに `runmqsc` コマンドを呼び出したときに MQSC コマンドを処理するキュー・マネージャーでもあります。

あるキュー・マネージャーをデフォルト・キュー・マネージャーとして指定すると、ノードについての既存のデフォルト・キュー・マネージャーの指定は置き換えられます。

デフォルトのキュー・マネージャーを変更すると、その他のユーザーまたはアプリケーションに影響を与える可能性があります。変更しても、現在接続されているアプリケーションに影響はありません。それらのアプリケーションは、以後の MQI 呼び出しで元の接続呼び出しからのハンドルを使用できるためです。このハンドルにより、呼び出しは確実に同じキュー・マネージャーに対して出されます。デフォルト・キュー・マネージャーを変更した後に接続されるアプリケーションは、新しいデフォルト・キュー・マネージャーに接続されます。こうしたことを行う場合には、デフォルト・キュー・マネージャーを変更する場合はその前に、上記の事柄を考慮する必要があります。

デフォルトのキュー・マネージャーの作成については、[8 ページの『デフォルト・キュー・マネージャーの作成』](#)で説明されています。

### 送達不能キューの指定

送達不能キューとは、正しい宛先に送ることができなかったメッセージを入れるローカル・キューのことです。

ネットワークの各キュー・マネージャーが送達不能キューを持っていることが重要です。送達不能キューを定義しない場合、アプリケーション・プログラム内にエラーが起こると、チャンネルがクローズされ、管理コマンドへの応答が受信されないことがあります。

例えば、アプリケーションが別のキュー・マネージャーのキューにメッセージを入れようとしたときに、誤ったキュー名が与えられると、チャンネルが停止し、メッセージは伝送キューに残ったままになります。その結果、他のアプリケーションは、このチャンネルを使用して自分のメッセージを送ることができなくなります。

キュー・マネージャーに送達不能キューがある場合には、チャンネルは影響を受けません。未配布メッセージは受信側の送達不能キューに入れられ、チャンネルとその伝送キューはそのまま使用できます。

キュー・マネージャーを作成するときには、**-u** フラグを使用して送達不能キューの名前を指定する必要があります。MQSC コマンドを使用することによって、既に定義したキュー・マネージャーの属性を変更し、使用する送達不能キューを指定することもできます。MQSC コマンドの ALTER の例については、[キュー・マネージャーの処理](#)を参照してください。

### デフォルト伝送キューの指定

伝送キューとは、リモート・キュー・マネージャーへ送られる途中のメッセージが、伝送されるまでの間キューイングされるローカル・キューのことです。デフォルト伝送キューとは、伝送キューが明示的に定義されなかった場合に使用されるキューのことです。各キュー・マネージャーには、1つのデフォルト伝送キューを割り当てることができます。

キュー・マネージャーを作成するときには、**-d** フラグを使用してデフォルト伝送キューの名前を指定する必要があります。これによって実際にキューが作成されるわけではありません。後で明示的にキューを作成する必要があります。詳しくは、[ローカル・キューの取り扱い](#)を参照してください。

### 必要なロギング・パラメーターの指定

`crtmqm` コマンドには、ロギングのタイプ、およびログ・ファイルのパスとサイズなどのロギング・パラメーターを指定できます。

開発環境では、デフォルト・ロギング・パラメーターで十分です。しかし、例えば、次のような場合にはデフォルトを変更できます。

- 大きなログをサポートできないロー・エンドのシステム構成である場合
- 同時に多数の長いメッセージがキューに留まることが予想される場合
- 多数の持続メッセージがキュー・マネージャーを介して渡されることが予想される場合

一度ロギング・パラメーターを設定すると、一部のロギング・パラメーターは、キュー・マネージャーを削除し、異なるロギング・パラメーターを持つ同じ名前のキュー・マネージャーを再作成してからでなければ変更できません。

ロギング・パラメーターの詳細については、397 ページの『高可用性、リカバリー、および再始動の構成』を参照してください。

### UNIX IBM MQ for UNIX システムにのみ該当します

**crtmqm** コマンドを使用する前に、別個のローカル・ファイル・システム上であっても、キュー・マネージャー・ディレクトリー `/var/mqm/qmgrs/qmgr` を作成できます。**crtmqm** の使用時に、`/var/mqm/qmgrs/qmgr` ディレクトリーが存在し、それが空で、mqm によって所有されている場合、そのディレクトリーはキュー・マネージャーのデータ用に使用されます。そのディレクトリーが mqm によって所有されていない場合は、その作成は失敗して、First Failure Support Technology (FFST) メッセージが出されます。そのディレクトリーが空でない場合は、新たにディレクトリーが作成されます。

## このタスクについて

キュー・マネージャーを作成するには、IBM MQ 制御コマンド **crtmqm** を使用します。詳しくは、**crtmqm** を参照してください。**crtmqm** コマンドは、必要なデフォルト・オブジェクトおよびシステム・オブジェクトを自動的に作成します (システム・デフォルト・オブジェクトで説明)。デフォルト・オブジェクトは、作成されるすべてのオブジェクト定義の基礎になります。システム・オブジェクトはキュー・マネージャーの操作に必要です。

**Windows** Windows システムでは、**crtmqm** コマンドの **sax** オプションを使用して、キュー・マネージャーの複数のインスタンスを開始するオプションがあります。

キュー・マネージャーとそのオブジェクトが作成されている場合は、**strmqm** コマンドを使用してそのキュー・マネージャーを開始します。

## 手順

- キュー・マネージャーの作成および管理に役立つ情報については、以下のサブトピックを参照してください。
  - 8 ページの『デフォルト・キュー・マネージャーの作成』
  - 10 ページの『既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする方法』
  - 11 ページの『キュー・マネージャーの作成後の構成ファイルのバックアップ』
  - 12 ページの『キュー・マネージャーの開始』
  - 12 ページの『キュー・マネージャーの停止』
  - 14 ページの『キュー・マネージャーの再始動』
  - 15 ページの『キュー・マネージャーの削除』

## 関連タスク

85 ページの『IBM MQ およびキュー・マネージャー構成情報の変更』

IBM MQ または個々のキュー・マネージャーの動作を、インストール済み環境の要件に合わせて変更できます。

657 ページの『z/OS でのキュー・マネージャーの構成』

IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

## 関連情報

[QM1 という名前前のキュー・マネージャーの作成](#)

[システムおよびデフォルト・オブジェクト](#)

[crtmqm](#)

## Multi デフォルト・キュー・マネージャーの作成

デフォルト・キュー・マネージャーとは、アプリケーションが MQCONN 呼び出しでキュー・マネージャー名を指定しなかった場合に、そのアプリケーションが接続されるキュー・マネージャーのことです。また、これは、キュー・マネージャー名を指定せずに **runmqsc** コマンドを呼び出したときに MQSC コマンドを処

理するキュー・マネージャーでもあります。キュー・マネージャーを作成するには、IBM MQ 制御コマンド `crtmqm` を使用します。

## 始める前に

デフォルト・キュー・マネージャーを作成する前に、5 ページの『マルチプラットフォームでのキュー・マネージャーの作成と管理』で説明されている考慮事項を参照してください。

**UNIX** UNIX で `crtmqm` を使用してキュー・マネージャーを作成する際に、`/var/mqm/qmgrs/qmgr` ディレクトリーが既に存在し、`mqm` によって所有され、空の場合、このディレクトリーはキュー・マネージャー・データ用に使用されます。ディレクトリーが `mqm` によって所有されていない場合、キュー・マネージャーの作成は失敗し、First Failure Support Technology (FFST) メッセージが表示されます。ディレクトリーが空でない場合は、キュー・マネージャー・データ用に新たにディレクトリーが作成されます。

この考慮事項は、`/var/mqm/qmgrs/qmgr` ディレクトリーが別のローカル・ファイル・システムに既に存在する場合でも適用されます。

## このタスクについて

`crtmqm` コマンドを使用してキュー・マネージャーを作成すると、このコマンドによって、必要なデフォルト・オブジェクトとシステム・オブジェクトが自動的に作成されます。デフォルト・オブジェクトは、作成するすべてのオブジェクト定義の基礎になります。システム・オブジェクトはキュー・マネージャーの操作に必要です。

このコマンドに関連するパラメーターを組み込むことにより、例えば、キュー・マネージャーが使用するデフォルト伝送キューの名前、および送達不能キューの名前を定義することもできます。

**Windows** Windows では、`crtmqm` コマンドの `sax` オプションを使用して、キュー・マネージャーの複数インスタンスを開始できます。

`crtmqm` コマンドとその構文について詳しくは、「[crtmqm](#)」を参照してください。

## 手順

- デフォルトのキュー・マネージャーを作成するには、`-q` フラグを指定して `crtmqm` コマンドを使用します。

`crtmqm` コマンドの以下の例では、`SATURN.QUEUE.MANAGER` というデフォルトのキュー・マネージャーが作成されます。

```
crtmqm -q -d MY.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE SATURN.QUEUE.MANAGER
```

ここで、

### **-q**

このキュー・マネージャーがデフォルト・キュー・マネージャーであることを示します。

### **-d MY.DEFAULT.XMIT.QUEUE**

このキュー・マネージャーによって使用されるデフォルト伝送キューの名前です。

**注:** IBM MQ では、デフォルト伝送キューは自動的に作成されません。ユーザー自身がデフォルト伝送キューを定義する必要があります。

### **-u SYSTEM.DEAD.LETTER.QUEUE**

インストール時に IBM MQ によって作成されたデフォルト送達不能キューの名前です。

### **SATURN.QUEUE.MANAGER**

このキュー・マネージャーの名前です。 `crtmqm` コマンドでは、これが最後のパラメーターでなければなりません。

## 次のタスク

キュー・マネージャーとそのオブジェクトが作成されている場合は、**strmqm** コマンドを使用してそのキュー・マネージャーを開始します。

### 関連タスク

11 ページの『[キュー・マネージャーの作成後の構成ファイルのバックアップ](#)』

IBM MQ 構成情報は、UNIX, Linux, and Windows 上の構成ファイルに保管されています。キュー・マネージャーの作成後、構成ファイルをバックアップしてください。その後、問題の原因となる別のキュー・マネージャーを作成した場合には、問題の原因を取り除いたときにバックアップを復元させることができます。

### 関連情報

[キュー・マネージャーの処理](#)

[ローカル・キューの処理](#)

[システムおよびデフォルト・オブジェクト](#)

## Multi 既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする方法

テキスト・エディターを使用するか、IBM MQ Explorer を使用して (Windows および Linux の場合)、既存のキュー・マネージャーを手動でデフォルトのキュー・マネージャーにすることができます。

### このタスクについて

既存のキュー・マネージャーをデフォルト・キュー・マネージャーにするためにテキスト・エディターを使用するには、以下の手順を実行します。

**Windows** **Linux** Windows および Linux (x86 および x86-64 プラットフォーム) システムで、IBM MQ Explorer を使用してこの変更を行う場合は、11 ページの『[IBM MQ Explorer を使用してキュー・マネージャーをデフォルトにする方法](#)』を参照してください。

デフォルト・キュー・マネージャーを作成すると、その名前が IBM MQ 構成ファイル (mqs.ini) 内の DefaultQueueManager スタンザの Name 属性に挿入されます。スタンザとその内容は、存在しない場合は自動的に作成されます。

### 手順

- 既存のキュー・マネージャーをデフォルトにするには、Name 属性のキュー・マネージャー名を、新しいデフォルト・キュー・マネージャーの名前に変更します。これは、テキスト・エディターを使用して手動で行うことができます。
- ノード上にデフォルト・キュー・マネージャーがなく、既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする場合、ユーザー自身、必要な名前を含む *DefaultQueueManager* スタンザを作成します。
- 別のキュー・マネージャーを誤ってデフォルトのキュー・マネージャーにしてしまったため、元のデフォルト・キュー・マネージャーに戻す場合、mqs.ini 内の DefaultQueueManager スタンザを編集し、誤ってデフォルト・キュー・マネージャーにしたキュー・マネージャーの名前を、希望するデフォルト・キュー・マネージャーの名前に置き換えます。

### 関連タスク

85 ページの『[IBM MQ およびキュー・マネージャー構成情報の変更](#)』

IBM MQ または個々のキュー・マネージャーの動作を、インストール済み環境の要件に合わせて変更できません。

## IBM MQ Explorer を使用してキュー・マネージャーをデフォルトにする方法

Windows Linux (x86 および x86-64 プラットフォーム) システムでは、IBM MQ Explorer を使用して既存のキュー・マネージャーをデフォルトのキュー・マネージャーにすることができます。

### このタスクについて

Windows および Linux (x86 および x86-64 プラットフォーム) システムで IBM MQ Explorer を使用して既存のキュー・マネージャーをデフォルトのキュー・マネージャーにするには、以下の手順を実行します。

テキスト・エディターを使用してこの変更を手動で行うには、[10 ページの『既存のキュー・マネージャーをデフォルト・キュー・マネージャーにする方法』](#)を参照してください。

### 手順

1. IBM MQ Explorer を開きます。
2. 「**IBM MQ**」を右クリックして、「**プロパティ...**」を選択します。「**IBM MQ のプロパティ (Properties for IBM MQ)**」パネルが表示されます。
3. 「**デフォルト・キュー・マネージャー名**」フィールドに、デフォルトのキュー・マネージャーの名前を入力します。
4. 「**OK**」をクリックします。

## キュー・マネージャーの作成後の構成ファイルのバックアップ

IBM MQ 構成情報は、UNIX, Linux, and Windows 上の構成ファイルに保管されています。キュー・マネージャーの作成後、構成ファイルをバックアップしてください。その後、問題の原因となる別のキュー・マネージャーを作成した場合には、問題の原因を取り除いたときにバックアップを復元させることができます。

### このタスクについて

一般に、新しいキュー・マネージャーを作成するたびに、構成ファイルのバックアップを取る必要があります。

構成ファイルには、次の 2 つのタイプがあります。

- プロダクトをインストールするときに、IBM MQ 構成ファイル (mqc.ini) が作成されます。これには、キュー・マネージャーのリストが含まれます。このファイルは、キュー・マネージャーを作成したり、削除したりするたびに更新されます。1 ノードに 1 つの mqc.ini ファイルがあります。
- 新しいキュー・マネージャーを作成するときに、新しいキュー・マネージャー構成ファイル (qm.ini) が自動的に作成されます。これには、そのキュー・マネージャーの構成パラメーターが含まれます。

AMQP サービスをインストールした場合は、以下に示す追加の構成ファイルをバックアップする必要があります。

- **Windows** Windows システムの場合: amqp\_win.properties
- **Linux** **UNIX** UNIX および Linux システムの場合: amqp\_unix.properties

### 関連タスク

[85 ページの『IBM MQ およびキュー・マネージャー構成情報の変更』](#)

IBM MQ または個々のキュー・マネージャーの動作を、インストール済み環境の要件に合わせて変更できます。

[555 ページの『IBM MQ キュー・マネージャー・データのバックアップと復元』](#)

ハードウェア障害に起因する破壊からキュー・マネージャーを保護できます。そのためには、キュー・マネージャーとキュー・マネージャー・データをバックアップするか、キュー・マネージャーの構成のみをバックアップするか、バックアップ・キュー・マネージャーを使用します。

## Multi キュー・マネージャーの開始

キュー・マネージャーを作成する際は、それを開始して、コマンドまたは MQI 呼び出しを処理できるようにしなければなりません。

### このタスクについて

**strmqm** コマンドを使用してキュー・マネージャーを開始できます。**strmqm** コマンドとそのオプションの説明については、[strmqm](#) を参照してください。

**Windows** **Linux** 代わりに、Windows および Linux (x86 および x86-64 プラットフォーム) システムでは、IBM MQ Explorer を使用してキュー・マネージャーを開始することができます。

**Windows** Windows では、IBM MQ Explorer を使用してシステムが始動したときに、自動的にキュー・マネージャーを始動することができます。詳しくは、[IBM MQ Explorer による管理](#)を参照してください。

### 手順

- **strmqm** コマンドを使用してキュー・マネージャーを開始するには、コマンドに続いて、開始するキュー・マネージャーの名前を入力します。  
例えば、QMB という名前のキュー・マネージャーを開始するには、次のコマンドを入力します。

```
strmqm QMB
```

**注:** **strmqm** コマンドは、作業対象のキュー・マネージャーに関連付けられたインストール済み環境から使用する必要があります。 **dspmq -o installation** コマンドを使用して、どのインストール済み環境にキュー・マネージャーが関連付けられているかを調べることができます。

**strmqm** コマンドは、キュー・マネージャーが開始して、接続要求を受け入れる用意ができるまで、制御を戻しません。

- **Windows** **Linux** IBM MQ Explorer を使用してキュー・マネージャーを開始するには、以下のステップを完了します。
  - a) IBM MQ Explorer を開きます。
  - b) ナビゲーター・ビューでキュー・マネージャーを選択します。
  - c) 「開始」をクリックします。

### タスクの結果

キュー・マネージャーが開始します。

キュー・マネージャーの開始に数秒より長い時間がかかると、開始の進行状況の詳細を示す情報メッセージが IBM MQ から断続的に発行されます。

## Multi キュー・マネージャーの停止

**endmqm** コマンドを使用して、キュー・マネージャーを停止できます。このコマンドでは、制御または静止状態でのシャットダウン、即時シャットダウン、およびプリエンティブ・シャットダウンの3つの方法でキュー・マネージャーを停止できます。または、Windows および Linux では、IBM MQ Explorer を使用してキュー・マネージャーを停止できます。

### このタスクについて

**endmqm** コマンドで単一インスタンス・キュー・マネージャーを停止する方法は3つあります。

## 制御 (静止) 状態でのシャットダウン

デフォルトでは、**endmqm** コマンドが指定されたキュー・マネージャーの静的シャットダウンを実行します。静止状態でのシャットダウンは、接続されたアプリケーションすべてが切断されるまで待機するため、完了するまで時間がかかる場合があります。

## 即時シャットダウン

即時シャットダウンの場合、現在の MQI 呼び出しを完了することはできますが、新しい呼び出しは失敗します。このタイプのシャットダウンは、アプリケーションがキュー・マネージャーに接続中でも実行されます。

## プリエンティブ・シャットダウン

キュー・マネージャーは即時に停止します。このタイプのシャットダウンは、例外的な状況でのみ使用します。例えば、キュー・マネージャーが通常の **endmqm** コマンドで停止しない場合などです。

**endmqm** コマンドは、単一インスタンスのキュー・マネージャーを停止する場合と同じ方法で、複数インスタンス・キュー・マネージャーのすべてのインスタンスを停止します。**endmqm** は、アクティブ・インスタンス、または複数インスタンス・キュー・マネージャーの 1 つのスタンバイ・インスタンスのいずれかで発行できます。ただし、キュー・マネージャーを終了するには、アクティブ・インスタンスで **endmqm** を実行する必要があります。

**endmqm** コマンドとそのオプションについて詳しくは、[endmqm](#) を参照してください。

**ヒント:** キュー・マネージャーのシャットダウンにおける問題は、アプリケーションによって頻繁に引き起こされます。例えば、次のような場合です。

- アプリケーションが MQI 戻りコードを正しく検査しない場合
- アプリケーションが静止の通知を要求しない場合
- アプリケーションが (MQDISC 呼び出しを出して)、キュー・マネージャーからの切断を行わずに終了する場合

キュー・マネージャーの停止中に問題が発生した場合は、Ctrl-C を使用して **endmqm** コマンドを中断してください。その後、別の **endmqm** コマンドを発行できますが、この場合は、必要なタイプのシャットダウンを指定するパラメーターを付加します。

 **endmqm** コマンドを使用する代わりに、on Windows and Linux, を使用して IBM MQ Explorer キュー・マネージャを停止し、制御されたシャットダウンまたは即時シャットダウンを実行することができます。

## 手順

- **endmqm** コマンドを使用してキュー・マネージャーを停止するには、コマンドに続いて必要に応じてパラメーターを入力し、停止するキュー・マネージャーの名前を入力します。

**注:** **endmqm** コマンドは、作業対象のキュー・マネージャーに関連付けられたインストール済み環境から使用する必要があります。キュー・マネージャーがどのインストール済み環境に関連付けられているかを調べるには、コマンド **dspm q -o installation** を使用します。

- 制御 (静止) 状態でのシャットダウンを実行するには、以下の例に示すように **endmqm** コマンドを入力します。これにより、QMB というキュー・マネージャーが停止します。

```
endmqm QMB
```

または、以下の例に示すように、**-c** パラメーターを指定して **endmqm** コマンドを入力することは、**endmqm QMB** コマンドと同等です。

```
endmqm -c QMB
```

どちらの場合も、制御は即時にユーザーに戻り、キュー・マネージャーが停止した時点は通知されません。すべてのアプリケーションが停止してキュー・マネージャーが終了するまでコマンドを待機

してから制御をユーザーに戻す場合は、以下の例に示すように、代わりに **-w** パラメーターを使用します。

```
endmqm -w QMB
```

- 即時シャットダウンを実行するには、以下の例に示すように、**-i** パラメーターを指定して **endmqm** コマンドを入力します。

```
endmqm -i QMB
```

- プリエンプティブ・シャットダウンを実行するには、以下の例に示すように、**-p** パラメーターを指定して **endmqm** コマンドを入力します。

```
endmqm -p QMB
```



**重要**：プリエンプティブ・シャットダウンは、接続されているアプリケーションに予測不能な結果を及ぼす可能性があります。このオプションは、通常の **endmqm** コマンドを使用したキュー・マネージャーを停止する他の試みがすべてした場合を除いて使用しないでください。

**ULW** プリエンプティブ・シャットダウンが機能しない場合は、代わりに手動でキュー・マネージャーを停止してみてください。

- 自動クライアント再接続を要求するには、**endmqm** コマンドを入力して **-e** パラメーターを指定します。このパラメーターには、クライアントがキュー・マネージャー・グループ内の他のキュー・マネージャーへの接続を再確立する効果があります。

**注**：デフォルトの **endmqm** コマンドを使用してキュー・マネージャーを終了しても、クライアントの自動再接続はトリガーされません。

- 複数インスタンス・キュー・マネージャーのアクティブ・インスタンスをシャットダウンした後にスタンバイ・インスタンスに移行するには、複数インスタンス・キュー・マネージャーのアクティブ・インスタンスで **endmqm** コマンドを入力して **-s** パラメーターを指定します。
- 複数インスタンス・キュー・マネージャーのスタンバイ・インスタンスを終了して、アクティブ・インスタンスの実行を続けるには、複数インスタンス・キュー・マネージャーのスタンバイ・インスタンスで **endmqm** コマンドを入力して **-x** パラメーターを指定します。

**Windows** **Linux**

Windows および Linux で IBM MQ Explorer を使用してキュー・マネージャーを停止するには、以下の手順を実行します。

- IBM MQ Explorer を開きます。
- ナビゲーター・ビューからキュー・マネージャーを選択します。
- 「停止」をクリックします。  
「キュー・マネージャーの終了」パネルが表示されます。
- 「制御」または「即時」を選択します。
- 「OK」をクリックします。  
キュー・マネージャーが停止します。

#### 関連情報

Windows での複数インスタンスのキュー・マネージャーへの保守レベル・アップデートの適用

UNIX および Linux での複数インスタンスのキュー・マネージャーへの保守レベル・アップデートの適用

## Multi キュー・マネージャーの再始動

**strmqm** コマンドを使用してキュー・マネージャーを再始動できます。Windows および Linux x86-64 システムでは、IBM MQ Explorer からキュー・マネージャーを再始動することもできます。

## このタスクについて

**strmqm** コマンドを使用してキュー・マネージャーを再始動できます。**strmqm** コマンドとそのオプションの詳細については、[strmqm](#) を参照してください。

**Windows** **Linux** Windows および Linux x86-64 システムでは、キュー・マネージャーを開始するときと同じ方法で IBM MQ Explorer を使用して、キュー・マネージャーを再始動することができます。

## 手順

- **strmqm** コマンドを使用してキュー・マネージャーを再始動するには、コマンドに続いて、再始動するキュー・マネージャーの名前を入力します。

例えば、**strmqm saturn.queue.manager** という名前のキュー・マネージャーを開始するには、次のコマンドを入力します。

```
strmqm saturn.queue.manager
```

- **Windows** **Linux** IBM MQ Explorer を使用してキュー・マネージャーを開始するには、以下のステップを完了します。
  - a) IBM MQ Explorer を開きます。
  - b) ナビゲーター・ビューでキュー・マネージャーを選択します。
  - c) 「開始」 をクリックします。

## タスクの結果

キュー・マネージャーが再始動します。

キュー・マネージャーの再始動に数秒より長い時間がかかると、開始の進行状況の詳細を示す情報メッセージが IBM MQ から断続的に発行されます。

## Multi キュー・マネージャーの削除

**dltmqm** コマンドを使用して、キュー・マネージャーを削除できます。あるいは、Windows および Linux システムでは、IBM MQ Explorer を使用してキュー・マネージャーを削除することもできます。

## 始める前に



### 重要:

- キュー・マネージャーを削除すると、それに関連したキューやそのメッセージなどすべてのリソースの他、すべてのオブジェクト定義も削除されるため、十分な注意が必要です。**dltmqm** コマンドを使用する場合、確認のためのプロンプトは表示されません。Enter キーを押すと、関連するすべてのリソースが失われます。
- **Windows** Windows では、キュー・マネージャーを削除すると、そのキュー・マネージャーは自動始動リスト (12 ページの『キュー・マネージャーの開始』を参照) から削除されます。コマンドが完了すると、IBM MQ queue manager ending メッセージが表示されます。キュー・マネージャーが削除されたことは通知されません。
- クラスター・キュー・マネージャーを削除しても、クラスターからはキュー・マネージャーは除去されません。詳しくは、[dltmqm](#) で使用上の注意を参照してください。

## このタスクについて

**dltmqm** コマンドを使用して、キュー・マネージャーを削除できます。**dltmqm** コマンドとそのオプションの説明については、[dltmqm](#) を参照してください。信頼できる管理者のみにこのコマンドの使用権限を与えるようにします。(セキュリティについては、[UNIX, Linux, and Windows でのセキュリティのセットアップ](#)を参照してください。)

**Windows** **Linux** あるいは、Windows および Linux (x86 および x86-64 プラットフォーム) システムでは、IBM MQ Explorer を使用してキュー・マネージャーを削除することもできます。

## 手順

- **dltmqm** コマンドを使用してキュー・マネージャーを削除するには、以下のステップを完了します。
  - a) キュー・マネージャーを停止させます。
  - b) 以下のコマンドを発行します。

```
dltmqm QMB
```

**注** : **dltmqm** コマンドは、作業対象のキュー・マネージャーに関連付けられたインストール済み環境から使用する必要があります。 **dspmqr -o installation** コマンドを使用して、どのインストール済み環境にキュー・マネージャーが関連付けられているかを調べることができます。

- **Windows** **Linux** IBM MQ Explorer を使用してキュー・マネージャーを削除するには、以下のステップを完了します。
  - a) IBM MQ Explorer を開きます。
  - b) ナビゲーター・ビューでキュー・マネージャーを選択します。
  - c) キュー・マネージャーが停止していない場合は、停止させます。  
キュー・マネージャーを停止するには、それを右クリックしてから、「**停止**」をクリックします。
  - d) キュー・マネージャーを削除してください。  
キュー・マネージャーを削除するには、それを右クリックしてから、「**削除**」をクリックします。

## タスクの結果

キュー・マネージャーが削除されます。

## サーバーとクライアント間の接続の構成

IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャンネルを定義します。

### このタスクについて

IBM MQ では、オブジェクト間の論理通信リンクをチャンネルと呼びます。IBM MQ MQI clients をサーバーに接続するのに使用されるチャンネルを、MQI チャンネルといいます。IBM MQ MQI client 側の IBM MQ アプリケーションがサーバー側のキュー・マネージャーと通信できるように、チャンネル定義をリンクの両端でセットアップします。

MQI チャンネルを定義する前に、使用する通信形式を決定し、チャンネルの両端の接続を定義する必要があります。

## 手順

1. 使用する予定の通信形式を決めます。  
[17 ページの『使用する通信タイプ』](#)を参照してください。
2. チャンネル両端の接続を定義します。  
接続を定義するには、次のステップを実行する必要があります。
  - a) 接続を設定します。
  - b) チャンネル定義のために必要になるパラメーターの値を記録しておきます。
  - c) リスナーを開始して、サーバーによる IBM MQ MQI client からの着信ネットワーク要求の検出を使用可能にします。

## 関連概念

### [19 ページの『拡張トランザクション・クライアントの構成』](#)

このトピックの集合では、トランザクション・マネージャーのカテゴリごとに拡張トランザクション機能を構成する方法を説明します。

### [30 ページの『MQI チャンネルの定義』](#)

新しいチャンネルを作成するには、同じチャンネル名および互換性のあるチャンネル・タイプを使用して、接続の両端について1つずつ、合わせて**2つ**のチャンネル定義を作成する必要があります。この場合、チャンネルタイプは *server-connection* と *client-connection* です。

### [48 ページの『MQI チャンネル用のチャンネル出口プログラム』](#)

UNIX, Linux, and Windows 上の IBM MQ MQI client 環境では、3つのタイプのチャンネル出口を使用できません。

## 関連タスク

### [36 ページの『異なるプラットフォーム上でのサーバー接続およびクライアント接続の定義の作成』](#)

各チャンネル定義を適用先のコンピューター上で作成できます。ただし、クライアント・コンピューター上でチャンネル定義を作成する方法に関して制限があります。

### [39 ページの『サーバー側でのサーバー接続およびクライアント接続の定義の作成』](#)

サーバー側で両方の定義を作成でき、次にクライアント接続定義をクライアントで使用可能にできます。

### [51 ページの『キュー共有グループへのクライアントの接続』](#)

キュー共有グループのメンバーであるサーバー上のクライアントとキュー・マネージャーとの間に、MQI チャンネルを作成することによって、キュー共有グループにクライアントを接続することができます。

### [52 ページの『構成ファイルを使用したクライアントの構成』](#)

テキスト・ファイル内の属性を使用して、クライアントを構成します。この属性は、環境変数、または他のプラットフォーム固有の方法を使用してオーバーライドすることができます。

## 関連情報

[キュー・マネージャーへの IBM MQ MQI クライアント・アプリケーションの接続](#)

[DISPLAY CHLAUTH](#)

[SET CHLAUTH](#)

## 使用する通信タイプ

サポートされる通信プロトコルはプラットフォームによって異なります。どの伝送プロトコルを選択するかは、IBM MQ MQI client とサーバー双方のプラットフォームの組み合わせ方によります。

### MQI チャンネル用の伝送プロトコルのタイプ

ご使用のクライアントとサーバーのプラットフォームに応じて、MQI チャンネルの伝送プロトコルには以下の最大4つのタイプがあります。

- TCP/IP
- LU 6.2
- NetBIOS
- SPX

MQI チャンネルを定義する際には、それぞれのチャンネル定義で伝送プロトコル(トランスポート・タイプ)属性を指定する必要があります。サーバーは1つのプロトコルに制約されることはなく、さまざまなチャンネル定義でさまざまなプロトコルを指定できます。IBM MQ MQI clients の場合、別の伝送プロトコルを使用する代替 MQI チャンネルがあると役に立つことがあります。

どの伝送プロトコルを選択するかは、IBM MQ クライアントとサーバー双方のプラットフォームの組み合わせ方によります。可能な組み合わせが、次の表に示されています。

表 1. 伝送プロトコル - IBM MQ MQI client とサーバーのプラットフォームの組み合わせ		
伝送プロトコル	IBM MQ MQI client	IBM MQ サーバー (IBM MQ Server)
TCP/IP	 IBM i  UNIX  Windows	 IBM i  UNIX  Windows  z/OS
LU 6.2	 UNIX <sup>1</sup>  Windows	 IBM i  UNIX <sup>1</sup>  Windows  z/OS
NetBIOS	 Windows	 Windows
SPX	 Windows	 Windows

注:

1. Linux (POWER プラットフォーム) を除く

#### 関連概念

[202 ページの『Windows での TCP 接続の定義』](#)

送信側でチャンネルを構成してターゲットのアドレスを指定し、受信側でリスナー・プログラムを実行することにより、TCP 接続を定義します。

[209 ページの『UNIX and Linux での TCP 接続の定義』](#)

送信側のチャンネル定義では、宛先のアドレスを指定します。受信側の接続については、リスナーまたは inet デーモンが構成されます。

[229 ページの『IBM i での TCP 接続の定義』](#)

「接続名」フィールドを使用して、チャンネル定義内に TCP 接続を定義できます。

[750 ページの『z/OS での TCP 接続の定義』](#)

TCP 接続を定義するために、構成するいくつかの設定があります。

[203 ページの『Windows での LU 6.2 接続の定義』](#)

2つのマシン間で LU 6.2 の会話が確立できるように SNA を構成する必要があります。

[213 ページの『UNIX and Linux での LU 6.2 接続の定義』](#)

2つのマシン間で LU 6.2 の会話が確立できるように SNA を構成する必要があります。

[230 ページの『IBM i での LU 6.2 接続の定義』](#)

完全修飾された LU 6.2 接続のモード名、TP 名、および接続名を使用して、LU 6.2 通信の詳細を定義します。

[205 ページの『Windows での NetBIOS 接続の定義』](#)

NetBIOS 接続は、Windows 稼働中のクライアントおよびサーバーにのみ適用されます。IBM MQ は、他の IBM MQ 製品との NetBIOS 接続を確立するとき、3つのタイプの NetBIOS リソース (セッション、コマンド、名前) を使用します。これらの各リソースには制限があり、それはデフォルトで、または NetBIOS のインストール中に選択によって設定されます。

#### 関連資料

[19 ページの『TCP/IP 接続の制限』](#)

1つのTCP/IPポートでキューに入れられる未解決の接続要求の数は、プラットフォームによって異なります。制限に達すると、エラーが発生します。

752 ページの『APPC/MVSを使用したz/OS用のLU6.2接続の定義』  
LU6.2接続を定義するために、構成するいくつかの設定があります。

## TCP/IP 接続の制限

1つのTCP/IPポートでキューに入れられる未解決の接続要求の数は、プラットフォームによって異なります。制限に達すると、エラーが発生します。

この接続制限は、IBM MQ サーバーに接続できるクライアントの最大数と同じではありません。1つのサーバーに対して、サーバー・システム・リソースによって決められたレベルまでのクライアントを接続できます。接続要求のバックログ値が、以下の表に示されています。

サーバーのプラットフォーム	接続要求の最大数
 AIX®	100
 HP-UX	20
 Linux	100
 IBM i	255
 Solaris	100
 Windows Server	100
 Windows ワークステーション	100
 z/OS	255

接続制限に達すると、クライアントはMQCONN呼び出しから戻りコードMQRC\_HOST\_NOT\_AVAILABLEを受け取り、クライアント・エラー・ログ (UNIX and Linux システムの場合は /var/mqm/errors/AMQERR0n.LOG、Windows 上の IBM MQ クライアント・インストール済み環境の errors サブディレクトリー内の amqerr0n.log) に AMQ9202 エラーを受け取ります。クライアントがMQCONNを再試行すると、正常に処理される場合があります。

使用できる接続要求の数を増やし、この制限のためにエラー・メッセージが生成されるのを避けるために、それぞれ異なるポートでlistenする複数のリスナーを持つか、複数のキュー・マネージャーを持つことができます。

## 拡張トランザクション・クライアントの構成

このトピックの集合では、トランザクション・マネージャーのカテゴリーごとに拡張トランザクション機能を構成する方法を説明します。

各プラットフォームで、拡張トランザクション・クライアントは、次の外部トランザクション・マネージャーをサポートします。

### XA 対応トランザクション・マネージャー

拡張トランザクション・クライアントは、XA 対応トランザクション・マネージャー (例えば、CICS® および Tuxedo) をサポートするために、XA リソース・マネージャー・インターフェースを提供します。

### Microsoft Transaction Server (Windows システムのみ)

Windows システムのみで、XA リソース・マネージャー・インターフェースは Microsoft Transaction Server (MTS) もサポートします。拡張トランザクション・クライアントに付属の IBM MQ MTS サポートは、MTS と XA リソース・マネージャー・インターフェースとの間のブリッジを提供します。

## WebSphere® Application Server

旧バージョンの IBM WebSphere MQ は、WebSphere Application Server 4 または 5 をサポートしていません。拡張トランザクション・クライアントを使用するには、特定の構成タスクを実行する必要があります。WebSphere Application Server 6 以降には IBM WebSphere MQ または IBM MQ メッセージング・プロバイダーが組み込まれているため、拡張トランザクション・クライアントを使用する必要はありません。

### 関連概念

20 ページの『[XA 対応トランザクション・マネージャーの構成](#)』

まず、IBM MQ ベース・クライアントを構成し、次に以下のトピックにある情報を使用して、拡張トランザクション機能を構成します。

29 ページの『[Microsoft Transaction Server](#)』

MTS をトランザクション・マネージャーとして使用する前に必要な追加の構成はありません。ただし、注意すべきいくつかのポイントがあります。

## XA 対応トランザクション・マネージャーの構成

まず、IBM MQ ベース・クライアントを構成し、次に以下のトピックにある情報を使用して、拡張トランザクション機能を構成します。

**注：**このセクションは、The Open Group によって *Distributed Transaction Processing: The XA Specification* の中で公開されている XA インターフェースについて、基本的な理解があることを前提としています。

拡張トランザクション・クライアントを構成するには、まず、IBM MQ ベース・クライアントを構成する必要があります。詳しくは以下を参照してください。

- ▶ **AIX** [AIX での IBM MQ クライアントのインストール](#)
- ▶ **HP-UX** [HP-UX での IBM MQ クライアントのインストール](#)
- ▶ **Linux** [Linux での IBM MQ クライアントのインストール](#)
- ▶ **Solaris** [Solaris での IBM MQ クライアントのインストール](#)
- ▶ **Windows** [Windows での IBM MQ クライアントのインストール](#)
- ▶ **IBM i** [IBM i での IBM MQ クライアントのインストール](#)

その後、使用するプラットフォームに関する情報を基に、XA 対応トランザクション・マネージャー (CICS や Tuxedo など) 用に拡張トランザクション機能を構成できます。

トランザクション・マネージャーは、キュー・マネージャーに接続されるクライアント・アプリケーションによって使用されるのと同じ MQI チャネルを使用して、リソース・マネージャーとしてのキュー・マネージャーと情報を交換します。トランザクション・マネージャーがリソース・マネージャー (xa\_) 関数呼び出しを発行すると、その呼び出しをキュー・マネージャーに転送し、キュー・マネージャーからの出力を受け取るために MQI チャネルが使用されます。

MQI チャネルは、トランザクション・マネージャーが、リソース・マネージャーとしてキュー・マネージャーを開くために xa\_open 呼び出しを発行する場合、またはクライアント・アプリケーションが、MQCONN または MQCONNX 呼び出しを発行する場合に開始されます。

- トランザクション・マネージャーが MQI チャネルを開始して、クライアント・アプリケーションが同じスレッド上でそれ以降に MQCONN または MQCONNX を呼び出す場合、この MQCONN または MQCONNX 呼び出しは正常に完了し、接続ハンドルがアプリケーションに戻されます。アプリケーションは、MQCC\_WARNING 完了コードを受け取らず、MQRC\_ALREADY\_CONNECTED 理由コードが出されます。
- クライアント・アプリケーションが MQI チャネルを開始して、トランザクション・マネージャーが同じスレッド上でそれ以降に xa\_open を呼び出す場合、この xa\_open 呼び出しは、MQI チャネルを使用してキュー・マネージャーに転送されます。

障害後のリカバリー状態では、クライアント・アプリケーションが動作していないので、トランザクション・マネージャーは、専用の MQI チャネルを使用して、障害の時点でキュー・マネージャーが参加していた未完了の作業単位をリカバリーすることができます。

XA 対応トランザクション・マネージャーと一緒に拡張トランザクション・クライアントを使用する場合は、次の条件に注意してください。

- 単一スレッド内で、クライアント・アプリケーションは一度に1つのキュー・マネージャーにしか接続できない。この制約は拡張トランザクション・クライアントを使用している場合のみ適用されます。IBM MQ ベース・クライアントを使用するクライアント・アプリケーションは、単一スレッド内で同時に複数のキュー・マネージャーに接続できます。
- クライアント・アプリケーションの各スレッドは、別々のキュー・マネージャーに接続できる。
- クライアント・アプリケーションは、共有接続ハンドルを使用できない。

拡張トランザクション機能を構成するには、リソース・マネージャーの役目をするキュー・マネージャーごとに、次の情報をトランザクション・マネージャーに提供する必要があります。

- `xa_open` スtring
- XA スイッチ構造を指すポインター

トランザクション・マネージャーが `xa_open` を呼び出して、キュー・マネージャーをリソース・マネージャーとして開く場合、その呼び出しで、引数 `xa_info` として `xa_open` String を拡張トランザクション・クライアントに渡します。拡張トランザクション・クライアントは、次のように `xa_open` String 内の情報を使用します。

- サーバー・キュー・マネージャーとの MQI チャンネルを開始する (クライアント・アプリケーションがまだ MQI チャンネルを開始していない場合)。
- トランザクション・マネージャーがリソース・マネージャーとして開くキュー・マネージャーが、クライアント・アプリケーションが接続する先のキュー・マネージャーと同じであるかどうかを調べる。
- キュー・マネージャーが動的登録を使用する場合、トランザクション・マネージャーの `ax_reg` 関数と `ax_unreg` 関数を見付ける。

`xa_open` String の形式について、および `xa_open` String 内の情報が拡張トランザクション・クライアントによってどのように使用されるかの詳細については、22 ページの『[xa\\_open String の形式](#)』を参照してください。

XA スイッチ構造により、トランザクション・マネージャーは、拡張トランザクション・クライアントが提供する `xa_` 関数を見付けることができます。また、XA スイッチ構造は、キュー・マネージャーが動的登録を使用するかどうかを指定します。拡張トランザクション・クライアントに付属の XA スイッチ構造については、26 ページの『[XA スイッチ構造](#)』を参照してください。

特定のトランザクション・マネージャー用に拡張トランザクション機能を構成する方法について、および拡張トランザクション・クライアントと一緒にトランザクション・マネージャーを使用する場合のその他の情報については、次のセクションを参照してください。

- [28 ページの『CICS 用の拡張トランザクション・クライアントの構成』](#)
- [29 ページの『Tuxedo 用の拡張トランザクション・クライアントの構成』](#)

### 関連概念

[24 ページの『xa\\_open String の CHANNEL、TRPTYPE、CONNAME、および QMNAME パラメータ』](#)

以下の情報を使用して、拡張トランザクション・クライアントがこれらのパラメータを使用して、接続先のキュー・マネージャーを判別する方法を理解します。

[26 ページの『xa\\_open の追加のエラー処理』](#)

特定の状況で、`xa_open` 呼び出しは失敗します。

### 関連タスク

[27 ページの『TLS チャンネルを持つ拡張トランザクション・クライアントの使用』](#)

TLS チャンネルは `xa_open` String を使用してセットアップすることができません。クライアント・チャンネル定義テーブル (CCDT) を使用するには、以下の指示に従ってください。

### 関連資料

[25 ページの『TPM および AXLIB パラメータ』](#)

拡張トランザクション・クライアントは、TPM および AXLIB パラメーターを使用して、トランザクション・マネージャーの `ax_reg` 関数および `ax_unreg` 関数を見付けます。これらの関数が使用されるのは、キュー・マネージャーが動的登録を使用する場合だけです。

#### 26 ページの『拡張トランザクション処理における障害後のリカバリー』

障害後、トランザクション・マネージャーは未完了の作業単位をリカバリーできなければなりません。これを行うには、トランザクション・マネージャーは、障害の時点で未完了の作業単位に参加していたすべてのキュー・マネージャーを、リソース・マネージャーとして開くことができなければなりません。

### **z/OS IBM MQ for z/OS の拡張トランザクション・クライアント接続での考慮事項**

XA トランザクション・マネージャーのなかには、IBM MQ for z/OS に接続するクライアントが標準的に利用できる機能と適合しないトランザクション調整呼び出しシーケンスを使用するものがあります。

適合しないシーケンスを検出すると、一般には、IBM MQ for z/OS は、その接続に対して異常終了を発行し、クライアントにエラー応答を返します。

例えば、`xa_prepare` が異常終了 5C6-00D4007D を受け取り、戻りコード -3 (XAER\_RMERR) がクライアントに返されることがあります。

別の例として、`xa_end` が異常終了 5C6-00D40079 を受け取る場合があります。

トランザクション・マネージャーでそのような状態が検出された場合は、トランザクション・マネージャーが IBM MQ for z/OS と対話できるように、以下の処置を実行してください。

- APAR [PI73140](#) のフィックスを適用します。
- トランザクション・マネージャーが使用するサーバー接続チャンネルに対して、PI73140 で適用された変更を有効にします。

変更を有効にするには、SVRCONN チャンネルの記述フィールドの任意の場所にキーワード `CSQSERVICE1` (大文字) を指定します。

キーワード `CSQSERVICE1` を指定したチャンネルには、以下の制限があります。

- リカバリー単位 `GROUP` の処理を実行できません。リカバリー単位 `QMGR` の処理のみが可能です。後処理は、`xa_open` 呼び出しで指定された名前によって決定されます。キュー共有グループ名が使用されている場合、XA 接続はグループ・リカバリー単位を要求します。

`xa_open` 呼び出しでは、**`xa_info`** パラメーターにキュー共有グループ名を指定すると、`xaer_inval` で失敗します。

- `MQGMO_LOCK` オプションと `MQGMO_UNLOCK` オプションは使用できません。`MQGET` 呼び出しで `MQGMO_LOCK` または `MQGMO_UNLOCK` を使用すると、`MQRC_ENVIRONMENT_ERROR` で失敗します。

#### 関連概念

##### 20 ページの『XA 対応トランザクション・マネージャーの構成』

まず、IBM MQ ベース・クライアントを構成し、次に以下のトピックにある情報を使用して、拡張トランザクション機能を構成します。

### **`xa_open` スtringの形式**

`xa_open` String には、定義済みパラメーターの名前と値のペアが含まれています。

`xa_open` String の形式は次のとおりです。

```
parm_name1 = parm_value1, parm_name2 = parm_value2, ...
```

ここで、`parm_name` はパラメーターの名前であり、`parm_value` はパラメーターの値です。パラメーターの名前には大文字小文字の区別がありませんが、特に指示のない限り、パラメーターの値には大文字小文字の区別があります。パラメーターは任意の順序で指定できます。

パラメーターの名前、意味、および有効な値は、次のとおりです。

#### 名前

##### 意味と有効な値

## CHANNEL

MQI チャンネルの名前。

これは任意指定のパラメーターです。このパラメーターを指定する場合、CONNAME パラメーターも指定する必要があります。

## TRPTYPE

MQI チャンネルの通信プロトコル。以下のプロトコルは有効な値です。

### LU62

SNA LU 6.2

### NETBIOS

NetBIOS

### SPX

IPX/SPX

### TCP

TCP/IP

これは任意指定のパラメーターです。このパラメーターを省略すると、デフォルト値 TCP が取られます。このパラメーターの値には、大文字小文字の区別がありません。

## CONNAME

MQI チャンネルのサーバー側にあるキュー・マネージャーのネットワーク・アドレス。このパラメーターの有効な値は、TRPTYPE パラメーターの値によって異なります。

### LU62

CPI-C サイド情報項目を識別するシンボリック宛先名。

パートナー LU のネットワーク修飾名は有効な値ではなく、パートナー LU 別名も有効な値ではありません。これは、トランザクション・プログラム (TP) 名とモード名を指定するための追加パラメーターがないからです。

### NETBIOS

NetBIOS の名前。

### SPX

4 バイトのネットワーク・アドレス、6 バイトのノード・アドレス、およびオプションの 2 バイトのソケット番号。これらの値は、16 進数表記で指定する必要があります。ネットワーク・アドレスとノード・アドレスの間をピリオドで区切る必要があります。ソケット番号が指定される場合、小括弧で囲む必要があります。以下に例を示します。

```
0a0b0c0d.804abcde23a1(5e86)
```

ソケット番号を省略すると、デフォルト値 5e86 が取られます。

### TCP

ホスト名または IP アドレス。オプションとして、ポート番号を小括弧に入れて続けることができます。ポート番号を省略すると、デフォルト値 1414 が取られます。セミコロン区切り文字を使用して、キュー・マネージャーの複数のホストとポートを指定できます。以下に例を示します。

```
host1(1415);host2(1416);host3(1417)
```

これは任意指定のパラメーターです。このパラメーターを指定する場合、CHANNEL パラメーターも指定する必要があります。

## QMNAME

MQI チャンネルのサーバー側にあるキュー・マネージャーの名前。この名前は、ブランクまたは単一アスタリスク (\*) にすることはできません。また、この名前の先頭をアスタリスクにすることもできません。つまり、このパラメーターは、特定のキュー・マネージャーを名前で識別する必要があります。

これは必須パラメーターです。

クライアント・アプリケーションが特定のキュー・マネージャーに接続されている場合は、トランザクション・リカバリーを同じキュー・マネージャーで処理する必要があります。

アプリケーションが z/OS キュー・マネージャーに接続されている場合は、そのアプリケーションで特定のキュー・マネージャーの名前、またはキュー共有グループ (QSG) の名前のいずれかを指定できます。キュー・マネージャー名またはキュー共有グループ名を使用して、アプリケーションは、リカバリー処理の単位として QMGR または GROUP を指定して、トランザクションに参加するかどうかを制御します。GROUP リカバリー単位属性指定では、トランザクションのリカバリーを QSG の任意のメンバーで処理することができます。リカバリー単位として GROUP を使用するには、**GROUPUR** キュー・マネージャー属性を有効にする必要があります。

**z/OS** GROUP リカバリー単位の使用については、[キュー共有グループでのリカバリー単位属性指定](#)を参照してください。

## TPM

使用されるトランザクション・マネージャー。有効な値は CICS および TUXEDO です。

拡張トランザクション・クライアントは、このパラメーターと AXLIB パラメーターを同じ目的に使用します。これらのパラメーターの詳細については、[TPM および AXLIB パラメーター](#)を参照してください。

これは任意指定のパラメーターです。このパラメーターの値には、大文字小文字の区別がありません。

## AXLIB

トランザクション・マネージャーの ax\_reg 関数と ax\_unreg 関数を含むライブラリーの名前。

これは任意指定のパラメーターです。

## UID

認証のためにキュー・マネージャーに提供されるユーザー ID。このパラメーターを指定する場合、PWD パラメーターも指定する必要があります。提供されたユーザー ID とパスワードが認証される場合、トランザクション・マネージャーの接続を識別するためにユーザー ID が使用されます。ユーザー ID とパスワードは、MQCONNX 呼び出しの際に MQCSP オブジェクトに取り込まれます。

**UID** パラメーターと **PWD** パラメーターは、クライアント・バイন্ディングとサーバー・バイন্ディングのどちらにも有効です。

## PWD

認証のためにキュー・マネージャーに提供されるパスワード。このパラメーターを指定する場合、**UID** パラメーターも指定する必要があります。

**警告:** クライアント・アプリケーションの MQCSP 構造のパスワードは、ネットワークを經由してプレーン・テキストで送信される場合があります。クライアント・アプリケーションのパスワードが適切に保護されるようにするには、[IBM MQCSP パスワード保護](#)を参照してください。

xa\_open スtringの例は次のとおりです。

```
channel=MARS.SVR,trptype=tcp,conname=MARS(1415),qmname=MARS,tpm=cics
```

## xa\_open スtringの CHANNEL、TRPTYPE、CONNAME、および QMNAME パラメーター

以下の情報を使用して、拡張トランザクション・クライアントがこれらのパラメーターを使用して、接続先のキュー・マネージャーを判別する方法を理解します。

CHANNEL および CONNAME パラメーターが xa\_open スtringに指定される場合、拡張トランザクション・クライアントはこれらのパラメーターと TRPTYPE パラメーターを使用して、サーバー・キュー・マネージャーとの MQI チャネルを開始します。

CHANNEL および CONNAME パラメーターが xa\_open スtringで指定されない場合、拡張トランザクション・クライアントは、MQSERVER 環境変数の値を使用して MQI チャネルを開始します。MQSERVER 環境変数が定義されていない場合、拡張トランザクション・クライアントは、QMNAME パラメーターによって識別されるクライアント・チャネル定義内の項目を使用します。

上記のそれぞれの場合、拡張トランザクション・クライアントは、QMNAME パラメーターの値が、MQI チャネルのサーバー側にあるキュー・マネージャーの名前であるかどうかを調べます。その名前でない場合、xa\_open 呼び出しは失敗し、トランザクション・マネージャーはその失敗をアプリケーションに報告します。

アプリケーションが 7.0.1 より前のバージョンのキュー・マネージャーに接続する場合、xa\_open 呼び出しは成功しますが、トランザクションには QMGR リカバリー単位属性指定があります。  GROUP リカバリー単位属性指定を必要とするアプリケーションが、7.0.1 以降のキュー・マネージャーにのみ接続されていることを確認してください。

 アプリケーションが QMNAME パラメーター・フィールドでキュー共有グループ名を使用している状態で、接続先のキュー・マネージャーで GROUPUR プロパティが無効の場合は、xa\_open 呼び出しは失敗します。

 アプリケーション・クライアントが 7.0.1 以降の z/OS キュー・マネージャーに接続する場合は、QMNAME パラメーターにキュー共有グループ (QSG) 名を指定できます。これにより、アプリケーション・クライアントは、リカバリー単位属性指定として GROUP を使用してトランザクションに参加することができます。GROUP リカバリー単位処理の詳細については、[リカバリー単位の処理](#)を参照してください。

トランザクション・マネージャーが xa\_open 呼び出しを発行するために使用したのと同じスレッド上で、クライアント・アプリケーションがそれ以降に MQCONN または MQCONNX を呼び出す場合、アプリケーションは、xa\_open 呼び出しによって開始された MQI チャネルの接続ハンドルを受け取ります。2 番目の MQI チャネルは開始されません。拡張トランザクション・クライアントは、MQCONN または MQCONNX 呼び出しの QMgrName パラメーターの値が、MQI チャネルのサーバー側にあるキュー・マネージャーの名前であるかどうかを調べます。その名前でない場合、MQCONN または MQCONNX 呼び出しは失敗し、理由コード MQRC\_ANOTHER\_Q\_MGR\_CONNECTED が出されます。QMgrName パラメーターの値が空白であるか、単一アスタリスク (\*) である場合、または先頭の文字がアスタリスクである場合、MQCONN または MQCONNX 呼び出しは失敗し、理由コード MQRC\_Q\_MGR\_NAME\_ERROR が出されます。

トランザクション・マネージャーが同じスレッド上で xa\_open を呼び出す前に、クライアント・アプリケーションが MQCONN または MQCONNX を呼び出して、MQI チャネルをすでに開始している場合、トランザクション・マネージャーは代わりにこの MQI チャネルを使用します。2 番目の MQI チャネルは開始されません。拡張トランザクション・クライアントは、xa\_open スtring 内の QMNAME パラメーターの値が、サーバー・キュー・マネージャーの名前であるかどうかを調べます。その名前でない場合、xa\_open 呼び出しは失敗します。

クライアント・アプリケーションが最初に MQI チャネルを開始する場合、MQCONN または MQCONNX 呼び出しの QMgrName パラメーターの値は、空白または単一アスタリスク (\*) にすることができます。または、先頭をアスタリスクにすることができます。しかし、この状態では、トランザクション・マネージャーが同じスレッド上で後で xa\_open を呼び出すときに、リソース・マネージャーとして開く予定のキュー・マネージャーとアプリケーションの接続先のキュー・マネージャーとが同じであることを確認する必要があります。したがって、QMgrName パラメーターの値が、キュー・マネージャーを名前で明示的に識別する場合の方が、発生する問題が少なくなります。

## TPM および AXLIB パラメーター

拡張トランザクション・クライアントは、TPM および AXLIB パラメーターを使用して、トランザクション・マネージャーの ax\_reg 関数および ax\_unreg 関数を見付けます。これらの関数が使用されるのは、キュー・マネージャーが動的登録を使用する場合だけです。

xa\_open スtring で TPM パラメーターが指定されるときに、AXLIB パラメーターが指定されない場合、拡張トランザクション・クライアントは、TPM パラメーターの値に基づいて、AXLIB パラメーターの値を想定します。想定される AXLIB パラメーターの値については、[25 ページの表 3](#) を参照してください。

TPM の値	プラットフォーム	想定される AXLIB の値
CICS	AIX	/usr/lpp/encina/lib/libEncServer.a(EncServer_shr.o)
CICS	HP-UX	/opt/encina/lib/libEncServer.sl
CICS	Solaris	/opt/encina/lib/libEncServer.so

表 3. 想定される AXLIB パラメーターの値 (続き)

TPM の値	プラットフォーム	想定される AXLIB の値
CICS	Windows システム	libEncServer
Tuxedo	AIX	/usr/lpp/tuxedo/lib/libtux.a(libtux.so.60)
Tuxedo	HP-UX	/opt/tuxedo/lib/libtux.sl
Tuxedo	Solaris	/opt/tuxedo/lib/libtux.so.60
Tuxedo	Windows システム	libtux

xa\_open ストリングで AXLIB パラメーターが指定される場合、拡張トランザクション・クライアントは、その値を使用して、TPM パラメーターの値に基づいて想定された値をオーバーライドします。また、TPM パラメーターに指定された値がないトランザクション・マネージャーにも、AXLIB パラメーターを使用できます。

### xa\_open の追加のエラー処理

特定の状況で、xa\_open 呼び出しは失敗します。

このセクションのトピックでは、xa\_open 呼び出しが失敗する状態について説明しています。次のいずれかの状態が発生する場合も、失敗します。

- xa\_open ストリングにエラーがある。
- MQI チャンネルを開始する十分な情報がない。
- MQI チャンネルを開始しようとするときに問題がある (例えば、サーバー・キュー・マネージャーが動作していない)。

### 拡張トランザクション処理における障害後のリカバリー

障害後、トランザクション・マネージャーは未完了の作業単位をリカバリーできなければなりません。これを行うには、トランザクション・マネージャーは、障害の時点で未完了の作業単位に参加していたすべてのキュー・マネージャーを、リソース・マネージャーとして開くことができなければなりません。

そのため、構成情報を変更する前に、未完了の作業単位がすべて解決されていることを確認する必要があります。

または、トランザクション・マネージャーが必要なキュー・マネージャーを開く機能に、構成の変更が影響を与えないことを確認する必要があります。このような構成変更の例を以下に示します。

- xa\_open ストリングの内容の変更
- MQSERVER 環境変数の値の変更
- クライアント・チャンネル定義テーブル (CCDT) 内の項目の変更
- サーバー接続チャンネル定義の削除

### XA スイッチ構造

各プラットフォームでは、次の 2 つの XA スイッチ構造が拡張トランザクション・クライアントと一緒に提供されています。

これらのスイッチ構造は、次のとおりです。

#### MQRMIXASwitch

このスイッチ構造は、リソース・マネージャーの役目をするキュー・マネージャーが動的登録を使用しない場合に、トランザクション・マネージャーによって使用されます。

#### MQRMIXASwitchDynamic

このスイッチ構造は、リソース・マネージャーの役目をするキュー・マネージャーが動的登録を使用する場合に、トランザクション・マネージャーによって使用されます。

これらのスイッチ構造は、[27 ページの表 4](#) に表示されているライブラリー内にあります。

表 4. XA スイッチ構造が入っている IBM MQ ライブラリー	
プラットフォーム	XA スイッチ構造が入っているライブラリー
AIX HP-UX Linux Solaris	MQ_INSTALLATION_PATH/lib/libmqcxa
Windows システム	MQ_INSTALLATION_PATH\bin\mqcxa.dll <sup>1</sup>

MQ\_INSTALLATION\_PATH は、IBM MQ がインストールされている上位ディレクトリーを表します。

各スイッチ構造内の IBM MQ リソース・マネージャーの名前は MQSeries\_XA\_RMI ですが、複数のキュー・マネージャーが同じスイッチ構造を共有することもできます。

### 関連概念

27 ページの『動的登録および拡張トランザクション処理』

動的登録の使用は、最適化の 1 つの形式です。これは、動的登録が、トランザクション・マネージャーによって発行される xa\_ 関数呼び出しの数を減らすことができるからです。

#### 動的登録および拡張トランザクション処理

動的登録の使用は、最適化の 1 つの形式です。これは、動的登録が、トランザクション・マネージャーによって発行される xa\_ 関数呼び出しの数を減らすことができるからです。

キュー・マネージャーが動的登録を使用しない場合、トランザクション・マネージャーは、すべての作業単位にキュー・マネージャーを関与させます。作業単位内で更新されるリソースがキュー・マネージャーにない場合であっても、トランザクション・マネージャーは xa\_start、xa\_end、および xa\_prepare を呼び出して、キュー・マネージャーを関与させます。

キュー・マネージャーが動的登録を使用する場合、トランザクション・マネージャーは、キュー・マネージャーが作業単位に関係ないものと想定して開始し、xa\_start を呼び出しません。キュー・マネージャーが作業単位に関与するのは、そのリソースが同期点制御内で更新される場合だけです。キュー・マネージャーのリソースが同期点制御内で更新される場合、拡張トランザクション・クライアントは ax\_reg を呼び出して、キュー・マネージャーの関与を登録します。

### TLS チャンネルを持つ拡張トランザクション・クライアントの使用

TLS チャンネルは xa\_open ストリングを使用してセットアップすることができません。クライアント・チャンネル定義テーブル (CCDT) を使用するには、以下の指示に従ってください。

### このタスクについて

xa\_open xa\_info ストリングのサイズには限界があるため、キュー・マネージャーに接続する xa\_open ストリング・メソッドを使って TLS チャンネルのセットアップに必要なすべての情報を渡すことができません。そのため、クライアント・チャンネル定義テーブルを使用するか、またはトランザクション・マネージャーが許可する場合は xa\_open 呼び出しを発行する前に MQCONN を使ってチャンネルを作成する必要があります。

クライアント・チャンネル定義テーブルを使用するには、以下の手順に従ってください。

### 手順

1. 必須の qmname (キュー・マネージャーの名前) パラメーターだけを含む xa\_open ストリングを指定します (例えば XA\_Open\_String=qmname=MYQM)。
2. キュー・マネージャーを使って、必須の TLS パラメーターと共に CLNTCONN (クライアント接続) チャンネルを定義します。CLNTCONN 定義の QMNAME 属性にキュー・マネージャーの名前を組み込みます。これは xa\_open ストリングの qmname とマッチングされます。
3. クライアント・システムのクライアント・チャンネル定義テーブル (CCDT) または Windows の場合はアクティブ・ディレクトリーで CLNTCONN 定義を利用できるようにします。

4. CCDT を使用している場合は、環境変数 MQCHLLIB および MQCHLTAB を使用している CLNTCONN チャネルの定義を含む CCDT を識別します。クライアント・アプリケーションとトランザクション・マネージャーの両方で使用される環境でこれらの変数を設定します。

## タスクの結果

これにより、トランザクション・マネージャーに、適切なキュー・マネージャーに対するチャネル定義と CipherSpec の正確な認証に必要な TLS 属性 (SSLCIPH を含む) が与えられます。

## CICS 用の拡張トランザクション・クライアントの構成

CICS によって使用されるように拡張トランザクション・クライアントを構成するには、CICS 領域に XAD リソース定義を追加します。

XAD リソース定義を追加するには、CICS オンライン・リソース定義 (RDO) コマンド **cicsadd** を使用します。XAD リソース定義は、次の情報を指定します。

- xa\_open ストリング
- スイッチ・ロード・ファイルの完全修飾パス名

AIX、HP-UX、Solaris、および Windows システムの各プラットフォームでは、CICS が使用するためのスイッチ・ロード・ファイルが1つずつ用意されています。各スイッチ・ロード・ファイルには、動的登録に使用される XA スイッチ構造 MQRMIXASwitchDynamic を指すポインタ関数が含まれています。各スイッチ・ロード・ファイルの完全修飾パス名については、[28 ページの表 5](#) を参照してください。

表 5. スイッチ・ロード・ファイル	
プラットフォーム	スイッチ・ロード・ファイル
AIX HP-UX Linux Solaris	MQ_INSTALLATION_PATH/lib/amqczsc
Windows システム	MQ_INSTALLATION_PATH\bin\mqcc4swi.dll <sup>1</sup>

MQ\_INSTALLATION\_PATH は、IBM MQ がインストールされている上位ディレクトリーを表します。

Windows システム用の XAD リソース定義の例は、次のとおりです。

```
cicsadd -c xad -r REGION1 WMQXA \
ResourceDescription="IBM MQ queue manager MARS" \
XAOpen="channel=MARS.SVR,trptype=tcp,connname=MARS(1415),qmname=MARS,tpm=cics" \
SwitchLoadFile="C:\Program Files\IBM\MQ\bin\mqcc4swi.dll"
```

CICS 領域に XAD リソース定義を追加する場合の詳細については、ご使用のプラットフォーム用の「*CICS Administration Reference*」および「*CICS Administration Guide*」を参照してください。

拡張トランザクション・クライアントと一緒に CICS を使用する場合、次の情報に注意してください。

- CICS 領域には、IBM MQ 用の XAD リソース定義を1つしか追加できません。つまり、1つの領域には1つのキュー・マネージャーしか関連付けることができません。また、領域内で実行されるすべての CICS アプリケーションは、そのキュー・マネージャーにしか接続できません。別のキュー・マネージャーに接続する CICS アプリケーションを実行したい場合は、別の領域でそのアプリケーションを実行する必要があります。
- 領域内の各アプリケーション・サーバーは、初期化時に xa\_open を呼び出し、その領域に関連したキュー・マネージャーとの MQI チャネルを開始します。つまり、アプリケーション・サーバーが始動する前に、キュー・マネージャーを始動しておく必要があります。そうしないと、xa\_open 呼び出しが失敗します。それ以降にアプリケーション・サーバーによって処理されるすべての IBM MQ MQI client・アプリケーションは、同じ MQI チャネルを使用します。

- MQI チャンネルが開始するときに、チャンネルのクライアント側にセキュリティー出口がない場合、クライアント・システムからサーバー接続 MCA に流れるユーザー ID は `cics` です。特定の状況では、サーバー接続 MCA がクライアント・アプリケーションに代わって後でキュー・マネージャー・リソースにアクセスしようとするときに、キュー・マネージャーはこのユーザー ID を権限検査に使用します。このユーザー ID が権限検査に使用される場合、必要なすべてのリソースにアクセスする権限がこのユーザー ID にあることを確認する必要があります。

キュー・マネージャーがこのユーザー ID を権限検査に使用する場合については、[保護](#)を参照してください。

- IBM MQ クライアント・システムで使用するために提供されている CICS タスク終了出口は、[29 ページの表 6](#) にリストされています。これらの出口の構成方法は、IBM MQ サーバー・システム用の対応する出口の構成方法と同じです。したがって、この情報については、[CICS ユーザー出口の有効化](#)を参照してください。

プラットフォーム	ソース	ライブラリー
AIX HP-UX Linux Solaris	amqzscgx.c	amqzscsg
Windows システム	amqzscgn.c	mqcc1415.dll

### Tuxedo 用の拡張トランザクション・クライアントの構成

Tuxedo で使用される XAD リソース定義を構成するには、UBBCONFIG ファイルおよびリソース・マネージャー・テーブルを更新します。

Tuxedo で使用される XAD リソース定義を構成するには、以下のアクションを実行します。

- アプリケーション用の Tuxedo UBBCONFIG ファイルの GROUPS セクションで、OPENINFO パラメーターを使用して `xa_open` スtring を指定します。

この方法の例は、UBBCONFIG サンプル・ファイルを参照してください。このファイルは、Tuxedo サンプル・プログラムで使用するために提供されています。AIX、HP-UX、および Solaris では、このファイルの名前は `ubbstxcx.cfg` です。Windows システムではファイルの名前は `ubbstxcn.cfg` です。

- Tuxedo リソース・マネージャー・テーブル内のキュー・マネージャーの次の項目で、

- `udataobj/RM` (AIX、HP-UX、および Solaris)
- `udataobj&#xa5;rm` (Windows システム)

XA スイッチ構造の名前、およびその構造が入っているライブラリーの完全修飾パス名を指定します。プラットフォームごとにこれを行う方法の例については、[TUXEDO サンプル](#)を参照してください。Tuxedo は、リソース・マネージャーの動的登録をサポートするので、MQRMIXASwitch か MQRMIXASwitchDynamic のどちらでも使用できます。

### Microsoft Transaction Server

MTS をトランザクション・マネージャーとして使用する前に必要な追加の構成はありません。ただし、注意すべきいくつかのポイントがあります。

拡張トランザクション・クライアントと一緒に MTS を使用する場合、次の情報に注意してください。

- MTS アプリケーションは、サーバー・キュー・マネージャーに接続すると、常に MQI チャンネルを開始します。その後、トランザクション・マネージャーの役割をする MTS は、同じ MQI チャンネルを使用してキュー・マネージャーと情報を交換します。
- 障害後、MTS は未完了の作業単位をリカバリーできなければなりません。これを行うには、MTS は、障害の時点で未完了の作業単位に参加していたすべてのキュー・マネージャーと情報を交換できなければなりません。

MTS アプリケーションがサーバー・キュー・マネージャーに接続し、MQI チャンネルを開始すると、拡張トランザクション・クライアントは、MQCONN または MQCONNX 呼び出しのパラメーターから十分な情報を抽出して、必要に応じて、障害後にチャンネルを再開できるようにします。拡張トランザクション・クライアントは、この情報を MTS に渡し、MTS はこの情報をログに記録します。

MTS アプリケーションが MQCONN 呼び出しを発行する場合、この情報は、単にキュー・マネージャーの名前です。MTS アプリケーションが MQCONNX 呼び出しを発行し、チャンネル定義構造 MQCD を提供する場合、その情報には、MQI チャンネルの名前、サーバー・キュー・マネージャーのネットワーク・アドレス、およびチャンネルの通信プロトコルも含まれます。

リカバリー状態では、MTS はこの情報を拡張トランザクション・クライアントに戻し、拡張トランザクション・クライアントはこの情報を使用して MQI チャンネルを再開します。

したがって、構成情報を変更する必要がある場合は、変更を加える前に、未完了の作業単位がすべて解決されていることを確認してください。または、拡張トランザクション・クライアントが MTS によって記録された情報を使用して MQI チャンネルを再開する機能に、構成の変更が影響を与えないことを確認してください。このような構成変更の例を以下に示します。

- MQSERVER 環境変数の値の変更
- クライアント・チャンネル定義テーブル (CCDT) 内の項目の変更
- サーバー接続チャンネル定義の削除
- MTS と一緒に拡張トランザクション・クライアントを使用する場合は、次の条件に注意してください。
  - 単一スレッド内で、クライアント・アプリケーションは一度に 1 つのキュー・マネージャーにしか接続できない。
  - クライアント・アプリケーションの各スレッドは、別々のキュー・マネージャーに接続できる。
  - クライアント・アプリケーションは、共有接続ハンドルを使用できない。

## MQI チャンネルの定義

新しいチャンネルを作成するには、同じチャンネル名および互換性のあるチャンネル・タイプを使用して、接続の両端について 1 つずつ、合わせて **2 つ** のチャンネル定義を作成する必要があります。この場合、チャンネルタイプは *server-connection* と *client-connection* です。

### ユーザー定義チャンネル

サーバーがチャンネルを自動定義しない場合に、チャンネル定義を作成し、IBM MQ MQI client ・マシン上の IBM MQ アプリケーションがチャンネルにアクセスできるようにする方法は、次に示すように 2 とおりあります。

この 2 つの方法については、以下で詳しく説明します。

#### 1. 一方のチャンネル定義を IBM MQ クライアント側、もう一方をサーバー側で作成する方法

これは、IBM MQ MQI client ・プラットフォームおよびサーバー・プラットフォームの任意の組み合わせに適用されます。この方法は、システムを始動するとき、またはセットアップをテストするときに利用してください。

この方式を使用する方法についての詳細は、[36 ページの『異なるプラットフォーム上でのサーバー接続およびクライアント接続の定義の作成』](#)を参照してください。

#### 2. 両方のチャンネル定義をサーバー・マシン側で作成する方法

複数のチャンネルおよび IBM MQ MQI client ・マシンを同時にセットアップする場合は、この方法を使用してください。

この方式を使用する方法についての詳細は、[39 ページの『サーバー側でのサーバー接続およびクライアント接続の定義の作成』](#)を参照してください。

### チャンネルの自動定義

z/OS 以外のプラットフォームにある IBM MQ 製品には、サーバー側にチャンネル定義が存在しない場合に、チャンネル定義を自動的に作成する機能があります。

クライアントからインバウンド接続要求を受信したときに、該当のチャンネル接続が見つからないと、IBM MQ は自動的に定義を作成して、キュー・マネージャーに追加します。自動定義は、デフォルトのサーバー接続チャンネル SYSTEM.AUTO.SVRCONN の定義に基づいて行われます。サーバー接続定義の自動定義を使用可能にするには、ALTER QMGR コマンドに CHAD パラメーターを付けて (または、PCF コマンド Change Queue Manager に ChannelAutoDef パラメーターを付けて) 使用し、キュー・マネージャー・オブジェクトを更新します。

## 関連概念

169 ページの『[チャンネル制御機能](#)』

チャンネル制御機能は、チャンネルを定義、モニター、および制御するための機能を備えています。

## ULW AMQP チャンネルの作成および使用

MQ Light API の IBM MQ サポートを IBM MQ インストール済み環境にインストールすると、IBM MQ MQSC コマンド (`runmqsc`) を実行してチャンネルを定義、変更、削除、開始、および停止することができます。チャンネルの状況を表示することもできます。

## 始める前に

このタスクは、AMQP チャンネルがインストールされていることを前提としています。これは、IBM MQ のインストール時に AMQP サービス・コンポーネントを選択することで実行します。詳しくは、プラットフォームのリンクに従い、「AMQP サービス」のテーブル行を検出します。

- ▶ **AIX** [AIX システム用の IBM MQ コンポーネント](#)
- ▶ **HP-UX** [HP-UX システム用の IBM MQ コンポーネント](#)
- ▶ **Linux** [Linux システム用の IBM MQ rpm コンポーネント](#)
- ▶ **Linux** [Linux Ubuntu システム用の IBM MQ Debian コンポーネント](#)
- ▶ **Solaris** [Solaris システム用の IBM MQ コンポーネント](#)
- ▶ **Windows** [Windows システム用の IBM MQ 機能](#)

キュー・マネージャーへのテスト接続を行うには、MQ Light クライアントが必要です。MQ Light クライアントは、Node.js、Ruby、Java、および Python で使用できます。使用可能なクライアントについて詳しくは、[IBMMQ Light コミュニティ Web サイト](#)をご覧ください。

以下の操作は MQ Light Node.js クライアントを想定しています。しかし、IBM MQ キュー・マネージャーに関連する手順はどのクライアントでも同じです。

## このタスクについて

以下の手順は、既にキュー・マネージャーがあることを想定しています。

新規のキュー・マネージャーが必要な場合のためのサンプル・スクリプトが、`mqinstall/amqp/samples` ディレクトリーにあります。このスクリプトは、新規キュー・マネージャーを作成し、AMQP サービスを開始し、SAMPLE.AMQP.CHANNEL という新規チャンネルを作成し、チャンネルを開始します。

注：AMQP チャンネルはユーザー定義の AMQP サービスをサポートしていません。AMQP チャンネルはシステム・デフォルトの SYSTEM.AMQP.SERVICE サービスのみをサポートしています。

**Windows** ▶ **Linux** サンプル・スクリプト `SampleMQM.sh` (Linux の場合) または `SampleMQM.bat` (Windows の場合) を実行する場合は、[33 ページの『6』](#) で以下の手順を開始できます。

デフォルト・チャンネル SYSTEM.DEF.AMQP を使用して、キュー・マネージャーへの MQ Light 接続をテストすることも、新規チャンネルを作成することもできます。

以下の手順では、デフォルトのチャンネルを使用します。

## 手順

1. `mqinstall/bin/` ディレクトリーから **runmqsc** を開始します。

```
runmqsc QMNAME
```

2. **V 9.0.5**

(キュー・マネージャーが IBM MQ 9.0.4 以前の場合のみ必要です。) AMQP 機能がインストールされ、正常に動作していることを確認します。

**START SERVICE** コマンドを使用して、JVM を制御する IBM MQ サービスを開始します。

```
START SERVICE(SYSTEM.AMQP.SERVICE)
```

**注:** IBM MQ 9.0.5 以降、SYSTEM.AMQP.SERVICE の **CONTROL** 属性が **QMGR** に設定されます。そのため、キュー・マネージャーの開始時にサービスが自動的に開始されます。**CONTROL** 属性を **MANUAL** に設定すると、キュー・マネージャーの開始時にサービスが自動的に開始されないようにすることができます。

キュー・マネージャーの開始時に、AMQP サービスおよび AMQP チャネルは、定義されていれば自動的に開始されます。

3. MCAUSER ユーザー ID を設定します。

AMQP クライアントがチャネルに接続する際に、チャネルは MCAUSER ユーザー ID を指定します。キュー・マネージャーへの接続では、このユーザー ID が使用されます。MCAUSER のデフォルト値はブランクです。MCAUSER 値を指定してからでなければ、どの AMQP クライアントもキュー・マネージャーに接続することはできません。この値は、IBM MQ トピックへのパブリッシュとサブスクライブを許可されている有効な IBM MQ ユーザーでなければなりません。

**注:** **Windows** Windows では、MCAUSER に設定できるユーザー ID の最大長は 12 文字です。

- a) MCAUSER ユーザー ID を設定するには、**ALTER CHANNEL** コマンドを使用します。

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) MCAUSER(User ID)
```

- b) 以下の 2 つの **setmqaut** コマンドを使用して、MCAUSER ユーザー ID にトピックのパブリッシュとサブスクライブを許可します。

```
setmqaut -m QMNAME -t topic -n SYSTEM.BASE.TOPIC -p MCAUSER  
-all +pub +sub
```

および

```
setmqaut -m QMNAME -t qmgr -p MCAUSER -all +connect
```

MCAUSER ユーザー ID を追加または変更する際にチャネルが実行中の場合、チャネルを停止して再始動する必要があります。

**注:** MCAUSER ユーザー ID が設定されていない、または MCAUSER ユーザー ID が IBM MQ トピックへのパブリッシュまたはサブスクライブを許可されていない場合、AMQP クライアントでエラー・メッセージを受け取ります。

4. **START CHANNEL** コマンドを使用して、デフォルトの SYSTEM.DEF.AMQP チャネルを開始します。

```
START CHANNEL(SYSTEM.DEF.AMQP)
```

5. チャネル状況を検査する場合は、**DISPLAY CHSTATUS** コマンドを使用します。

```
DISPLAY CHSTATUS(SYSTEM.DEF.AMQP) CHLTYPE(AMQP)
```

チャンネルが正常に実行中であれば、コマンド出力に STATUS (RUNNING) が表示されます。

6. デフォルトのポートを変更します。

AMQP 1.0 接続のデフォルト・ポートは 5672 です。MQ Light がインストールされていたなどの理由で、すでにポート 5672 が使用されている場合は、AMQP チャンネルが使用するポートを変更する必要があります。ポートを変更するには、次の **ALTER CHANNEL** コマンドを使用します。

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) PORT(NEW PORT NUMBER)
```

7. チャンネル認証 (CHLAUTH) 規則を使用して AMQP チャンネルへの接続をブロックまたはフィルタリングする必要がない場合は、次のコマンドを実行してキュー・マネージャーでのチャンネル認証を無効にします。

```
alter qmgr chlauth(disabled)
```

実動キュー・マネージャーでは接続認証を無効にしないことをお勧めします。接続認証を無効にしてよいのは、開発環境に限られます。

または、キュー・マネージャーのチャンネル認証規則を構成して、AMQP チャンネルへの特定の接続を許可することができます。

8. オプション: キュー・マネージャーの構成済み鍵リポジトリを使用してチャンネルの SSL/TLS 暗号化を有効にするには、チャンネルの SSLCIPH 属性に適切な暗号仕様を設定する必要があります。デフォルトでは、暗号仕様は空白で、これはチャンネルで SSL/TLS 暗号化が使用されないことを意味します。暗号仕様を設定するには、**ALTER CHANNEL** コマンドを使用します。以下に例を示します。

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCIPH(CIPHER SPECIFICATION)
```

加えて、SSL/TLS 暗号化に関連するチャンネル構成オプションは他にいくつもあり、以下のようして設定することが可能です。

- デフォルトでは、キュー・マネージャーの鍵リポジトリにある、キュー・マネージャー CERTLABL 属性に対応するラベルが付けられた証明書は、チャンネルの SSL/TLS 暗号化で使用される名前です。CERTLABL を設定することにより、他の証明書を選択することもできます。**ALTER CHANNEL** コマンドを使用して、必要な証明書のラベルを指定します。

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) CERTLABL(CERTIFICATE LABEL)
```

- チャンネルが SSL/TLS クライアント接続で証明書を必要とするように設定することができます。SSL/TLS クライアント接続で証明書を必要とするかどうかの選択は、SSLCAUTH を設定することにより行えます。**ALTER CHANNEL** コマンドを使用して、SSL/TLS クライアント接続で証明書を必要とするかどうかを設定します。以下に例を示します。

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCAUTH(REQUIRED or OPTIONAL)
```

- V 9.0.0.10** SSLCAUTH 属性を REQUIRED に設定すると、クライアントの証明書の識別名 (DN) が確認されます。クライアントの証明書の識別名 (DN) を確認するには、SSLPEER 属性を設定します。**ALTER CHANNEL** コマンドを使用して、クライアントの証明書の識別名を確認します。以下に例を示します。

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLPEER (DN SPECIFICATION)
```

あるいは、チャンネル認証レコードを使用して接続を許可またはブロックすることもできます。この方法は、SSLPEER 属性を使用する場合と比較して、大幅に詳細な設定を行うことができます。

SSLPEER の設定およびその代替方法としてのチャンネル認証レコードの使用については、[SSL ピア](#)を参照してください。

9. 次のコマンドを実行して、MQ Light Node.js クライアントをインストールします。

```
npm install mqlight
```

10. `node_modules/mqlight/samples` ディレクトリーに移動して、サンプルの受信側アプリケーションを実行します。

- デフォルトのポート番号を使用している場合は、このサンプルの受信側アプリケーションを実行できます。

```
node recv.js
```

- 別のポート番号を使用するように AMQP チャンネルを構成してある場合は、次のようにその新しいポート番号をパラメーターに指定して、サンプルの受信側アプリケーションを実行できます。

```
node recv.js -s amqp://localhost:6789
```

デフォルトのチャンネルへの接続が成功すると、次のメッセージが表示されます。

```
Connected to amqp://localhost:5672 using client-id recv_e79c55d
Subscribed to pattern: public
```

これで、アプリケーションはキュー・マネージャーに接続し、メッセージの受信を待機している状態になります。トピック `public` にサブスクライブしています。

**注:** `client-id` は、`-i` パラメーターで特に指定していなければ、自動的に生成されます。

11. 新しいコマンド・ウィンドウで、`node_modules/mqlight/samples` ディレクトリーに移動し、次のコマンドを実行してサンプルの送信側アプリケーションを実行します。

```
node send.js
```

受信側アプリケーションのコマンド・ウィンドウに、Hello World メッセージが表示されます。

12. **AMQSSUB** IBM MQ サンプルを使用して、MQ Light サンプル・メッセージを受信します。

Linux および Windows では、サンプルは以下の場所にあります:

- **Linux** Linux の場合 `mqinstall/samp/bin` ディレクトリー。
- **Windows** Windows の場合 `mqinstall/Tools\c\Samples\Bin` ディレクトリー。

- a) 次のコマンドを実行して、サンプルを実行します。

```
amqssub public QM-name.
```

- b) 次のコマンドを再度実行して、IBM MQ アプリケーションにメッセージを送信します。

```
node send.js
```

13. 次の **DEFINE CHANNEL** コマンドを使用して、さらに AMQP チャンネルを作成します。

```
DEFINE CHANNEL(MY.AMQP.CHANNEL) CHLTYPE(AMQP) PORT(2345)
```

チャンネルを定義したときは、**START CHANNEL** コマンドを使用して、そのチャンネルを手動で開始する必要があります。

```
START CHANNEL(MY.AMQP.CHANNEL)
```

チャンネルが正常に動作しているかを確認するために、その新しいチャンネルのポートを指定して、サンプルの受信側アプリケーションを実行することができます。

```
node recv.js -s amqp://localhost:2345
```

## 次のタスク

以下のコマンドを使用して、IBM MQ 接続の表示、チャンネルの停止、およびチャンネルの削除を行うことができます。

### DISPLAY CONN(\*) TYPE(CONN) WHERE (CHANNEL EQ SYSTEM.DEF.AMQP)

AMQP チャンネルによってキュー・マネージャーに作成された IBM MQ 接続を表示します。

### DISPLAY CHSTATUS(\*) CHLTYPE(AMQP) CLIENTID(\*) ALL

指定されたチャンネルに接続されている AMQP クライアントのリストを表示します。

### STOP CHANNEL (MY.AMQP.CHANNEL)

AMQP チャンネルを停止し、リスニング中のポートを閉じます。

### DELETE CHANNEL (MY.AMQP.CHANNEL)

作成したすべてのチャンネルを削除します。

注：デフォルトのチャンネル SYSTEM.DEF.AMQP は削除しないでください。

AMQP 機能が IBM MQ インストール済み環境にインストールされているかどうか、およびその機能に関連付けられているキュー・マネージャーがあるかどうかを、**runmqsc** または PCF のいずれかを使用して判別できます。

- **runmqsc** を使用する場合は、キュー・マネージャーの属性を表示して、AMQPCAP(YES) が指定されているかどうかを確認します。
- PCF を使用する場合は、**MQCMD\_INQUIRE\_Q\_MGR** コマンドを実行して、MQIA\_AMQP\_CAPABILITY の値を確認します。

## 関連情報

strmqm

[AMQP クライアント・アプリケーションの開発](#)

[AMQP クライアントの保護](#)

## ULW キュー・マネージャーからの AMQP チャンネルの削除

キュー・マネージャーから AMQP チャンネルを削除するには、インストール・ディレクトリーからフォルダーを削除します。

## 手順

1. キュー・マネージャーを停止させます。
2. MQ Light API の IBM MQ サポートを削除します。

- **AIX** AIX では、以下のコマンドを実行します。

```
installp -u mqm.amqp.rte
```

- **Linux** Linux の場合、AMQP RPM を削除します。この RPM をインストールする前に再パッケージしている場合は、再パッケージした RPM の名前を指定してください。

```
rpm -e MQSeriesAMQP
```

- **Windows** Windows の場合、IBM MQ インストール済み環境から **amqp** フォルダーを削除します。それ以外のファイルやフォルダーを IBM MQ インストール・パスから削除することがないように注意してください。

3. キュー・マネージャーを再始動する。

## 関連情報

[AMQP クライアント・アプリケーションの開発](#)

[AMQP クライアントの保護](#)

## ULW AMQP チャネルのログ・ファイル

AMQP チャネルのログ・ファイルは、IBM MQ ログ・ファイルと同じ IBM MQ データ・ディレクトリーに保管されます。

Windows 上のデフォルトのデータ・ディレクトリーは `C:\ProgramData\IBM\MQ` です。

Linux 上のデフォルトのデータ・ディレクトリーは `/var/mqm` です。

AMQP チャネルは、ログ情報を IBM MQ データ・ディレクトリーの以下のファイルに書き込みます。

- `amqp.stdout`。 `qmgrs/QM-name` フォルダーに書き込まれます。
- `amqp.stderr`。 `qmgrs/QM-name` フォルダーに書き込まれます。
- `amqp_*.log`。 `qmgrs/QM-name/errors` フォルダーに書き込まれます。

MQ Light クライアントが認証エラーまたは許可エラーを受け取ると、管理者は、`amqp_0.log` ファイルおよび `MQ AMQERR*.log` ファイルでセキュリティ障害の原因についての詳細情報を見つけることができます。

FDC ファイルは `AMQP*.FDC` という名前で作成され、`data-directory/errors` フォルダーに書き込まれます。

一部の構成ファイルは `qmgrs/QM-name/amqp` ディレクトリーに書き込まれます。このディレクトリー内のファイルは、どれも修正する必要はありません。

## 関連情報

[UNIX, Linux, and Windows 用エラー・ログ](#)

[AMQP クライアント・アプリケーションの開発](#)

[AMQP クライアントの保護](#)

## 異なるプラットフォーム上でのサーバー接続およびクライアント接続の定義の作成

各チャネル定義を適用先のコンピューター上で作成できます。ただし、クライアント・コンピューター上でチャネル定義を作成する方法に関して制限があります。

### このタスクについて

すべてのプラットフォームで、IBM MQ Script (MQSC) コマンド、プログラマブル・コマンド・フォーマット (PCF) コマンド、または IBM MQ Explorer を使用して、サーバー・マシンでサーバー接続チャネルを定義できます。

**z/OS** z/OS では、操作パネルおよび制御パネルも使用できます。

**IBM i** IBM i では、パネル・インターフェースも使用できます。

MQSC コマンドは、IBM MQ が IBM MQ MQI client としてのみインストールされているマシンでは使用できないので、別の方法を使用して、クライアント・マシンでクライアント接続チャネルを定義する必要があります。

`runmqsc` の場合、以下の考慮事項が適用されます。

- `-c` パラメーターを指定し、オプションで `-u` パラメーターを指定して、管理するキュー・マネージャーへのクライアントとして `runmqsc` を接続することができます。
- `-u` パラメーターを使用してユーザー ID を指定する場合、対応するパスワードを求めるプロンプトが出されます。

- CHCKLOCL(REQUIRED) または CHCKLOCL(REQDADM) を指定して CONNAUTH AUTHINFO レコードを構成した場合、**-u** パラメーターを使用する必要があります。このパラメーターを使用しないと **runmqsc** でキュー・マネージャーを管理することはできません。

## 手順

- サーバーでサーバー接続チャンネルを定義するには、[37 ページの『サーバー側でのサーバー接続チャンネルの定義』](#)を参照してください。
- IBM MQ MQI client でクライアント接続チャンネルを作成するには、[37 ページの『IBM MQ MQI client でのクライアント接続チャンネルの作成』](#)を参照してください。

## サーバー側でのサーバー接続チャンネルの定義

必要に応じて、MQSC を開始し、次にサーバー接続チャンネルを定義します。

## 手順

1. オプション: サーバー・プラットフォームが z/OS 以外の場合は、まずキュー・マネージャーを作成し始動してから、MQSC コマンドを開始します。
  - a) 例えば QM1 と呼ばれるキュー・マネージャーを作成します。

```
critmqm QM1
```

- b) キュー・マネージャーを始動します。

```
stirmqm QM1
```

- c) MQSC コマンドを始動します。

```
runmqsc QM1
```

2. 選択した名前をもつチャンネル、およびサーバー接続のチャンネル・タイプを定義します。

```
DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_1')
```

このチャンネル定義は、サーバー上で稼働するキュー・マネージャーに関連しています。

3. 以下のコマンドを使用して、キュー・マネージャーへのインバウンド接続アクセスを許可します。

```
SET CHLAUTH(CHAN1) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- ここで、SET CHLAUTH は前のステップで定義されたチャンネルの名前を使用します。
- 'IP address' とは、クライアントの IP アドレスのことです。
- 'userid' とは、ターゲット・キューへのアクセス制御のためにチャンネルに提供する ID のことです。このフィールドは、大文字と小文字が区別されます。

いくつかの異なる属性を使用して、インバウンド接続の識別を選択することができます。例では、IP アドレスを使用しています。代替属性には、クライアントのユーザー ID および TLS のサブジェクト識別名が含まれています。詳しくは、[チャンネル認証レコード](#)を参照してください。

## IBM MQ MQI client でのクライアント接続チャンネルの作成

MQSERVER を使用するか、または MQCONNX 呼び出しで MQCNO 構造体を使用して、クライアント・ワークステーション上でクライアント接続チャンネルを定義できます。

## MQSERVER の使用法

MQSERVER 環境変数を使用すると、クライアント接続チャンネルの簡単な定義を指定できます。つまり、このメソッドを使用すると、チャンネルの属性をいくつか指定するだけで済みます。

- Windows では、以下のようにして簡単なチャンネル定義を指定します。

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

- UNIX and Linux システムでは、以下のようにして簡単なチャンネル定義を指定します。

```
export MQSERVER=ChannelName/TransportType/ConnectionName
```

- IBM i システムでは、以下のようにして簡単なチャンネル定義を指定します。

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('ChannelName/TransportType/ConnectionName')
```

ここで、

- ChannelName は、サーバー側で定義した名前と同じ名前にする必要があります。これには、スラッシュ (/) を含めることはできません。
- TransportType は、ご使用の IBM MQ MQI client ・プラットフォームに応じて、次の値のいずれかになります。
  - LU62
  - TCP
  - NETBIOS
  - SPX

**注:** UNIX and Linux システムでは、TransportType には大文字小文字の区別があり、大文字にする必要があります。MQCONN 呼び出しまたは MQCONNX 呼び出しは、TransportType が認識されないと 2058 を戻します。

- ConnectionName は、通信プロトコル (TransportType) に対して定義したサーバーの名前です。

例えば、Windows では、以下を使用します。

```
SET MQSERVER=CHANNEL1/TCP/MCID66499
```

UNIX and Linux システムの場合は、次のとおりです。

```
export MQSERVER=CHANNEL1/TCP/'MCID66499'
```

**注:** TCP/IP ポート番号を変更するには、80 ページの『MQSERVER』を参照してください。

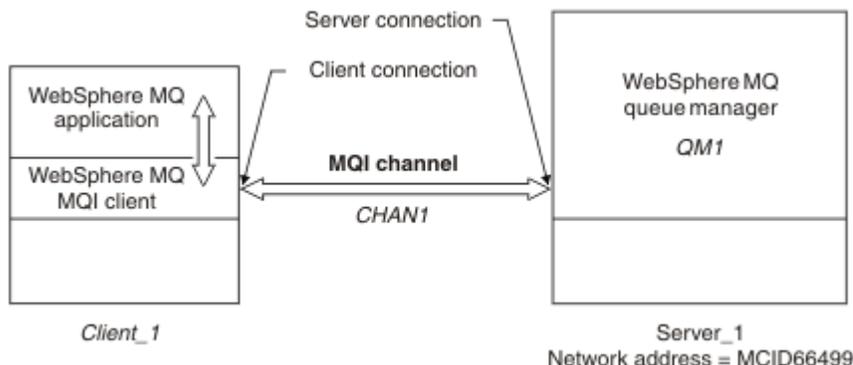


図 1. 簡単なチャンネル定義

簡単なチャンネル定義の例をさらにいくつか挙げます。

- Windows の場合:

```
SET MQSERVER=CHANNEL1/TCP/9.20.4.56
SET MQSERVER=CHANNEL1/NETBIOS/BOX643
```

- UNIX and Linux システムの場合:

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56'
export MQSERVER=CHANNEL1/LU62/BOX99
```

- **IBM i** IBM i の場合:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('CHANNEL1/TCP/9.20.4.56(1416)')
```

この BOX99 は、LU 6.2 ConnectionName です。

IBM MQ MQI client では、定義したチャンネルが **MQCONN** に提供された MQCNO 構造体が参照する MQCD 構造体でオーバーライドされていない限り、すべての **MQCONN** または **MQCONN** 要求で、このチャンネルの使用が試行されます。

注: **MQSERVER** 環境変数の詳細については、[80 ページの『MQSERVER』](#)を参照してください。

## MQCONN 呼び出しでの MQCNO 構造体の使用

IBM MQ MQI client・アプリケーションは、**MQCONN** 呼び出しで接続オプション構造体 MQCNO を使用して、クライアント接続チャンネルの定義を含むチャンネル定義構造体 MQCD を参照できます。

それにより、クライアント・アプリケーションは、実行時にチャンネルの **ChannelName**、**TransportType**、および **ConnectionName** 属性を指定でき、クライアント・アプリケーションは、同時に複数のサーバー・キュー・マネージャーに接続できます。

**MQSERVER** 環境変数を使用してチャンネルを定義した場合、実行時に **ChannelName**、**TransportType**、および **ConnectionName** 属性を指定できないことに注意してください。

また、クライアント・アプリケーションは、**MaxMsgLength** や **SecurityExit** など、チャンネルの属性を指定できます。それらの属性を指定することにより、クライアント・アプリケーションは、属性にデフォルト値以外の値を指定することができます。また、MQI チャンネルのクライアント側でチャンネル出口プログラムを呼び出すことができます。

チャンネルで Transport Layer Security (TLS) が使用されている場合、クライアント・アプリケーションは、MQCD 構造体で TLS に関する情報も提供できます。TLS に関連する追加情報は、TLS 構成オプション構造体 MQSCO で提供できます。この情報は、**MQCONN** 呼び出しで MQCNO 構造体によって参照することもできます。

MQCNO、MQCD、および MQSCO 構造体の詳細については、[MQCNO](#)、[MQCD](#)、および [MQSCO](#) を参照してください。

注: **MQCONN** のサンプル・プログラムは **amqscnxc** です。 **amqssslc** という別のサンプル・プログラムは、MQSCO 構造体の使用方法を示します。

## サーバー側でのサーバー接続およびクライアント接続の定義の作成

サーバー側で両方の定義を作成でき、次にクライアント接続定義をクライアントで使用可能にできます。

### このタスクについて

最初にサーバー接続チャンネルを定義し、次にクライアント接続チャンネルを定義します。

- すべてのプラットフォームで、IBM MQ Script (MQSC) コマンド、プログラマブル・コマンド・フォーマット (PCF) コマンドを使用して、サーバー・マシンでサーバー接続チャンネルを定義できます。

- **Windows** **Linux** Linux および Windows では、IBM MQ Explorer を使用することもできます。
- **z/OS** z/OS では、操作パネルおよび制御パネルも使用できます。
- **IBM i** IBM i では、パネル・インターフェースも使用できます。

サーバー側で作成されたクライアント接続チャンネルの定義は、クライアント・チャンネル定義テーブル (CCDT) を使用して、クライアントで使用可能にできます。

## 手順

1. サーバー接続チャンネルを定義するには、[44 ページの『サーバー側でのサーバー接続チャンネルの定義』](#)を参照してください。
2. クライアント接続チャンネルを定義するには、[45 ページの『サーバー側でのクライアント接続チャンネルの定義』](#)を参照してください。

## 関連概念

### [40 ページの『クライアント・チャンネル定義テーブル』](#)

クライアント・チャンネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャンネル定義と認証情報を判別します。Multiplatforms の場合、CCDT は自動的に作成されます。それを、クライアント・アプリケーションに対して使用可能にする必要があります。

## 関連タスク

### [44 ページの『サーバー側でのサーバー接続チャンネルの定義』](#)

キュー・マネージャー用にサーバー接続チャンネルの定義を作成します。

### [45 ページの『サーバー側でのクライアント接続チャンネルの定義』](#)

サーバー接続チャンネルの定義が完了したため、ここでは対応するクライアント接続チャンネルを定義します。

### [46 ページの『クライアント接続チャンネル定義へのアクセス』](#)

クライアント・チャンネル定義テーブル (CCDT) をクライアント・アプリケーションで使用できるようにするには、CCDT をコピーまたは共有してからクライアント・コンピュータでその位置と名前を指定します。

**V 9.0.0** IBM MQ 9.0 以降、この製品には、URL を使用してクライアント・チャンネル定義テーブル (CCDT) を見つける機能も用意されています。

## クライアント・チャンネル定義テーブル

クライアント・チャンネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャンネル定義と認証情報を判別します。Multiplatforms の場合、CCDT は自動的に作成されます。それを、クライアント・アプリケーションに対して使用可能にする必要があります。

クライアント・チャンネル定義テーブル (CCDT) の目的は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャンネル定義を判別することです。また、チャンネル定義は、接続に適用される認証情報を指定します。

CCDT はバイナリー・ファイルです。これは、キュー・マネージャーによって生成されます。キュー・マネージャーは、CCDT ファイルを読み取りません。

**Multi** マルチプラットフォームでは、CCDT は、キュー・マネージャーが作成されるときに作成されます。キュー・マネージャーに関連付けられている CCDT はオブジェクト定義との同期が維持されるため、クライアント・チャンネル・オブジェクトを定義、変更、削除する場合、同じ操作の一部としてキュー・マネージャー・オブジェクト定義と CCDT 内のエントリーの両方が更新されます。

## 注:

- IBM MQ CCDT ファイルの設計では、ユーザーが定義したすべてのクライアント接続チャンネルが実際に定義された後にのみ、CCDT ファイルが縮小されます。クライアント接続チャンネルが削除されると、CCDT ファイルで削除済みのマークが付けられますが、物理的に削除されたわけではありません。
- CCDT ファイルの縮小を強制するには、1 つ以上のクライアント接続チャンネルを削除した後に以下のコマンドを実行します。

```
rcrmqobj -m QM80 -t clchltab
```

CCDT を使用してクライアントに認証情報を提供し、TLS 証明書が失効しているかどうかの検査を行うことができます。認証情報オブジェクトが含まれる名前リストを定義し、キュー・マネージャー属性 **SSLCRLNameList** をその名前リストの名前に設定します。

## デフォルトの CCDT AMQCLCHL.TAB

### Multi

マルチプラットフォームでは、キュー・マネージャーを作成するときに AMQCLCHL.TAB というデフォルトの CCDT が作成されます。

デフォルトでは、AMQCLCHL.TAB はサーバー上の以下のディレクトリーに置かれています。

- **IBM i** IBM i では、統合ファイル・システムに置かれています。

```
/QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/&ipcc
```

- **Linux** **UNIX** UNIX and Linux システムの場合:

```
/prefix/qmgrs/QUEUEMANAGERNAME/@ipcc
```

UNIX and Linux システムでは、*QUEUEMANAGERNAME* によって参照されるディレクトリー名の大文字と小文字が区別されます。キュー・マネージャー名に特殊文字が含まれている場合、ディレクトリー名は、キュー・マネージャー名と同じにならない可能性があります。

- **Windows** Windows の場合:

```
MQ_INSTALLATION_PATH\data\qmgrs\QUEUEMANAGERNAME\@ipcc
```

*MQ\_INSTALLATION\_PATH* は、IBM MQ がインストールされている上位ディレクトリーを表します。

ただし、キュー・マネージャー・データ用に別のディレクトリーの使用を選択している場合もあります。**crtmqm** コマンドを使用した場合は、パラメーター **-md DataPath** を指定できます。その場合、AMQCLCHL.TAB は、指定した *DataPath* の @ipcc ディレクトリーに配置されます。

CCDT のパスは、MQCHLLIB を設定することで変更できます。なお、同じサーバー上に複数のキュー・マネージャーが存在する場合、MQCHLLIB を設定すると、それらのキュー・マネージャーは同じ CCDT ロケーションを共有します。

CCDT は、キュー・マネージャーの作成時に作成されます。CCDT の各項目は、特定のキュー・マネージャーへのクライアント接続を表します。**DEFINE CHANNEL** コマンドを使用してクライアント接続チャンネルを定義すると新しい項目が追加され、**ALTER CHANNEL** コマンドを使用してクライアント接続チャンネルを変更すると項目が更新されます。

## クライアント・チャンネル定義テーブルの場所

クライアント・アプリケーションが CCDT を使用する方法はいくつかあります。CCDT は、クライアント・コンピューターにコピーできます。複数のクライアントが共有する位置に、CCDT をコピーすることができます。CCDT がサーバー上に置かれたまま、共有ファイルとしてクライアントからアクセス可能にすることができます。

FTP を使用してファイルをコピーする場合は、bin オプションを使用してバイナリー・モードを設定してください。デフォルトの ASCII モードは使用しないでください。CCDT を使用可能にするためにどの方法を選択するにしても、位置が安全で、チャンネルに対する不正な変更から保護されている必要があります。

**V 9.0.0** IBM MQ 9.0 以降、URI を介してアクセスできる中央の場所で CCDT のホスティングが可能になり、デプロイされた各クライアントで個別に CCDT を更新する必要がなくなりました。IBM MQ 9.0 では、ネイティブ (C/C++、COBOL および RPG) および非管理 .NET アプリケーション用の機能が追加され、ローカル・ファイル、FTP、または HTTP リソースでも、CCDT を URL から取得できるようになりました。



**重要:** IBM MQ では、ファイル、FTP、または HTTP の URL からの CCDT の取得がサポートされません。

**V 9.0.0** IBM MQ クライアントのデフォルトのキャッシング動作では、CCDT ファイルの変更時刻が最後にファイルを取得した時刻と異なる場合にのみ、ファイルを取得します。ほとんどのクライアント構成オプションと同じく、次に示すさまざまな方法で URL の位置を指定できます。

- MQCNO 構造体で MQCONNX MQI 呼び出しに渡される CCDTUrlPtr/CCDTUrlOffset
- MQCCDTURL 環境変数
- mqclient.ini の Channels スタンザの ChannelDefinitionDirectory 属性

**V 9.0.0** 認証 URL および非認証 URL の両方がサポートされます。例えば、次のとおりです。

```
export MQCCDTURL=ftp://myuser:password@myhost.sample.com//var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

```
export MQCCDTURL=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

**V 9.0.0** このサポートを FTP または HTTP で使用する必要がある場合は、サーバーで CCDT ファイルをホストする必要があるということですが、IBM MQ 9.0 で追加されたサポートを使用すれば、すべてのクライアント・アプリケーションが自動的にチャンネル定義への変更を取得できるため、手動で更新を提供したり、ネットワーク・ファイル・システムを各クライアントでマウントする必要がなくなります。詳しくは、43 ページの『クライアント・チャンネル定義テーブルへの Web アドレス指定可能アクセス』を参照してください。

## runmqsc を使用してクライアント・マシン上に直接 CCDT を作成する方法

IBM MQ 8.0 以降では、**-n** パラメーターを指定して runmqsc コマンドを使用することにより、クライアント・マシン上に直接 CCDT を作成できます。CCDT は、MQCHLLIB で指定した場所に、MQCHLTAB で指定したファイル名 (デフォルトは AMQCLCHL.TAB) で作成されます。

**重要:** **-n** パラメーターを指定する場合は、その他のパラメーターを指定しないでください。

CCDT の各項目は、特定のキュー・マネージャーへのクライアント接続を表します。**DEFINE CHANNEL** コマンドを使用してクライアント接続チャンネルを定義すると新しい項目が追加され、**ALTER CHANNEL** コマンドを使用してクライアント接続チャンネルを変更すると項目が更新されます。

## クライアントで CCDT の位置を指定する方法

クライアント・システムでは、以下の方法で CCDT の位置を指定できます。

- 環境変数 MQCHLLIB を使用して、テーブルが位置するディレクトリーを指定します。また、環境変数 MQCHLTAB を使用してテーブルのファイル名を指定します。
- クライアント構成ファイルを使用します。CHANNELS スタンザで、属性 ChannelDefinitionDirectory を使用してテーブルが位置するディレクトリーを指定します。また、属性 ChannelDefinitionFile を使用してファイル名を指定します。
- **V 9.0.0** 中央の位置でホストされている CCDT の URL (file、ftp、または http) を指定します (41 ページの『クライアント・チャンネル定義テーブルの場所』参照)。

クライアント構成ファイルと環境変数の両方で場所が指定されている場合は、環境変数が優先します。この機能を使用して、標準の位置をクライアント構成ファイルで指定し、必要な場合に環境変数を使用して標準の位置をオーバーライドすることができます。

**V 9.0.0** URL を使用して CCDT の位置を指定する場合、ネイティブ・クライアント・アプリケーションがクライアント・チャンネル定義を検索する順序は、43 ページの『クライアント・チャンネル定義テーブルへの Web アドレス指定可能アクセス』に記載している順序になります。

### 関連資料

77 ページの『MQCHLLIB』

MQCHLLIB は、クライアント・チャンネル定義テーブル (CCDT) を含むファイルへのディレクトリー・パスを指定します。ファイルはサーバー上に作成されますが、IBM MQ MQI client・ワークステーションにコピーすることができます。

## 関連情報

[取り消された証明書の取り扱い](#)

## V 9.0.0 クライアント・チャンネル定義テーブルへの Web アドレス指定可能アクセス

IBM MQ 9.0 以降、この製品には、MQCNO を使用してプログラミングするか、環境変数を使用するか、または mqclient.ini ファイル・スタンザを使用することにより、URL を介してクライアント・チャンネル定義テーブル (CCDT) を見つける機能が用意されています。



**重要:** 環境変数オプションを使用できるのは、クライアントとして接続するネイティブ・プログラム、つまり C、COBOL、または C++ のアプリケーションに対してのみです。Java、JMS、および管理対象 .NET アプリケーションの場合、環境変数は効果がありません。

IBM MQ では、ファイル、FTP、または HTTP の URL からの CCDT の取得がサポートされます。

環境変数 75 ページの『MQCCDTURL』を使用すると、クライアント・チャンネル定義テーブルを取得できる単一値として file、ftp、または http の URL を指定できます。

また、既存のローカル・ファイル・システム・ディレクトリー /var/mqm の他に、77 ページの『MQCHLLIB』(または 62 ページの『クライアント構成ファイルの CHANNELS スタンザ』の下の **ChannelDefinitionDirectory** で指定されたもの) を使用して、file、ftp、または http URL のいずれかを介して、CCDT ファイルを位置指定することもできます。

77 ページの『MQCHLLIB』値はディレクトリー・ステムであり、完全修飾 URL を派生させるためには 79 ページの『MQCHLTAB』と組み合わせることで機能することに注意してください。

接続に関する基本認証は、URL 内にエンコードされる資格情報を使用してサポートされます:

### 認証される接続

```
export MQCHLLIB=ftp://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
```

### 認証されない接続

```
export MQCHLLIB=ftp://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=file:///var/mqm/qmgrs/QMGR/@ipcc
```

**注:** 認証される接続を使用する場合は、JMS と同様に、URL 内にエンコードされたユーザー名とパスワードを指定する必要があります。

クライアント・チャンネル定義を検索するため、ネイティブ・クライアント・アプリケーションの優先順位は次のとおりです:

1. MQCNO 内の **ClientConnOffset** および **ClientConnPtr** によって提供される MQCD。
2. MQCNO の **CCDTUrloffset** および **CCDTUrIPtr** によって指定される URL。
3. 80 ページの『MQSERVER』環境変数。
4. mqclient.ini ファイルが定義され、ServerConnectionParms を含んでいる場合、それが定義しているチャンネルが使用されます。詳しくは、52 ページの『構成ファイルを使用したクライアントの構成』および 62 ページの『クライアント構成ファイルの CHANNELS スタンザ』を参照してください。
5. 75 ページの『MQCCDTURL』環境変数。
6. 77 ページの『MQCHLLIB』および 79 ページの『MQCHLTAB』環境変数。
7. 62 ページの『クライアント構成ファイルの CHANNELS スタンザ』の **ChannelDefinitionDirectory**。

**重要:** URL を使用して CCDT ファイルにアクセスすると、file:// プロトコルを使用した場合でも、常にファイルの読み取り専用コピーが開きます。

例えばクライアントから **runmqsc** DEFINE CHANNEL コマンドを使用するときに、書き込みアクセス用に CCDT ファイルを開こうとすると、書き込みアクセス用にファイルを開けなかったことを示すエラー・メッセージが返されます。

ただし、**runmqsc** を使用してチャンネルおよび認証情報の定義を読み取ることはできます。

## 関連概念

### 40 ページの『クライアント・チャンネル定義テーブル』

クライアント・チャンネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャンネル定義と認証情報を判別します。Multiplatforms の場合、CCDT は自動的に作成されます。それを、クライアント・アプリケーションに対して使用可能にする必要があります。

## 関連タスク

### 46 ページの『クライアント接続チャンネル定義へのアクセス』

クライアント・チャンネル定義テーブル (CCDT) をクライアント・アプリケーションで使用できるようにするには、CCDT をコピーまたは共有してからクライアント・コンピュータでその位置と名前を指定します。

**V 9.0.0** IBM MQ 9.0 以降、この製品には、URL を使用してクライアント・チャンネル定義テーブル (CCDT) を見つける機能も用意されています。

## 関連情報

### CCDTURL

[IBM MQ classes for JMS での CCDT の使用](#)

[XMSC\\_WMQ CCDTURL](#)

## Windows Active Directory 内のクライアント接続チャンネル

Active Directory をサポートする Windows システムでは、IBM MQ は Active Directory 内のクライアント接続チャンネルを公開して、動的クライアント/サーバーのバインディングを提供します。

クライアント接続チャンネル・オブジェクトが定義される場合、それらのオブジェクトは、デフォルトで AMQCLCHL.TAB と呼ばれるクライアント・チャンネル定義ファイルに書き込まれます。クライアント接続チャンネルが TCP/IP プロトコルを使用する場合、IBM MQ サーバーは、Active Directory 内のクライアント接続チャンネルも公開します。IBM MQ クライアントがサーバーへの接続方法を決定するとき、以下の検索順序で、関連するクライアント接続チャンネルのオブジェクト定義を検索します。

1. MQCONNX MQCD データ構造
2. MQSERVER 環境変数
3. クライアント・チャンネル定義ファイル
4. Active Directory

この順序であれば、現行のアプリケーションは変更によって何も影響を受けることはありません。Active Directory 内のこれらの項目をクライアント・チャンネル定義ファイル内のレコードと見なすことができます。IBM MQ クライアントは同じ方法でこれらの項目を処理します。Active Directory でクライアント接続チャンネル定義を公開するためのサポートを構成および管理するには、[setmqscp](#) に記載されているとおり、`setmqscp` コマンドを使用します。

## サーバー側でのサーバー接続チャンネルの定義

キュー・マネージャー用にサーバー接続チャンネルの定義を作成します。

## 手順

1. サーバー・マシン側で、選択した名前をもつチャンネル、およびサーバー接続のチャンネル・タイプを定義します。  
以下に例を示します。

```
DEFINE CHANNEL(CHAN2) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
DESCR('Server-connection to Client_2')
```

2. 以下のコマンドを使用して、キュー・マネージャーへのインバウンド接続アクセスを許可します。

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- ここで、SET CHLAUTH は前のステップで定義されたチャンネルの名前を使用します。
- 'IP address' とは、クライアントの IP アドレスのことです。
- 'userid' とは、ターゲット・キューへのアクセス制御のためにチャンネルに提供する ID のことです。このフィールドは、大文字と小文字が区別されます。

いくつかの異なる属性を使用して、インバウンド接続の識別を選択することができます。例では、IP アドレスを使用しています。代替属性には、クライアントのユーザー ID および TLS のサブジェクト識別名が含まれています。詳しくは、[チャンネル認証レコード](#)を参照してください。

このチャンネル定義は、サーバー上で稼働するキュー・マネージャーに関連しています。

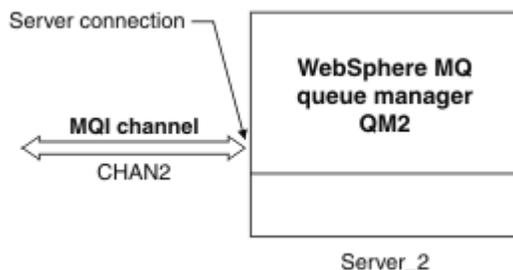


図 2. サーバー接続チャンネルの定義

## サーバー側でのクライアント接続チャンネルの定義

サーバー接続チャンネルの定義が完了したため、ここでは対応するクライアント接続チャンネルを定義します。

### 始める前に

サーバー接続チャンネルを定義します。

### 手順

1. サーバー接続チャンネルと同じ名前で、チャンネル・タイプがクライアント接続のチャンネルを定義します。接続名 (CONNNAME) を指定する必要があります。TCP/IP の場合、接続名はサーバー・マシンのネットワーク・アドレスまたはホスト名です。また、クライアント環境で実行される IBM MQ アプリケーションの接続先のキュー・マネージャー名 (QMNAME) を指定することをお勧めします。キュー・マネージャー名を変更することにより、異なるキュー・マネージャーに接続するチャンネルのセットを定義できます。

```
DEFINE CHANNEL(CHAN2) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME(9.20.4.26) QMNAME(QM2) DESCR('Client-connection to Server_2')
```

2. 以下のコマンドを使用して、キュー・マネージャーへのインバウンド接続アクセスを許可します。

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP-address') MCAUSER('userid')
```

- ここで、SET CHLAUTH は前のステップで定義されたチャンネルの名前を使用します。
- 'IP address' とは、クライアントの IP アドレスのことです。
- 'userid' とは、ターゲット・キューへのアクセス制御のためにチャンネルに提供する ID のことです。このフィールドは、大文字と小文字が区別されます。

いくつかの異なる属性を使用して、インバウンド接続の識別を選択することができます。例では、IP アドレスを使用しています。代替属性には、クライアントのユーザー ID および TLS のサブジェクト識別名が含まれています。詳しくは、[チャンネル認証レコード](#)を参照してください。

## タスクの結果

**Multi** マルチプラットフォームでは、このチャンネル定義は、キュー・マネージャーに関連付けられたクライアント・チャンネル定義テーブル (CCDT) と呼ばれるファイルに格納されます。クライアント・チャンネル定義テーブルには、複数のクライアント接続チャンネル定義を組み込むことができます。クライアント・チャンネル定義テーブルの詳細と、クライアント接続チャンネル定義を z/OS に格納する方法については、40 ページの『クライアント・チャンネル定義テーブル』を参照してください。

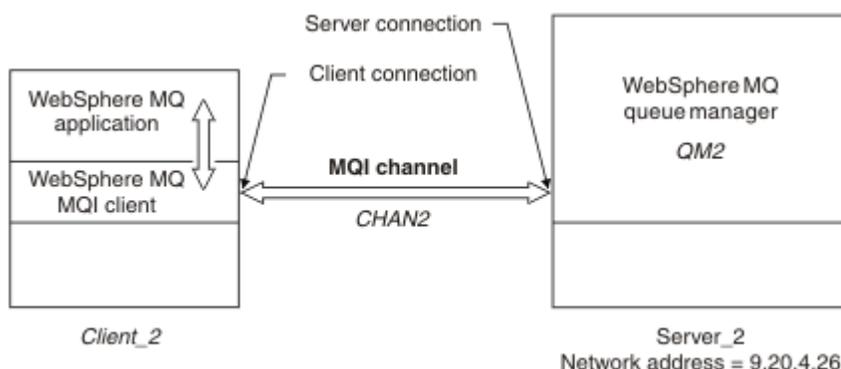


図 3. クライアント接続チャンネルの定義

## クライアント接続チャンネル定義へのアクセス

クライアント・チャンネル定義テーブル (CCDT) をクライアント・アプリケーションで使用できるようにするには、CCDT をコピーまたは共有してからクライアント・コンピューターでその位置と名前を指定します。

**V 9.0.0** IBM MQ 9.0 以降、この製品には、URL を使用してクライアント・チャンネル定義テーブル (CCDT) を見つける機能も用意されています。

## 始める前に

必要なクライアント接続チャンネルが定義済みであること。

**z/OS** z/OS の場合、CCDT を作成済みであること。

**Multi** マルチプラットフォームの場合は、CCDT が自動的に作成/更新されます。

## このタスクについて

クライアント・アプリケーションでクライアント・チャンネル定義テーブル (CCDT) を使用するには、CCDT をアプリケーションで使用可能にして、位置と名前を指定する必要があります。これはいくつかの方法で行うことができます。

- CCDT をクライアント・コンピューターにコピーすることができます。
- 複数のクライアントが共有する位置に、CCDT をコピーすることができます。
- CCDT がサーバー上に置かれたまま、共有ファイルとしてクライアントからアクセス可能にすることができます。

**V 9.0.0** IBM MQ 9.0 以降、IBM MQ のネイティブ (C/C++、COBOL および RPG) および非管理 .NET アプリケーションは、中央の位置でホストされているローカル・ファイル、FTP または HTTP リソースの CCDT を URL から取得できるようになりました。

## 手順

1. 以下のいずれかの方法で、CCDT をクライアント・アプリケーションで使用できるようにします。

- a) オプション: クライアントのコンピューターに CCDT をコピーします。
- b) オプション: 複数のクライアントが共有する位置に CCDT をコピーします。
- c) オプション: CCDT をサーバーに残しますが、クライアントが共有できるようにします。
- d) **V 9.0.0**  
オプション: 中央の位置でホストする CCDT に対するローカル・ファイル、ftp、または http の URL を定義して、その URL からネイティブ (C/C++、COBOL および RPG) および非管理 .NET アプリケーションが CCDT を取得できるようにします。

CCDT の位置としていずれを選択する場合でも、位置は安全で、チャンネルに対する不正な変更から保護されている必要があります。

2. クライアント上で、CCDT が含まれるファイルの位置と名前を、次の 3 つのいずれかの方法で指定します。

- a) オプション: クライアント構成ファイルの CHANNELS スタンザを使用します。詳しくは、[62 ページの『クライアント構成ファイルの CHANNELS スタンザ』](#)を参照してください。
- b) オプション: 環境変数 MQCHLLIB および MQCHLTAB を使用します。

例えば、次のように入力して環境変数を設定できます。

- UNIX and Linux システムの場合:

```
export MQCHLLIB= MQ_INSTALLATION_PATH/qmgrs/ QUEUEMANAGERNAME /@ipcc
export MQCHLTAB=AMQCLCHL.TAB
```

- **IBM i** IBM i の場合:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE('/QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/@ipcc')
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(AMQCLCHL.TAB)
```

MQ\_INSTALLATION\_PATH は、IBM MQ がインストールされている上位ディレクトリーを表します。

- c) オプション: Windows でのみ、**setmqscp** 制御コマンドを使用して、Active Directory でクライアント接続チャンネル定義を公開することができます。
- d) **V 9.0.0**  
MQCNO を使用してプログラミングするか、環境変数を使用するか、mqclient.ini ファイルのスタンザを使用して、中央でホストされている CCDT を URL を介して指定します。詳しくは、[41 ページの『クライアント・チャンネル定義テーブルの場所』](#) および [43 ページの『クライアント・チャンネル定義テーブルへの Web アドレス指定可能アクセス』](#)を参照してください。

MQSERVER 環境変数が設定されている場合、IBM MQ クライアントは、MQSERVER によって指定されたクライアント接続チャンネル定義を、クライアント・チャンネル定義テーブル内のどの定義よりも優先して使用します。

## 関連概念

[40 ページの『クライアント・チャンネル定義テーブル』](#)

クライアント・チャンネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャンネル定義と認証情報を判別します。Multiplatforms の場合、CCDT は自動的に作成されます。それを、クライアント・アプリケーションに対して使用可能にする必要があります。

[43 ページの『クライアント・チャンネル定義テーブルへの Web アドレス指定可能アクセス』](#)

IBM MQ 9.0 以降、この製品には、MQCNO を使用してプログラミングするか、環境変数を使用するか、または mqclient.ini ファイル・スタンザを使用することにより、URL を介してクライアント・チャンネル定義テーブル (CCDT) を見つける機能が用意されています。

## 関連情報

[MQI クライアント: クライアント・チャンネル定義テーブル \(CCDT\)](#)

## ULW MQI チャンネル用のチャンネル出口プログラム

UNIX, Linux, and Windows 上の IBM MQ MQI client 環境では、3つのタイプのチャンネル出口を使用できます。

次のとおりです。

- 送信出口
- 受信出口
- セキュリティー出口

これらの出口は、チャンネルのクライアント側とサーバー側の両方で使用できます。MQSERVER 環境変数を指定している場合はアプリケーションで出口を使用することはできません。チャンネル出口については、[チャンネル・メッセージ用のチャンネル出口プログラム](#)で説明しています。

送信出口と受信出口は同時に機能します。これらの出口を使用する方法は、次のようにいくつかあります。

- メッセージを分割し、再アセンブルする。
- メッセージ内のデータを圧縮し、解凍する(この機能は、IBM MQの一部として提供されていますが、別の圧縮技術を使用することもできます)。
- ユーザー・データを暗号化し、復号する(この機能は、IBM MQの一部として提供されていますが、別の暗号化技術を使用することもできます)。
- 送受信される各メッセージのジャーナル記録を行う。

セキュリティー出口を使用して、IBM MQ クライアントとサーバーが正しく識別されるようにし、アクセスを制御することができます。

チャンネル・インスタンスのサーバー接続サイドの送信または受信出口が、関連先の接続上でMQI呼び出しを実行する必要がある場合、MQCXP Hconn フィールドで提供される接続ハンドルを使用します。注意すべき点として、クライアント接続の送信および受信出口はMQI呼び出しを実行できません。

### 関連概念

[49 ページの『クライアント接続でのセキュリティー出口』](#)

セキュリティー出口プログラムを使用して、チャンネルの反対側のパートナーが正しいかを確認することができます。セキュリティー出口がクライアント接続に適用される場合は、特別な考慮事項が適用されます。

### 関連資料

[48 ページの『出口へのパス』](#)

チャンネル出口の位置へのデフォルト・パスは、クライアント構成ファイルで定義されます。チャンネルが初期化されるときに、チャンネル出口がロードされます。

[50 ページの『送信または受信出口プログラムでのAPI呼び出しの識別』](#)

クライアントにMQIチャンネルを使用する際に、エージェント・バッファの10バイト目で、送信または受信出口が呼び出されるときに使用されるAPI呼び出しを識別します。これは、ユーザー・データを含み、かつ暗号化やデジタル署名などの処理を必要とする可能性があるチャンネル・フローを識別する場合に役立ちます。

### 関連情報

[キュー・マネージャーの機能の拡張](#)

[ユーザー出口、API出口、およびIBM MQインストール可能サービス](#)

## ULW 出口へのパス

チャンネル出口の位置へのデフォルト・パスは、クライアント構成ファイルで定義されます。チャンネルが初期化されるときに、チャンネル出口がロードされます。

UNIX, Linux, and Windows システムでは、IBM MQ MQI client のインストール中に、クライアント構成ファイルがシステムに追加されます。次のスタンプを使用して、クライアント上のチャンネル出口の位置へのデフォルト・パスがこのファイルで定義されます。

```
ClientExitPath:
```

```
ExitsDefaultPath= string
ExitsDefaultPath64= string
```

この *string* は、プラットフォームに適した形式のファイルの場所です。

チャンネルが初期設定されると、MQCONN 呼び出しまたは MQCONNX 呼び出しの後で、クライアント構成ファイルが検索されます。ClientExitPath スタンザが読み込まれ、チャンネル定義で指定されている任意のチャンネル出口がロードされます。

## ULW クライアント接続でのセキュリティー出口

セキュリティー出口プログラムを使用して、チャンネルの反対側のパートナーが正しいかを確認することができます。セキュリティー出口がクライアント接続に適用される場合は、特別な考慮事項が適用されます。

50 ページの図 4 は、クライアント接続でのセキュリティー出口の使用を示しており、ここでは IBM MQ オブジェクト権限マネージャーを使用してユーザーを認証します。SecurityParmsPtr または SecurityParmsOffset のいずれかがクライアントの MQCNO 構造体に設定され、チャンネルの両側にセキュリティー出口があります。通常のセキュリティー・メッセージの交換が終了して、チャンネルが実行可能になると、MQCXP SecurityParms フィールドからアクセスされた MQCSP 構造体がクライアントのセキュリティー出口に渡されます。出口タイプは MQXR\_SEC\_PARMS に設定されています。セキュリティー出口では、ユーザー ID およびパスワードに対して何も行わないように選択することも、どちらかを変更することも、あるいは両方を変更することもできます。次に、出口から返されたデータがチャンネルのサーバー接続側に送信されます。MQCSP 構造体がチャンネルのサーバー接続側で再ビルドされ、MQCXP SecurityParms フィールドからアクセスされたサーバー接続セキュリティー出口に渡されます。セキュリティー出口はこのデータを受信して処理します。この処理では通常、ユーザー ID およびパスワード・フィールドに対してクライアント出口で行われた変更が元に戻されます。これは、キュー・マネージャー接続を許可するために使用されます。結果として得られた MQCSP 構造体は、キュー・マネージャー・システム上の MQCNO 構造体の SecurityParmsPtr を使用して参照されます。

MQCXP SecurityParms フィールドによって戻されるメモリー・アドレスは、アドレス指定が可能で、MQXR\_TERM まで変更のないままでなければなりません。出口が MQXR\_TERM に対して呼び出される前に、その出口がメモリーを無効にしたり、解放してシステムに戻したりすることはできません。

SecurityParmsPtr または SecurityParmsOffset が MQCNO 構造体に設定され、セキュリティー出口がチャンネルの片側のみにある場合は、セキュリティー出口は MQCSP 構造体を受信して処理します。補足的な処置を実行する出口がないため、暗号化などの処置は単一のユーザー出口には不適切です。

SecurityParmsPtr および SecurityParmsOffset が MQCNO 構造体に設定されておらず、セキュリティー出口がチャンネルの片側または両側にある場合は、セキュリティー出口 (複数の場合あり) が呼び出されます。どのセキュリティー出口からでも、SecurityParmsPtr を使用して処理された各自の MQCSP 構造体を返すことができます。セキュリティー出口は、終了される (MQXR\_TERM の ExitReason) まで再呼び出しされません。この段階で、出口書き込みプログラムによって、MQCSP で使用されたメモリーを解放することができます。

サーバー接続チャンネル・インスタンスが複数の会話を共有している場合は、セキュリティー出口の呼び出しパターンが、2 番目以降の会話に対して制限されます。

最初の会話のパターンは、チャンネル・インスタンスが会話を共有していない場合と同じです。2 番目以降の会話の場合、MQXR\_INIT、MQXR\_INIT\_SEC または MQXR\_SEC\_MSG を使用して、セキュリティー出口が呼び出されることは絶対にありません。呼び出しは MQXR\_SEC\_PARMS を使用して行われます。

共有会話を行うチャンネル・インスタンスで MQXR\_TERM が呼び出されるのは、最後に実行された会話に対してのみです。

各会話では、出口の MQXR\_SEC\_PARMS 呼び出しで MQCD を変更することができます。このフィーチャーは、チャンネルのサーバー接続側で、例えば、キュー・マネージャーへの接続を確立する前に、MCAUserIdentifier 値または LongMCAUserIdPtr 値を変更する場合に役立ちます。

Server-connection exit	Client-connection exit
	Invoked with MQXR_INIT Responds with MQXCC_OK
Invoked with MQXR_INIT Responds with MQXCC_OK	
	Invoked with MQXR_INIT_SEC Responds with MQXCC_OK
Invoked with MQXR_INIT_SEC Responds with MQXCC_OK	
	Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK
Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK	
Data transfer begins	
Invoked with MQXR_TERM Responds with MQXCC_OK	Invoked with MQXR_TERM Responds with MQXCC_OK

図 4. クライアント接続で開始された、セキュリティー・パラメーターを使用したクライアント接続に対する合意のやりとり

注: IBM WebSphere MQ 7.1 のリリースより前に構成されたセキュリティー出口アプリケーションは、更新が必要になる場合があります。詳しくは、[チャンネル・セキュリティー出口プログラム](#)を参照してください。

### ULW 送信または受信出口プログラムでの API 呼び出しの識別

クライアントに MQI チャンネルを使用する際に、エージェント・バッファーの 10 バイト目で、送信または受信出口が呼び出されるときに使用される API 呼び出しを識別します。これは、ユーザー・データを含み、かつ暗号化やデジタル署名などの処理を必要とする可能性があるチャンネル・フローを識別する場合に役立ちます。

次の表には、API 呼び出しの処理中にチャンネル・フローの 10 バイト目に出現するデータを示します。

注: このバイトには、これらの値のみが出現するわけではありません。その他に予約済みの値があります。

API 呼び出し	要求用のバイト 10 の値	応答用のバイト 10 の値
MQCONN <a href="#">51 ページの『1』</a> , <a href="#">51 ページの『2』</a>	X'81'	X'91'
MQDISC <a href="#">51 ページの『1』</a>	X'82'	X'92'
MQOPEN <a href="#">51 ページの『3』</a>	X'83'	X'93'
MQCLOSE	X'84'	X'94'

表 7. API 呼び出しの識別 (続き)

API 呼び出し	要求用のバイト 10 の値	応答用のバイト 10 の値
MQGET <a href="#">51 ページの『4』</a>	X'85'	X'95'
MQPUT <a href="#">51 ページの『4』</a>	X'86'	X'96'
MQPUT1 要求 <a href="#">51 ページの『4』</a>	X'87'	X'97'
MQSET 要求	X'88'	X'98'
MQINQ 要求	X'89'	X'99'
MQCMIT 要求	X'8A'	X'9A'
MQBACK 要求	X'8B'	X'9B'
MQSTAT 要求	X'8D'	X'9D'
MQSUB 要求	X'8E'	X'9E'
MQSUBRQ 要求	X'8F'	X'9F'
xa_start 要求	X'A1'	X'B1'
xa_end 要求	X'A2'	X'B2'
xa_open 要求	X'A3'	X'B3'
xa_close 要求	X'A4'	X'B4'
xa_prepare 要求	X'A5'	X'B5'
xa_commit 要求	X'A6'	X'B6'
xa_rollback 要求	X'A7'	X'B7'
xa_forget 要求	X'A8'	X'B8'
xa_recover 要求	X'A9'	X'B9'
xa_complete 要求	X'AA'	X'BA'

**注:**

1. クライアントとサーバーの間の接続は、クライアント・アプリケーションによって MQCONN を使用して開始されます。そのため、特にこのコマンド用に、その他のいくつかのネットワーク・フローがあります。同じことが、ネットワーク接続を終了する MQDISC にも該当します。
2. MQCONNX は、クライアント/サーバー接続において、MQCONN と同じ方法で処理されます。
3. 大きな配布リストを開いている場合、必要なすべてのデータを SVRCONN MCA に渡すために、MQOPEN 呼び出しごとに複数のネットワーク・フローが発生する可能性があります。
4. 大容量のメッセージは伝送セグメント・サイズを超過します。この場合、単一の API 呼び出しから多くのネットワーク・フローが発生する可能性があります。

**z/OS キュー共有グループへのクライアントの接続**

キュー共有グループのメンバーであるサーバー上のクライアントとキュー・マネージャーとの間に、MQI チャンネルを作成することによって、キュー共有グループにクライアントを接続することができます。

**このタスクについて**

キュー共有グループは、同じ共有キューのセットにアクセスできるキュー・マネージャーのセットにより構成されます。共有キューについては、[共有キューとキュー共有グループ](#)を参照してください。

共有キューにおかれたクライアントは、このキュー共有グループの任意のメンバーに接続することができます。キュー共有グループに接続すると、フロントエンドおよびバックエンドの可用性が増し、容量が増すという利点があります。特定のキュー・マネージャー、または汎用インターフェースに接続できます。

キュー共有グループにあるキュー・マネージャーに直接接続すると、共有ターゲット・キューにメッセージを入れられるという利点があるので、バックエンドの可用性を増すことができます。

キュー共有グループの汎用インターフェースに接続すると、グループ内のいずれかのキュー・マネージャーを使用してセッションをオープンできます。これにより、クライアント・キュー・マネージャーがグループ内の任意のキュー・マネージャーに接続できるため、フロントエンドの可用性が増すことになります。キュー共有グループにある特定のキュー・マネージャーに接続する必要がない場合には、汎用インターフェースを使用してこのグループに接続します。

汎用インターフェースは、シスプレックス・ディストリビューター VIPA アドレスまたは VTAM 汎用リソース名、あるいはキュー共有グループへの別の共通インターフェースにすることができます。汎用インターフェースのセットアップについての詳細は、[キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ](#)を参照してください。

## 手順

キュー共有グループの汎用インターフェースに接続するには、グループ内の任意のキュー・マネージャーがアクセスできるチャンネル定義を作成する必要があります。このために、グループ内のそれぞれのキュー・マネージャーで同じ定義にする必要があります。

1. 次の例に示されているように SVRCONN チャンネルを定義します。

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
QSGDISP(GROUP)
```

サーバー上のチャンネル定義は、共有 Db2® リポジトリに保管されます。キュー共有グループにある個々のキュー・マネージャーは定義のローカル・コピーを作成し、MQCONN または MQCONNX 呼び出しを発行する際に正しいサーバー接続チャンネルに常に接続できるようにします。

2. 次の例に示されているように CLNTCONN チャンネルを定義します。

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME( VIPA address ) QMNAME(QSG1) +
DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
```

## タスクの結果

クライアント接続チャンネルにある CONNAME フィールドに、キュー共有グループの汎用インターフェースが保管されるため、このグループの任意のキュー・マネージャーに接続してから、そのグループが所有する共有キューに入れることができるようになります。

## 構成ファイルを使用したクライアントの構成

テキスト・ファイル内の属性を使用して、クライアントを構成します。この属性は、環境変数、または他のプラットフォーム固有の方法を使用してオーバーライドすることができます。

### このタスクについて

UNIX and Linux プラットフォームで使用されるキュー・マネージャー設定ファイル `qm.ini` に似た IBM MQ MQI clients テキスト・ファイルを使用して、設定を行います。ファイルには多数のスタンザがあり、各スタンザには **attribute-name=value** 形式の行が多数含まれています。

IBM MQ MQI client 通常、構成ファイルは `mqclient.ini`、という名前になりますが、別の名前を付けることを選択できます。このファイルの構成情報は、すべてのプラットフォーム、および以下を使用するクライアントに適用されます。

- MQI

- IBM MQ classes for Java
- IBM MQ classes for JMS
- IBM MQ classes for .NET
- XMS

IBM MQ MQI client 構成ファイル内の属性はほとんどの IBM MQ クライアントに適用されますが、管理対象の .NET クライアントまたは XMS .NET クライアントによって、読み取られない属性、IBM MQ classes for Java または IBM MQ classes for JMS を使用するクライアントによって読み取られない属性がいくつかあります。詳しくは、[55 ページの『各属性を読み取ることができる IBM MQ クライアント』](#)を参照してください。

構成機能は、クライアント・アプリケーションがあらゆるキュー・マネージャーに対して行うすべての接続に適用されます。1つのキュー・マネージャーに対する個々の接続に固有のものではありません。個々のキュー・マネージャーへの接続に関する属性は、プログラムで構成することができます。例えば、MQCD 構造体やクライアント・チャンネル定義テーブル (CCDT) を使用して構成可能です。

以下にクライアント構成ファイルの例を示します。

```

** Module Name: mqclient.ini                **
** Type       : IBM MQ MQI client configuration file      **
** Function   : Define the configuration of a client      **
**          **                                          **
** Notes     :                                          **
** 1) This file defines the configuration of a client     **
**          **                                          **
ClientExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

TCP:
  Library1=DLLName1
  KeepAlive = Yes
  ClntSndBuffSize=32768
  ClntRcvBuffSize=32768
  Connect_Timeout=0

MessageBuffer:
  MaximumSize=-1
  Updatepercentage=-1
  PurgeTime=0

LU62:
  TPName
  Library1=DLLName1
  Library2=DLLName2

PreConnect:
  Module=myMod
  Function=myFunc
  Data=ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
  Sequence=1

CHANNELS:
  DefRecon=YES
  ServerConnectionParms=SALES.SVRCONN/TCP/hostname.x.com(1414)

```

クライアント構成ファイルを使用して複数のチャンネル接続をセットアップすることはできません。

IBM WebSphere MQ 7.0 より前のリリースでサポートされていた環境変数は後のリリースでの引き続きサポートされます。このような環境変数がクライアント構成ファイル内の同等の値と一致する場合は、環境変数によってクライアント構成ファイルの値がオーバーライドされます。

IBM MQ classes for JMS を使用するクライアント・アプリケーションの場合は、以下の方法でクライアント構成ファイルをオーバーライドすることもできます。

- JMS 構成ファイルのプロパティを設定する。

- Java システム・プロパティを設定する (このプロパティは JMS 構成ファイルもオーバーライドします)。

.NET クライアントの場合は、.NET アプリケーション構成ファイルを使用して、クライアント構成ファイルおよび同等の環境変数をオーバーライドすることもできます。

## Linux UNIX 構成ファイルのコメント

セミコロン「;」とハッシュ「#」の文字を使用して、構成ファイル内でコメントの先頭にマークを付けることができます。これにより、行全体をコメントとしてマークしたり、設定の値に含まれることのない行末のコメントを表示したりできます。

値にこれらの文字のいずれかが必要な場合は、円記号文字「\」を使用してその文字をエスケープする必要があります。

以下の例は、構成ファイルでのコメントの使用方法を示しています。

```
# Example of an SSL stanza with comments
SSL:
  ClientRevocationChecks=REQUIRED ; Example of an end of line comment
  SSLCryptoHardware=GSK_PKCS11=/driver\;label\;password\;SYMMETRIC_CIPHER_ON # Example of
  escaped comment characters.
```

## 手順

- 以下のトピックの情報を使用して、クライアントを構成します。
  - [54 ページの『クライアント構成ファイルの場所』](#)
  - [55 ページの『各属性を読み取ることができる IBM MQ クライアント』](#)

## クライアント構成ファイルの場所

IBM MQ MQI client 構成ファイルは複数の場所で保持することができます。

クライアント・アプリケーションは、以下の検索パスを使用して IBM MQ MQI client 構成ファイルを見つけます。

### 1. 環境変数 MQCLNTCF で指定された場所。

この環境変数の形式は、完全な URL です。これは、ファイル名が必ずしも `mqclient.ini` とは限らない場合があることを意味し、これにより、ネットワークに接続されているファイル・システムにファイルを配置することが容易になります。

次の事項に注意してください。

- C、.NET、および XMS クライアントは `file:` プロトコルのみをサポートします。URL スtring が `protocol:` で始まらない場合、`file:` プロトコルが想定されます。
- 環境変数の読み取りをサポートしていない Java 1.4.2 JRE を有効にする場合、MQCLNTCF 環境変数を MQCLNTCF Java システム・プロパティでオーバーライドできます。

### 2. アプリケーションの現行作業ディレクトリー内の `mqclient.ini` というファイル。

### 3. Windows、UNIX and Linux システムの IBM MQ データ・ディレクトリーにある `mqclient.ini` という名前のファイル。

次の事項に注意してください。

- IBM MQ データ・ディレクトリーは、IBM i や z/OS などの特定のプラットフォームには存在しません。また、クライアントが別の製品と一緒に提供されている場合もあります。
- UNIX and Linux システムでは、ディレクトリーは `/var/mqm` です。
- Windows プラットフォームでは、データ・ディレクトリーを指すように、インストール時に環境変数 `MQ_DATA_PATH` を構成します。これは通常、`C:\ProgramData\IBM\MQ` です。

注：クライアントのみをインストールする場合、環境変数は `MQ_FILE_PATH` のようになります。

- 環境変数の読み取りをサポートしない Java 1.4.2 JRE を使用できるようにするには、MQ\_DATA\_PATH 環境変数を MQ\_DATA\_PATH Java システム・プロパティーで手動でオーバーライドします。
4. 各プラットフォームに該当する、ユーザー・アクセス可能な標準ディレクトリー内の mqclient.ini というファイル。
- すべての Java クライアントでは、これは user.home Java システム・プロパティーの値になります。
  - UNIX and Linux プラットフォーム上の C クライアントでは、これは HOME 環境変数の値になります。
  - Windows 上の C クライアントでは、これは HOMEDRIVE 環境変数および HOMEPATH 環境変数を連結した値になります。

## 各属性を読み取ることができる IBM MQ クライアント

IBM MQ MQI client 構成ファイル内のほとんどの属性は、C クライアントおよび非管理対象 .NET クライアントで使用できます。ただし、一部の属性は、管理対象 .NET および XMS .NET クライアントによって、または IBM MQ classes for Java や IBM MQ classes for JMS を使用するクライアントによって読み取られません。

表 8. 各タイプのクライアントに適用される属性						
mqclient.ini スタンザ名および属性	説明	C および非管理対象 .NET	Java	JMS	管理対象 .NET	管理対象 XMS .NET
<b>CHANNELS</b> スタンザ						
<u>CCSID</u>	使用するコード化文字セット番号。	Yes	No	No	Yes	Yes
<u>ChannelDefinitionDirectory</u>	クライアント・チャネル定義テーブルを含むファイルのディレクトリー・パス。	Yes	No	No	Yes	Yes
<u>ChannelDefinitionFile</u>	クライアント・チャネル定義テーブルを含むファイルの名前。	Yes	No	No	Yes	Yes
<u>ReconDelay</u>	自動再接続が可能なクライアント・プログラムに対して再接続遅延を構成するための管理オプション。	Yes	No	Yes	Yes	Yes

表 8. 各タイプのクライアントに適用される属性 (続き)

<b>mqclient.ini</b> スタンザ名および属性	説明	<b>C</b> および非管理対象 <b>.NET</b>	<b>Java</b>	<b>JMS</b>	管理対象 <b>.NET</b>	管理対象 <b>XMS .NET</b>
<a href="#">DefRecon</a>	クライアント・プログラムが自動的に再接続できるようにしたり、自動的に再接続するように作成されたクライアント・プログラムの自動再接続を無効にしたりするための管理オプション。	Yes	No	Yes	Yes	Yes
<a href="#">MQReconnectTimeout</a>	クライアントに再接続するタイムアウト(秒)。	Yes	No	No	Yes	No
<a href="#">ServerConnectionParms</a>	IBM MQ サーバーの位置および使用する通信方式。	Yes	No	No	Yes	Yes
<a href="#">Put1DefaultAlwaysSync</a>	MQPMO_RESPONSE_AS_QDEF オプションを使用して、MQPUT1 関数呼び出しの動作を制御します。	Yes	Yes	Yes	Yes	Yes
<a href="#">PasswordProtection</a>	SSL または TLS を使用する代わりに、保護されたパスワードを MQCSP 構造に設定できます。	Yes	Yes	Yes	Yes	Yes
<b>ClientExitPath</b> スタンザ						
<a href="#">ExitsDefaultPath</a>	クライアントの 32 ビット・チャネル出口の位置を指定します。	Yes	Yes	Yes	Yes	Yes

表 8. 各タイプのクライアントに適用される属性 (続き)

<b>mqclient.ini</b> スタンザ名および属性	説明	<b>C</b> および非管理対象 <b>.NET</b>	<b>Java</b>	<b>JMS</b>	管理対象 <b>.NET</b>	管理対象 <b>XMS .NET</b>
<a href="#">ExitsDefaultPath64</a>	クライアントの 64 ビット・チャンネル出口の位置を指定します。	Yes	Yes	Yes	Yes	Yes
<a href="#">JavaExitsClassPath</a>	Java 出口の実行時に、クラスパスに追加される値。	No	Yes	Yes	No	No
<b>JMQI</b> スタンザ						
<a href="#">useMQCSPAuthentication</a>	IBM MQ classes for Java および IBM MQ classes for JMS のアプリケーションがキュー・マネージャーへの認証で互換モードと MQCSP 認証モードのどちらを使用するかを制御します。	No	Yes	Yes	No	No
<b>MessageBuffer</b> スタンザ						
<a href="#">MaximumSize</a>	1 から 999 999 の範囲の先読みバッファのサイズ (キロバイト単位)。	Yes	Yes	Yes	Yes	Yes
<a href="#">PurgeTime</a>	先読みバッファに残ったメッセージが除去される間隔 (秒単位)。	Yes	Yes	Yes	Yes	Yes

表 8. 各タイプのクライアントに適用される属性 (続き)

<b>mqclient.ini</b> スタンザ名および属性	説明	<b>C</b> および非管理対象 <b>.NET</b>	<b>Java</b>	<b>JMS</b>	管理対象 <b>.NET</b>	管理対象 <b>XMS .NET</b>
<u>UpdatePercentage</u>	クライアント・アプリケーションがサーバーに対して新しい要求をいつ行うかを決定するしきい値の計算に使用される、1 から 100 の範囲の更新のパーセント値。	Yes	Yes	Yes	Yes	Yes
<b>PreConnect</b> スタンザ						
<u>データ</u>	接続定義が保管されるリポジトリの URL。	Yes	No	No	No	No
<u>関数</u>	PreConnect 出口コードを含むライブラリーへの、関数のエントリー・ポイントの名前。	Yes	No	No	No	No
<u>Module</u>	API 出口コードを含むモジュールの名前。	Yes	No	No	No	No
<u>Sequence</u>	その他の出口に関してこの出口が呼び出される順序。	Yes	No	No	No	No
<b>セキュリティー・スタンザ</b>						
<u>DisableClientAMS</u>	キュー・マネージャーへのクライアント接続で AMS を無効/有効にします。	Yes	Yes	Yes	No	No
<b>SSL</b> スタンザ						

表 8. 各タイプのクライアントに適用される属性 (続き)

<b>mqclient.ini</b> スタンザ名および属性	説明	<b>C</b> および非管理対象 <b>.NET</b>	<b>Java</b>	<b>JMS</b>	管理対象 <b>.NET</b>	管理対象 <b>XMS .NET</b>
<a href="#">CDPCheckExtensions</a>	このキュー・マネージャーの SSL チャネルまたは TLS チャネルが、CrlDistributionPoint 証明書拡張で指定されている CDP サーバーの検査を試行するかどうかを指定します。	Yes	No	No	No	No
<a href="#">CertificateLabel</a>	チャネル定義の証明書ラベル。	Yes	No	No	No	No
<a href="#">CertificateValidationPolicy</a>	使用する証明書妥当性検査のタイプを決定します。	Yes	No	No	No	No
<a href="#">ClientRevocationChecks</a>	クライアント接続呼び出しで SSL/TLS チャネルを使用する場合の証明書の失効検査をどのように構成するかを指定します。	Yes	No	No	No	No
<a href="#">EncryptionPolicySuiteB</a>	チャネルで Suite-B 準拠の暗号化を使用するかどうか、および使用する際の強度のレベルを決定します。	Yes	No	No	No	No

表 8. 各タイプのクライアントに適用される属性 (続き)

<b>mqclient.ini</b> スタンザ名および属性	説明	<b>C</b> および非管理対象 <b>.NET</b>	<b>Java</b>	<b>JMS</b>	管理対象 <b>.NET</b>	管理対象 <b>XMS .NET</b>
<a href="#">OCSPAuthentication</a>	OCSP が有効な状態で、OCSP 取り消し検査で証明書取り消し状況を判別できない場合の、IBM MQ の動作を定義します。	Yes	No	No	No	No
<a href="#">OCSPCheckExtensions</a>	IBM MQ が AuthorityInfo Access 証明書拡張に反応するかどうかを制御します。	Yes	No	No	No	No
<a href="#">SSLCryptoHardware</a>	システム上に存在する PKCS #11 暗号ハードウェアの構成に必要なパラメーター・ストリングを設定します。	Yes	No	No	No	No
<a href="#">SSLFipsRequired</a>	IBM MQ で暗号化を実行する場合に FIPS 認証アルゴリズムのみを使用するかどうかを指定します。	Yes	No	No	No	No
<a href="#">SSLHTTPProxyName</a>	ストリングは、OCSP チェックのために GSKit で使用される HTTP プロキシ・サーバーのホスト名またはネットワーク・アドレスのいずれかです。	Yes	No	No	No	No

表 8. 各タイプのクライアントに適用される属性 (続き)

<b>mqclient.ini</b> スタンザ名および属性	説明	<b>C</b> および非管理対象 <b>.NET</b>	<b>Java</b>	<b>JMS</b>	管理対象 <b>.NET</b>	管理対象 <b>XMS .NET</b>
<a href="#">SSLKeyRepository</a>	ユーザーのデジタル証明書を保持するキー・リポジトリの位置 (語幹形式で指定)。	Yes	No	No	No	No
<a href="#">SSLKeyResetCount</a>	秘密鍵が再ネゴシエーションされる前に、SSL チャネルまたは TLS チャネルで送受信された暗号化されていないバイトの数。	Yes	No	No	No	No
<b>TCP</b> スタンザ						
<a href="#">ClntRcvBufferSize</a>	クライアント接続チャネルとサーバー接続チャネルのクライアント側で使用される TCP/IP 受信バッファのサイズ (バイト)。	Yes	Yes	Yes	Yes	Yes
<a href="#">ClntSndBufferSize</a>	クライアント接続チャネルとサーバー接続チャネルのクライアント側で使用される TCP/IP 送信バッファのサイズ (バイト)。	Yes	Yes	Yes	Yes	Yes
<a href="#">ConnectTimeout</a>	ソケットへの接続試行がタイムアウトになるまでの秒数。	Yes	Yes	Yes	No	No
<a href="#">IPAddressVersion</a>	チャネル接続に使用する IP プロトコルを指定します。	Yes	No	No	Yes	Yes

表 8. 各タイプのクライアントに適用される属性 (続き)

mqclient.ini スタンザ名および属性	説明	C および非管理対象 .NET	Java	JMS	管理対象 .NET	管理対象 XMS .NET
KeepAlive	KeepAlive 機能をオンまたはオフに切り替えます。	Yes	Yes	Yes	Yes	Yes
<b>Windows</b> Library1	Windows のみ。TCP/IP ソケット DLL の名前。	Yes	No	No	No	No

## クライアント構成ファイルの CHANNELS スタンザ

CHANNELS スタンザを使用して、クライアント・チャンネルに関する情報を指定します。

注：このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。すべての IBM MQ MQI client 構成ファイル・スタンザのサマリー表は、[各クライアントが読み取り可能な IBM MQ 属性を参照してください](#)。

次の属性を CHANNELS スタンザに含めることができます。

### CCSID = number

使用するコード化文字セット番号。

この属性は、C、非管理対象 .NET、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

CCSID 番号は、MQCCSID 環境パラメーターと同等です。

### ChannelDefinitionDirectory = path

クライアント・チャンネル定義テーブルを含むファイルのディレクトリー・パス。

この属性は、C、非管理対象 .NET、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

**Windows** Windows システムの場合、デフォルトは IBM MQ データおよびログ・ファイル・ディレクトリーで、通常は C: ¥ProgramData¥IBM¥MQ です。

**Linux** **UNIX** UNIX and Linux システムの場合、デフォルトは /var/mqm です。

**V 9.0.0** ChannelDefinitionDirectory には、ChannelDefinitionFile 属性との組み合わせで機能する URL を含めることができます (43 ページの『クライアント・チャンネル定義テーブルへの Web アドレス指定可能アクセス』を参照してください)。

ChannelDefinitionDirectory パスは、MQCHLLIB 環境パラメーターと同等です。

### ChannelDefinitionFile = filename|AMQCLCHL . TAB

クライアント・チャンネル定義テーブルを含むファイルの名前。

この属性は、C、非管理対象 .NET、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

クライアント・チャンネル定義テーブルは、MQCHLTAB 環境パラメーターと同等です。

### ReconDelay=(delay[, rand])(delay[, rand])...

ReconDelay 属性は、自動再接続が可能なクライアント・プログラムに対して再接続遅延を構成するための管理オプションを提供します。

この属性は、C、非管理対象 .NET、IBM MQ classes for JMS、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

構成の例は以下のとおりです。

```
ReconDelay=(1000,200)(2000,200)(4000,1000)
```

示されている例では、初回の遅延を 1 秒に 200 ミリ秒までのランダム間隔を加算した値として定義します。その次の遅延は 2 秒に 200 ミリ秒までのランダム間隔を加算した値です。その後の遅延はすべて、4 秒に 1000 ミリ秒までのランダム間隔を加算した値です。

### DefRecon = NO|YES|QMGR |DISABLED

DefRecon 属性は、クライアント・プログラムによる自動再接続を有効にするか、自動的に再接続するように作成されたクライアント・プログラムの自動再接続を無効にするための管理オプションを提供します。プログラムが再接続との互換性がない MQPMO\_LOGICAL\_ORDER のようなオプションを使用する場合は、後者の設定を選択できます。

この属性は、C、非管理対象 .NET、IBM MQ classes for JMS、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

IBM MQ classes for Java では、クライアントの自動再接続はサポートされていません。

DefRecon オプションの解釈は、クライアント・プログラムに MQCNO\_RECONNECT\_\* 値も設定されているか、および設定されている値が何かによって異なります。

クライアント・プログラムが、MQCONN を使用して接続するか、MQCONNX を使用して MQCNO\_RECONNECT\_AS\_DEF オプションを設定する場合、DefRecon によって設定された再接続値が有効になります。再接続値がプログラムまたは DefRecon オプションで設定されていなければ、クライアント・プログラムは自動的に再接続されません。

#### NO

**MQCONNX** によってオーバーライドされない限り、クライアントは自動的に再接続されません。

#### YES

**MQCONNX** によってオーバーライドされない限り、クライアントは自動的に再接続します。

#### QMGR

**MQCONNX** によってオーバーライドされない限り、クライアントは、同じキュー・マネージャーに対してのみ自動的に再接続します。QMGR オプションは MQCNO\_RECONNECT\_Q\_MGR と同じ効果があります。

#### DISABLED

**MQCONNX** MQI 呼び出しを使用してクライアント・プログラムによって要求された場合でも、再接続は無効になります。

自動クライアント再接続は、2 つの値によって決まります。

- アプリケーションの再接続オプション・セット
- mqclient.ini ファイルの DefRecon 値

表 9. 自動再接続はアプリケーションおよび mqclient.ini ファイルで設定された値によって異なる				
mqclient.ini の DefRecon 値	アプリケーションで設定される再接続オプション			
	MQCNO_RECONNECT	MQCNO_RECONNECT_Q_MGR	MQCNO_RECONNECT_AS_DEF	MQCNO_RECONNECT_DISABLED
NO	YES	QMGR	NO	NO
YES	YES	QMGR	YES	NO
QMGR	YES	QMGR	QMGR	NO
DISABLED	NO	NO	NO	NO

## MQReconnectTimeout

クライアントに再接続するタイムアウト (秒)。デフォルト値は 1800 秒 (30 分) です。

この属性は、C および非管理対象 .NET クライアント、および管理対象 .NET クライアントが読み取ることができます。

IBM MQ classes for JMS クライアントは、接続ファクトリー・プロパティー `CLIENTRECONNECTTIMEOUT` を使用して、再接続するタイムアウトを指定できます。このプロパティーのデフォルト値は 1800 秒 (30 分) です。

IBM MQ classes for XMS .NET クライアントは、以下のプロパティーを使用して、再接続するタイムアウトを指定できます。

- 接続ファクトリー・プロパティー `CLIENTRECONNECTTIMEOUT`。このプロパティーのデフォルト値は 1800 秒 (30 分) です。このプロパティーは、管理対象モードでのみ有効です。
- プロパティー `XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT`。このプロパティーのデフォルト値は 1800 秒 (30 分) です。このプロパティーは、管理対象モードでのみ有効です。

## ServerConnectionParms

`ServerConnectionParms` は `MQSERVER` 環境パラメーターと同等で、使用する IBM MQ サーバーおよび通信方式の位置を指定します。

この属性は、C、非管理対象 .NET、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

`ServerConnectionParms` 属性で定義するのは単純なチャンネルのみです。この属性を使用して、TLS チャンネルやチャンネル出口があるチャンネルを定義することはできません。これは `ChannelName/TransportType/ConnectionName` という形式のストリングで、`ConnectionName` は完全修飾ネットワーク名であることが必要です。`ChannelName` に、スラッシュ文字 (/) を使用することはできません。スラッシュ文字はチャンネル名、トランスポート・タイプ、および接続名を区切るために使用されるからです。

クライアント・チャンネルを定義するために `ServerConnectionParms` が使用されている場合は、最大メッセージ長として 100 MB が使用されます。そのため、チャンネルの最大メッセージ・サイズは、実際にはサーバーの `SVRCONN` チャンネルに指定した値となります。

クライアント・チャンネル接続は 1 つしか作成できないことに注意してください。例えば、次のような 2 つの項目があるとします。

```
ServerConnectionParms=R1.SVRCONN/TCP/localhost(1963)
ServerConnectionParms=R2.SVRCONN/TCP/localhost(1863)
```

この場合、2 番目だけが使用されます。

`ConnectionName` を、記述済みトランスポート・タイプ名のコンマ区切りのリストとして指定します。通常、必要な名前は 1 つのみです。複数の `hostnames` を指定して、同じプロパティーで複数の接続を構成することができます。接続は、正常に確立されるまで、接続リストに指定された順序で試行されます。正常に接続が行われなかった場合は、クライアントが再び処理を開始します。接続リストは、キュー・マネージャー・グループの代替として、再接続可能なクライアントの接続を構成するために使用されます。

## Put1DefaultAlwaysSync = NO| YES

`MQPMO_RESPONSE_AS_Q_DEF` オプションを使用して、`MQPUT1` 関数呼び出しの動作を制御します。

この属性は、C、非管理対象 .NET、IBM MQ classes for Java、および IBM MQ classes for JMS、管理対象 .NET、および管理対象 XMS .NET クライアントによって読み取ることができます。

### NO

`MQPUT1` が `MQPMO_SYNCPOINT` で設定される場合、`MQPMO_ASYNC_RESPONSE` として動作します。同様に、`MQPUT1` が `MQPMO_NO_SYNCPOINT` で設定される場合、`MQPMO_SYNC_RESPONSE` として動作します。これはデフォルト値です。

## YES

MQPUT1 は、MQPMO\_SYNCPOINT または MQPMO\_NO\_SYNCPOINT が設定されているかどうかに関係なく MQPMO\_SYNC\_RESPONSE が設定されている場合のように動作します。

### PasswordProtection = Compatible|always|optional

IBM MQ 8.0 より、TLS を使用する代わりに、保護されたパスワードを MQCSP 構造に設定できます。

この属性は、C、非管理対象 .NET、IBM MQ classes for Java、および IBM MQ classes for JMS、管理対象 .NET、および管理対象 XMS .NET クライアントによって読み取ることができます。

MQCSP パスワード保護は TLS 暗号化より設定が簡単なので、テストや開発目的に役立ちます。ただし、それほど安全ではありません。

詳しくは、[MQCSP パスワード保護](#)を参照してください。

## 関連情報

[キュー・マネージャーへの IBM MQ MQI アプリケーションの接続](#)

## クライアント構成ファイルの ClientExitPath スタンザ

ClientExitPath スタンザを使用して、クライアントのチャネル出口のデフォルト位置を指定します。

**注：**このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。すべての IBM MQ MQI client 構成ファイル・スタンザのサマリー表は、[各クライアントが読み取り可能な IBM MQ 属性](#)を参照してください。

次の属性を ClientExitPath スタンザに含めることができます。

### ExitsDefaultPath = string

クライアント用の 32 ビット・チャネル出口の場所を指定します。

この属性は、C、非管理対象 .NET、管理対象 .NET、管理対象 XMS .NET、IBM MQ classes for Java、および IBM MQ classes for JMS クライアントが読み取ることができます。IBM MQ classes for Java および IBM MQ classes for JMS クライアントは、この属性を使用して、Java で記述されていない 32 ビット・チャネル出口を見つけます。

### ExitsDefaultPath64 = string

クライアントに対する 64 ビット・チャネル出口の場所を指定します。

この属性は、C、非管理対象 .NET、管理対象 .NET、管理対象 XMS .NET、IBM MQ classes for Java、および IBM MQ classes for JMS クライアントが読み取ることができます。IBM MQ classes for Java および IBM MQ classes for JMS クライアントは、この属性を使用して、Java で記述されていない 64 ビット・チャネル出口を見つけます。

### JavaExitsClassPath = string

Java 出口の実行時に、クラスパスに追加される値。これは、他言語の出口では無視されます。

この属性は、IBM MQ classes for Java および IBM MQ classes for JMS クライアントが読み取ることができます。

JMS 構成ファイルでは、JavaExitsClass パス名に標準 com.ibm.mq.cfg が与えられます。接頭部およびこのフルネームは、IBM WebSphere MQ 7.0 以降のシステム・プロパティーでも使用されます。IBM WebSphere MQ 6.0 では、この属性は、IBM WebSphere MQ 6.0 の README ファイルに記載されたシステム・プロパティー com.ibm.mq.exitClasspath を使用して指定されました。

com.ibm.mq.exitClasspath の使用は推奨されません。JavaExitsClassPath と exitClasspath の両方が存在する場合は、JavaExitsClassPath が受け入れられます。exitClasspath の使用のみが存在する場合は、引き続き IBM WebSphere MQ 7.0 以降でも受け入れられます。

## クライアント構成ファイルの JMQUI スタンザ

JMQUI スタンザを使用して、IBM MQ classes for Java および IBM MQ classes for JMS が使用する Java メッセージ・キュー・インターフェース (JMQUI) の構成パラメーターを指定します。

注: このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。すべての IBM MQ MQI client 構成ファイル・スタンザのサマリー表は、[各クライアントが読み取り可能な IBM MQ 属性を参照してください](#)。

次の属性を JMQUI スタンザに含めることができます。

#### **useMQCSPauthentication = NO | YES**

IBM MQ classes for Java および IBM MQ classes for JMS のアプリケーションがキュー・マネージャーへの認証で互換モードと MQCSP 認証モードのどちらを使用するかを制御します。

この属性は、IBM MQ classes for Java および IBM MQ classes for JMS クライアントが読み取ることができます。

この属性には、以下の値を設定できます。

#### **NO**

キュー・マネージャーへの認証で互換モードを使用します。これはデフォルト値です。

#### **YES**

キュー・マネージャーへの認証で MQCSP 認証モードを使用します。

互換モードと MQCSP 認証モードの詳細については、[Java クライアントを使用した接続認証を参照してください](#)。

## **Windows クライアント構成ファイルの LU62、NETBIOS および SPX の各スタンザ**

Windows システム上でのみ、以下のスタンザを使用して、特定のネットワーク・プロトコル用の構成パラメーターを指定します。

### **LU62 スタンザ**

SNA LU 6.2 プロトコル構成パラメーターを指定する場合は、LU62 スタンザを使用します。次の属性をこのスタンザに含めることができます。

#### **Library1 = *DLLName*|WCPI32**

APPC DLL の名前。

#### **Library2 = *DLLName*|WCPI32**

Library1 と同様に、コードが 2 つの別個のライブラリーに格納されている場合に使用されます。

#### **TPName**

リモート・サイトで始動する TP 名。

### **NETBIOS スタンザ**

NetBIOS プロトコル構成パラメーターを指定する場合は、NETBIOS スタンザを使用します。次の属性をこのスタンザに含めることができます。

#### **AdapterNum = *number*|0**

LAN アダプターの番号。

#### **Library1 = *DLLName*|NETAPI32**

NetBIOS DLL の名前。

#### **LocalName = *name***

このコンピューターを LAN 上で認識する際に使用する名前。

これは MQNAME 環境パラメーターと同等です。

#### **NumCmds = *number*|1**

割り振るコマンドの数。

#### **NumSess = *number*|1**

割り振るセッションの数。

## SPX スタンザ

SPX プロトコル構成パラメーターを指定する場合は、SPX スタンザを使用します。次の属性をこのスタンザに含めることができます。

### BoardNum = *number*|0

LAN アダプターの番号。

### KeepAlive = YES|NO

KeepAlive 機能のオン/オフを切り替えます。

KeepAlive=YES にすると、SPX で、接続の相手側が引き続き有効であるかが定期的に確認されます。有効でない場合、チャンネルは閉じた状態です。

### Library1 = *DLLName*|WSOCK32.DLL

SPX DLL の名前。

### Library2 = *DLLName*|WSOCK32.DLL

Library1 と同様に、コードが 2 つの別個のライブラリーに格納されている場合に使用されます。

### Socket = *number*|5E86

16 進表記の SPX ソケット番号。

## クライアント構成ファイルの MessageBuffer スタンザ

MessageBuffer スタンザを使用して、メッセージ・バッファーに関する情報を指定します。

**注:** このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。すべての IBM MQ MQI client 構成ファイル・スタンザのサマリー表は、[各クライアントが読み取り可能な IBM MQ 属性を参照してください](#)。

次の属性を MessageBuffer スタンザに含めることができます。

### MaximumSize = *integer*|1

1 から 999 999 の範囲の先読みバッファーのサイズ (キロバイト単位)。

この属性は、C、非管理対象 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

以下の特殊値があります。

**-1**

クライアントが適切な値を決定します。

**0**

クライアントでは先読みは無効です。

### PurgeTime = *integer*|600

先読みバッファーに残ったメッセージが除去される間隔 (秒単位)。

この属性は、C、非管理対象 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

クライアント・アプリケーションが MsgId または CorrelId に基づいてメッセージを選択している場合、前に要求した MsgId または CorrelId でクライアントに送信されたメッセージが先読みバッファーに含まれている場合があります。その場合、こうしたメッセージは、適切な MsgId または CorrelId を付けて MQGET が発行されるまで先読みバッファー内に保留になります。先読みバッファーからのメッセージの除去は、PurgeTime を設定して行うことができます。除去間隔より長く先読みバッファーに残っているメッセージは、自動的に除去されます。こうしたメッセージはキュー・マネージャー上のキューから既に削除されているので、ブラウズされていない限り失われています。

有効な範囲は 1 から 999 999 秒、または特殊値 0 (除去が行われなことを意味します) です。

### UpdatePercentage = *integer*|-1

クライアント・アプリケーションがサーバーに対して新しい要求をいつ行うかを決定するしきい値の計算に使用される、1 から 100 の範囲の更新のパーセント値。特殊値 -1 は、クライアントが適切な値を決定することを示します。

この属性は、C、非管理対象 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

クライアントは要求を定期的にサーバーに送信して、クライアント・アプリケーションが消費したデータの量を示します。MQGET 呼び出しによってクライアントが取得したバイト数  $n$  がしきい値  $T$  を超えると、要求が送信されます。 $n$  は、新しい要求がサーバーに送信されるたびにゼロにリセットされます。

しきい値  $T$  は次のように計算されます。

$$T = \text{Upper} - \text{Lower}$$

上位値は先読みバッファのサイズと同じであり、*MaximumSize* 属性によってキロバイト単位で指定されます。デフォルトは 100 キロバイトです。

下位値は、上位値より低く、*UpdatePercentage* 属性によって指定されます。この属性は、1 から 100 の範囲の数値であり、デフォルトは 20 です。下位値は、次のように計算されます。

$$\text{Lower} = \text{Upper} \times \text{UpdatePercentage} / 100$$

#### 例 1:

*MaximumSize* 属性と *UpdatePercentage* 属性は、デフォルトとして、100 キロバイトと 20 キロバイトを取ります。

クライアントは、MQGET を呼び出してメッセージを取得して、これを繰り返します。これは、MQGET が  $n$  バイトを消費するまで続きます。

次の計算を使用します。

$$T = \text{Upper} - \text{Lower}$$

$T$  は  $(100 - 20) = 80$  キロバイトとなります。

MQGET 呼び出しがキューから 80 キロバイトを除去した場合、クライアントは自動的に新しい要求を行います。

#### 例 2:

*MaximumSize* 属性のデフォルトは 100 キロバイトであり、*UpdatePercentage* の値は 40 です。

クライアントは、MQGET を呼び出してメッセージを取得して、これを繰り返します。これは、MQGET が  $n$  バイトを消費するまで続きます。

次の計算を使用します。

$$T = \text{Upper} - \text{Lower}$$

$T$  は  $(100 - 40) = 60$  キロバイトとなります。

MQGET 呼び出しがキューから 60 キロバイトを除去した場合、クライアントは自動的に新しい要求を行います。これは、デフォルトを使用する例 1 よりもすぐに生じます。

そのため、しきい値  $T$  に大きな値を選択した場合は、クライアントからサーバーに要求が送信される頻度が減少する傾向にあります。逆に、しきい値  $T$  に小さな値を選択した場合は、クライアントからサーバーに要求が送信される頻度が増加する傾向にあります。

ただし、しきい値  $T$  に大きな値を選択するということは、先読みバッファが空になる機会が増える可能性があるため、先読みによるパフォーマンス向上の効果が小さくなるという意味にもなります。これが生じると、MQGET 呼び出しは、サーバーからデータが届くのを待つために休止しなければならない場合があります。

## クライアント構成ファイルの PreConnect スタンザ

PreConnect スタンザを使用して、mqclient.ini ファイルで PreConnect 出口を構成します。

**注:** このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。すべての IBM MQ MQI client 構成ファイル・スタンザのサマリー表は、[各クライアントが読み取り可能な IBM MQ 属性を参照してください](#)。

次の属性を PreConnect スタンザに含めることができます。

#### **Data = user\_data**

この属性では、接続前出口に渡すユーザー・データを指定します。接続前出口に渡すデータは、その接続前出口の実装と、渡すデータに合わせた固有のデータになります。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

例えば、LDAP サーバーを使用する場合などでは、この属性を使用して、接続定義の保管場所であるリポジトリの URL を指定できます。

```
Data = ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
```

#### **Function = myFunc**

PreConnect 出口コードを含むライブラリーへの、関数のエントリー・ポイントの名前。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

関数定義は、PreConnect 出口のプロトタイプ `MQ_PRECONNECT_EXIT` に従います。

このフィールドの最大長は `MQ_EXIT_NAME_LENGTH` です。

#### **Module = myMod**

API 出口コードを含むモジュールの名前。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

このフィールドにモジュールの絶対パス名が指定されていると、そのまま使用されます。

#### **Sequence = sequence\_number**

その他の出口に関してこの出口が呼び出される順序。小さなシーケンス番号の出口は、より大きなシーケンス番号の出口よりも先に呼び出されます。出口のシーケンス番号は連続である必要はありません。つまり、1、2、3 の順序は、7、42、1096 の順序と同じ結果となります。この属性は、符号なし数値です。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

`mqclient.ini` ファイル内で複数の PreConnect スタンザを定義できます。各出口の処理順序は、スタンザの Sequence 属性によって決定されます。

#### **関連情報**

[リポジトリから接続前出口を使用した接続定義の参照](#)

## **クライアント構成ファイルの Security スタンザ**

Security スタンザを使用して、キュー・マネージャーへのクライアント接続で AMS を無効/有効にします。

**注:** このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。すべての IBM MQ MQI client 構成ファイル・スタンザのサマリー表は、[各クライアントが読み取り可能な IBM MQ 属性を参照してください](#)。

セキュリティ・スタンザには、以下の属性を含めることができます:

#### **DisableClientAMS = NO|YES**

DisableClientAMS 属性で、以前のバージョンの製品からキュー・マネージャーに接続するために IBM WebSphere MQ 7.5 またはそれ以降のクライアントを使用していて、IBM MQ Advanced Message Security(AMS)を使用不可にすることができます。また、2085 (MQRC\_UNKNOWN\_OBJECT\_NAME) エラーが報告されます。

IBM WebSphere MQ 7.5 以降、IBM MQ クライアントでは IBM MQ Advanced Message Security (AMS) が自動的に有効になっているため、クライアントはデフォルトで、キュー・マネージャーのオブジェクトに関するセキュリティ・ポリシーをチェックします。ただし、それより前のバージョンの製品の

サーバー (IBM WebSphere MQ 7.1 など) では、AMS が有効になっていないため、これによって 2085 (MQRC\_UNKNOWN\_OBJECT\_NAME) エラーが報告されます。

DisableClientAMS 属性の使用例を以下に示します。

- AMS を無効にするには、次のようにします。

```
Security:  
DisableClientAMS=Yes
```

- AMS を有効にするには、次のようにします。

```
Security:  
DisableClientAMS=No
```

この属性は、C、IBM MQ classes for Java、および IBM MQ classes for JMS クライアントが読み取ることができます。

## 関連情報

[クライアントでの Advanced Message Security の無効化](#)

## クライアント構成ファイルの SSL スタンザ

SSL スタンザを使用して、TLS の使用に関する情報を指定します。

注：このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。すべての IBM MQ MQI client 構成ファイル・スタンザのサマリー表は、[各クライアントが読み取り可能な IBM MQ 属性を参照してください](#)。

次の属性を SSL スタンザに含めることができます。

### CDPCheckExtensions=YES|NO

CDPCheckExtensions は、このキュー・マネージャーの TLS チャンネルまたは TLS チャンネルが、CrldistributionPoint 証明書拡張で指定されている CDP サーバーの検査を試行するかどうかを指定します。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

この属性に指定できる値は以下のとおりです。

- YES: TLS チャンネルまたは TLS チャンネルは、CDP サーバーを検査して、デジタル証明書が取り消されているかどうかを判別しようとします。
- NO: TLS チャンネルまたは TLS チャンネルは、CDP サーバーの検査を試行しません。この値はデフォルトです。

### CertificateLabel = string

チャンネル定義の証明書ラベル。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

詳しくは、[証明書ラベル \(CERTLABL\)](#) を参照してください。

### CertificateValPolicy = string

使用する証明書妥当性検査のタイプを決定します。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

この属性に指定できる値は以下のとおりです。

#### ANY

基礎となるセキュア・ソケット・ライブラリーによってサポートされているいずれかの証明書妥当性検査ポリシーを使用します。この設定はデフォルト設定です。

#### RFC5280

RFC 5280 標準に準拠する証明書妥当性検査のみを使用します。

## ClientRevocationChecks = REQUIRED|OPTIONAL|DISABLED

クライアント接続呼び出しで TLS チャンネルを使用する場合の証明書の失効検査をどのように構成するかを指定します。 [OCSPAuthentication](#) も参照してください。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

この属性に指定できる値は以下のとおりです。

### REQUIRED (デフォルト)

CCDT から証明書失効構成をロードし、構成に従って失効検査を実行することを試みます。 CCDT ファイルを開くことができない場合、または証明書を検証できない場合 (例えば、OCSP または CRL サーバーを使用できないため) には、MQCONN 呼び出しは失敗します。 CCDT に失効構成が含まれない場合には、失効検査が実行されませんが、これによりチャンネルが失敗することはありません。

**Windows** Windows システムでは、CRL 失効検査に Active Directory を使用することもできます。 OCSP 失効検査に Active Directory を使用することはできません。

### オプション

REQUIRED と同様ですが、証明書失効構成をロードできない場合にはチャンネルは失敗しません。

### 無効化

CCDT から証明書失効構成をロードしようとせず、証明書の失効検査を実行しません。

注: MQCONN 呼び出しではなく、MQCONNX 呼び出しを使用する場合は、MQSCO により認証情報レコード (MQAIR) を設定することもできます。 そのため、CCDT ファイルを開くことができない場合の MQCONNX のデフォルト動作は、失敗することではなく、MQAIR が設定されていると見なすことです (設定しないことにした場合も含む)。

## EncryptionPolicySuiteB = string

チャンネルで Suite-B 準拠の暗号化を使用するかどうか、および使用する際の強度のレベルを決定します。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

この属性に指定できる値は以下のとおりです。

### NONE

Suite-B 準拠の暗号化を使用しない。 この設定はデフォルト設定です。

### 128\_BIT,192\_BIT

セキュリティー強度を 128 ビットと 192 ビットの両方のレベルに設定する。

### 128\_BIT

セキュリティー強度を 128 ビット・レベルに設定する。

### 192\_BIT

セキュリティー強度を 192 ビット・レベルに設定する。

## OCSPAuthentication = OPTIONAL|REQUIRED|WARN

OCSP が有効な状態で、OCSP 取り消し検査で証明書取り消し状況を判別できない場合の、IBM MQ の動作を定義します。 [ClientRevocationChecks](#) も参照してください。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

この属性に指定できる値は以下のとおりです。

### オプション

OCSP 検査で取り消し状況を判別できない証明書が受け入れられ、警告メッセージやエラー・メッセージは生成されません。 取り消し検査が行われなかった場合と同じように、SSL または TLS 接続が続行されます。

### REQUIRED

OCSP 検査で、検査されたすべての SSL または TLS 証明書に対して最終的な取り消し結果を示す必要があります。 取り消し状況が確認できない SSL または TLS 証明書はすべて拒否され、エラー・メッセージが表示されます。 キュー・マネージャーの SSL イベント・メッセージが有効な場合は、ReasonQualifier が MQRQ\_SSL\_HANDSHAKE\_ERROR である MQRQ\_CHANNEL\_SSL\_ERROR メッセージが生成されます。 接続はクローズされます。

この値がデフォルト値です。

#### **WARN**

OCSP 取り消し検査で任意の SSL または TLS 証明書の取り消し状況を判別できない場合に、キュー・マネージャーのエラー・ログに警告が報告されます。キュー・マネージャーの SSL イベント・メッセージが有効な場合は、ReasonQualifier が MQRQ\_SSL\_UNKNOWN\_REVOCATION である MQRQ\_CHANNEL\_SSL\_WARNING メッセージが生成されます。接続は続行可能です。

#### **OCSPCheckExtensions=YES|NO**

IBM MQ が AuthorityInfoAccess 証明書拡張に反応するかどうかを制御します。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

値を NO に設定すると、IBM MQ は AuthorityInfoAccess 証明書拡張を無視し、OCSP セキュリティ検査を試行しません。デフォルト値は YES です。

#### **SSLCryptoHardware = string**

システム上に存在する PKCS #11 暗号ハードウェアの構成に必要なパラメーター・ストリングを設定します。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

GSK\_PKCS11 = *driver path and filename;token label;token password;symmetric cipher setting*; の形式でストリングを指定します。

例: GSK\_PKCS11=/usr/lib/pkcs11/  
PKCS11\_API.so;tokenlabel;passw0rd;SYMMETRIC\_CIPHER\_ON

ドライバー・パスは、PKCS #11 カードのサポートを提供する共有ライブラリーへの絶対パスです。ドライバーのファイル名は共有ライブラリーの名前です。PKCS #11 ドライバーのパスおよびファイル名に必要な値は、一例として /usr/lib/pkcs11/PKCS11\_API.so のようにします。GSKit を介して対称暗号操作にアクセスするには、対称暗号設定パラメーターを指定します。このパラメーターの値は次のいずれかです。

#### **SYMMETRIC\_CIPHER\_OFF**

対称暗号操作を使用しません。この設定はデフォルト設定です。

#### **SYMMETRIC\_CIPHER\_ON**

対称暗号操作を使用します。

ストリングの最大長は 256 文字です。デフォルト値はブランクです。形式が正しくないストリングを指定すると、エラーが発生します。

 ストリングのさまざまな構成要素を指定する場合、円記号 (&#xa5;) 文字を使用してセミコロン文字をエスケープする必要があります。セミコロン文字はコメントとして扱われるためです。例: '\;'

#### **SSLFipsRequired = YES|NO**

IBM MQ で暗号化を実行する場合に FIPS 認証アルゴリズムのみを使用するかどうかを指定します。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

暗号ハードウェアが構成されている場合、ハードウェア製品で提供される暗号モジュールが使用されます。それらは、使用されているハードウェア製品によって一定レベルまで FIPS の認定を受けている場合もあれば、そうではない場合もあります。

#### **SSLHTTPProxyName = string**

ストリングは、OCSP チェックのために GSKit で使用される HTTP プロキシ・サーバーのホスト名またはネットワーク・アドレスのいずれかです。このアドレスの後に、オプションのポート番号が括弧で囲まれて続く場合もあります。ポート番号を指定しないと、デフォルトの HTTP ポートである 80 が使用されます。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

HP-UX PA-RISC と Sun Solaris SPARC プラットフォーム、および AIX の 32 ビット・クライアントの場合、ネットワーク・アドレスには IPv4 アドレスのみ使用できます。その他のプラットフォームの場合は、IPv4 アドレスまたは IPv6 アドレスを使用できます。

例えば、ファイアウォールが OCSP 応答側の URL へのアクセスを回避する場合など、この属性が必要になる場合があります。

#### **SSLKeyRepository = *pathname***

ユーザーのデジタル証明書を保持するキー・リポジトリの位置 (語幹形式で指定)。つまり、これには絶対パスおよび拡張子なしのファイル名が含まれています。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

#### **SSLKeyResetCount = *integer*|0**

秘密鍵が再ネゴシエーションされる前に、TLS チャンネルまたは TLS チャンネルで送受信された暗号化されていないバイトの数。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

値は 0 から 999999999 の範囲でなければなりません。

デフォルトは 0 で、これは秘密鍵が絶対に再ネゴシエーションされないことを意味します。

1 から 32768 の値を指定すると、TLS チャンネルと TLS チャンネルは、秘密鍵のリセット・カウントとして 32768 (32 キロバイト) を使用します。これは、秘密鍵のリセット値が小さい場合に生じる可能性のある、鍵の過度のリセットを防ぐためです。

## **クライアント構成ファイルの TCP スタンザ**

TCP スタンザを使用して、TCP ネットワーク・プロトコル構成パラメーターを指定します。

注: このスタンザの各属性に関する記述は、どの IBM MQ クライアントがその属性を読み取ることができるのかを示しています。すべての IBM MQ MQI client 構成ファイル・スタンザのサマリー表は、[各クライアントが読み取り可能な IBM MQ 属性を参照してください](#)。

次の属性を TCP スタンザに含めることができます。

#### **ClntRcvBuffSize = *number*|0**

クライアント接続チャンネルとサーバー接続チャンネルのクライアント側で使用される TCP/IP 受信バッファのサイズ (バイト)。

この属性は、C、非管理対象 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

値 0 は、IBM MQ によってバッファ・サイズが修正されるのではなく、オペレーティング・システムがバッファ・サイズを管理することを意味します。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

#### **ClntSndBuffSize = *number*|0**

クライアント接続チャンネルとサーバー接続チャンネルのクライアント側で使用される TCP/IP 送信バッファのサイズ (バイト)。

この属性は、C、非管理対象 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

値 0 は、IBM MQ によってバッファ・サイズが修正されるのではなく、オペレーティング・システムがバッファ・サイズを管理することを意味します。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

#### **Connect\_Timeout = *number***

ソケットへの接続試行がタイムアウトになるまでの秒数。デフォルト値である 0 は、接続のタイムアウトがないことを指定します。

この属性は、C、非管理対象 .NET、IBM MQ classes for Java、および IBM MQ classes for JMS クライアントが読み取ることができます。

IBM MQ チャネル・プロセスは非ブロッキング・ソケットを介して接続します。このため、ソケットの  
もう一方の側の準備ができていない場合、connect() は即時に *EINPROGRESS* または *EWOULDBLOCK* を  
返します。これに続いて接続が最大 20 回まで再試行された後、通信エラーがレポートされます。

Connect\_Timeout がゼロ以外の値に設定されている場合、IBM MQ は、ソケットの準備ができるまで、  
select() 呼び出しに対して規定された期間待機します。これにより、後続の connect() 呼び出しが成功  
する確率は上昇します。このオプションは、ネットワークの負荷が高くなっていて、接続に多少の待  
機時間が必要なときに役に立つ場合があります。

Connect\_Timeout、ClntSndBuffSize、および ClntRcvBuffSize パラメーターの間には関係はありません。

#### **IPAddressVersion = MQIPADDR\_IPV4|MQIPADDR\_IPV6**

チャネル接続に使用する IP プロトコルを指定します。

この属性は、C、非管理対象 .NET、管理対象 .NET、および管理対象 XMS .NET クライアントが読み取  
ることができます。

指定可能なストリング値は MQIPADDR\_IPV4 または MQIPADDR\_IPV6 です。これらの値は、**ALTER  
QMGR IPADDRV** の IPV4 および IPV6 と同じ意味です。

#### **KeepAlive = YES|NO**

KeepAlive 機能のオン/オフを切り替えます。KeepAlive=YES を指定すると、TCP/IP は、接続の相手側  
がまだ使用可能かどうかを定期的に検査します。有効でない場合、チャネルは閉じた状態です。

この属性は、C、非管理対象 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、管理対  
象 .NET、および管理対象 XMS .NET クライアントが読み取ることができます。

#### **Windows Library1 = DLLName|WSOCK32**

(Windows のみ) TCP/IP ソケット DLL の名前。

この属性は、C および非管理対象 .NET クライアントが読み取ることができます。

## **IBM MQ 環境変数の使用**

コマンドを使用して、現在の設定を表示したり、IBM MQ 環境変数の値をリセットしたりすることができま  
す。

### **このタスクについて**

以下の方法で環境変数を使用できます。

- システム・プロファイル内の変数を設定して永続的な変更を行う
- コマンド・ラインからコマンドを発行し、このセッションのみの変更を行う
- 実行中のアプリケーションに応じて 1 つ以上の変数に特定の値を指定するために、そのアプリケーショ  
ンが使用するコマンド・スクリプト・ファイルにコマンドを追加する

それぞれの環境変数について、コマンドを使用して、現在の設定値を表示することや変数の値をリセット  
することができます。これらのコマンドは、特に指定のない限り、すべての IBM MQ MQI client ・プラッ  
トフォームで使用できます。

ユーザーが設定しなかった変数については、IBM MQ はデフォルト値を使用します。

**注:** **z/OS** IBM MQ for z/OS は、いずれの IBM MQ 環境変数もサポートしていません。このプラッ  
トフォームをサーバーとして使用している場合は、クライアント・チャネル定義テーブルを z/OS 上で生成  
する方法について、[クライアント・チャネル定義テーブルを参照してください](#)。ただし、ご使用のクライ  
アント・プラットフォームでは IBM MQ 環境変数を使用できます。

### **手順**

- **Windows**  
Windows では、環境変数ごとに次のコマンドを使用して、現在の設定を表示するか、変数の値をリセッ  
トします。

- 環境変数の値を削除するには、コマンド SET MQSERVER= を使用します。
- 環境変数の現在の設定を表示するには、コマンド SET MQSERVER を使用します。
- セッションのすべての環境変数を表示するには、コマンド set を使用します。

Linux UNIX

UNIX and Linux では、環境変数ごとに次のコマンドを使用して、現在の設定を表示するか、変数の値をリセットします。

- 環境変数の値を削除するには、コマンド unset MQSERVER を使用します。
- 環境変数の現在の設定を表示するには、コマンド echo \$MQSERVER を使用します。
- セッションのすべての環境変数を表示するには、コマンド set を使用します。

## 関連タスク

52 ページの『[構成ファイルを使用したクライアントの構成](#)』

テキスト・ファイル内の属性を使用して、クライアントを構成します。この属性は、環境変数、または他のプラットフォーム固有の方法を使用してオーバーライドすることができます。

## 関連情報

環境変数

## V 9.0.0 MQCCDTURL

MQCCDTURL は、MQCHLLIB と MQCHLTAB 環境変数の組み合わせを設定する機能と同等の機能を提供します。



**重要:** 環境変数オプションを使用できるのは、クライアントとして接続するネイティブ・プログラム、つまり C、COBOL、または C++ のアプリケーションに対してのみです。Java、JMS、および管理対象 .NET アプリケーションの場合、環境変数は効果がありません。

IBM MQ では、ファイル、FTP、または HTTP の URL からの CCDT の取得がサポートされます。

ただし、MQCCDTURL は URL 値のみ受け入れます。MQCCDTURL は既存のローカル・ファイル・システムのディレクトリー形式は受け入れません。

MQCHLLIB および MQCHLTAB の代わりに MQCCDTURL を使用してローカル・ファイルを使用する場合は、'file:/' プロトコルを使用できます。したがって、

```
export MQCCDTURL=file:///var/mqm/qmgrs/QMGR/@ipcc/MYCHL.TAB
```

これは、下記と同じです。

```
export MQCHLLIB=/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLTAB=MYCHL.TAB
```

## JSON ファイルを使用する Windows の例

```
set MQCCDTURL=file:/c:/mq-channels/CCDT-QMGR1.json
```

これは、下記と同じです。

```
set MQCHLLIB=C:\mq-channels
set MQCHLTAB=CCDT-QMGR1.json
```

## 関連概念

43 ページの『[クライアント・チャンネル定義テーブルへの Web アドレス指定可能アクセス](#)』

IBM MQ 9.0 以降、この製品には、MQCNO を使用してプログラミングするか、環境変数を使用するか、または mqclient.ini ファイル・スタanzas を使用することにより、URL を介してクライアント・チャンネル定義テーブル (CCDT) を見つける機能が用意されています。

40 ページの『[クライアント・チャンネル定義テーブル](#)』

クライアント・チャンネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャンネル定義と認証情報を判別します。Multiplatforms の場合、CCDT は自動的に作成されます。それを、クライアント・アプリケーションに対して使用可能にする必要があります。

#### 関連資料

77 ページの『MQCHLLIB』

MQCHLLIB は、クライアント・チャンネル定義テーブル (CCDT) を含むファイルへのディレクトリー・パスを指定します。ファイルはサーバー上に作成されますが、IBM MQ MQI client・ワークステーションにコピーすることができます。

79 ページの『MQCHLTAB』

MQCHLTAB は、クライアント・チャンネル定義テーブル (CCDT) を含むファイルの名前を指定します。デフォルト・ファイル名は AMQCLCHL.TAB です。

#### 関連情報

[CCDTURL](#)

[XMSC WMQ CCDTURL](#)

[キュー・マネージャーへの IBM MQ MQI アプリケーションの接続](#)

## MQCCSID

MQCCSID は、使用するコード化文字セット番号を指定し、サーバーの構成に使用された CCSID 値をオーバーライドします。

詳しくは、[クライアントまたはサーバーのコード化文字セット ID \(CCSID\) の選択](#)を参照してください。

この変数を設定するには、以下のいずれかのコマンドを使用します。

- **Windows** Windows の場合:

```
SET MQCCSID=number
```

- **Linux** **UNIX** UNIX and Linux の場合:

```
export MQCCSID=number
```

- **IBM i** IBM i の場合:

```
ADDENVVAR ENVVAR(MQCCSID) VALUE(number)
```

## MQCERTLABL

MQCERTLABL はチャンネル定義の証明書ラベルを指定します。

詳しくは、[証明書ラベル \(CERTLABL\)](#) を参照してください。

## MQCERTVPOL

MQCERTVPOL は、使用する証明書妥当性検査ポリシーを指定します。

IBM MQ での証明書妥当性検査ポリシーについて詳しくは、[IBM MQ における証明書妥当性検査ポリシー](#)を参照してください。

この環境変数は、クライアント ini ファイルの SSL スタンザにある *CertificateValPolicy* 設定をオーバーライドします。変数は、次のいずれかの値に設定できます。

#### ANY

基礎となるセキュア・ソケット・ライブラリーによってサポートされているいずれかの証明書妥当性検査ポリシーを使用します。

## RFC5280

RFC 5280 標準に準拠する証明書妥当性検査のみを使用します。

この変数を設定するには、次のコマンドのうちの1つを使用します。

- ▶ **Windows** Windows の場合:

```
SET MQCERTVPOL= value
```

- ▶ **Linux** ▶ **UNIX** UNIX and Linux システムの場合:

```
export MQCERTVPOL= value
```

- ▶ **IBM i** IBM i の場合:

```
ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
```

## MQCHLLIB

MQCHLLIB は、クライアント・チャネル定義テーブル (CCDT) を含むファイルへのディレクトリー・パスを指定します。ファイルはサーバー上に作成されますが、IBM MQ MQI client・ワークステーションにコピーすることができます。

MQCHLLIB を設定しない場合、クライアントのパスは次のようにデフォルト指定されます。

- ▶ **Windows** Windows の場合: `MQ_INSTALLATION_PATH`
- ▶ **Linux** ▶ **UNIX** UNIX and Linux の場合: `/var/mqm/`
- ▶ **IBM i** IBM i の場合、`/QIBM/UserData/mqm/`

`crtmqm` コマンドおよび `strmqm` コマンドの場合、パスはデフォルトで以下の2つのパス・セットのいずれかになります。`datapath` が設定される場合、パスはデフォルトで1つ目のセットのいずれかになります。`datapath` が設定されない場合、パスはデフォルトで2つ目のセットのいずれかになります。

- ▶ **Windows** Windows の場合: `datapath\@ipcc`
- ▶ **Linux** ▶ **UNIX** UNIX and Linux の場合: `datapath/@ipcc`
- ▶ **IBM i** IBM i の場合: `datapath/&ipcc`

または:

- ▶ **Windows** Windows の場合: `MQ_INSTALLATION_PATH\data\qmgrs\qmgrname\@ipcc`
- ▶ **Linux** ▶ **UNIX** UNIX and Linux の場合: `/prefix/qmgrs/qmgrname/@ipcc`
- ▶ **IBM i** IBM i の場合: `/prefix/qmgrs/qmgrname/&ipcc`

ここで、

- `MQ_INSTALLATION_PATH` は、IBM MQ がインストールされている上位ディレクトリーを表します。
- `datapath` は、キュー・マネージャー・スタンザで定義される `DataPath` の値です (存在する場合)。
- `prefix` は、キュー・マネージャー・スタンザで定義されるプレフィックスの値です。プレフィックスは、通常は以下のいずれかの値になります。

- ▶ **Linux** ▶ **UNIX** UNIX and Linux システムでは `/var/mqm`。
- ▶ **IBM i** IBM i では `/QIBM/UserData/mqm/`。

- `qmgrname` は、キュー・マネージャー・スタンザで定義される `Directory` 属性の値です。この値は、実際のキュー・マネージャー名とは異なる場合があります。この値は、特殊文字の置換により変更されている可能性があります。
- キュー・マネージャー・スタンザが定義されている場所は、プラットフォームによって異なります。

-    `mq5.ini` ファイル (IBM i、UNIX、および Linux の場合)。
-  レジストリー (Windows の場合)。

注:

1.  IBM MQ for z/OS をサーバーとして使用する場合、このファイルは IBM MQ クライアント・ワークステーション上に保持する必要があります。
2. `MQCHLLIB` を設定すると、`CCDT` の配置に使用されるパスをオーバーライドします。
3.  `MQCHLLIB` には、`MQCHLTAB` 環境変数との組み合わせで機能する URL を含めることができます (43 ページの『クライアント・チャネル定義テーブルへの Web アドレス指定可能アクセス』を参照してください)。
4. `MQCHLLIB` などの環境変数は、プラットフォーム固有の方法で、単一のプロセスまたはジョブをスコープにするか、あるいはシステム全体をスコープにすることができます。
5. サーバーで `MQCHLLIB` をシステム全体に設定すると、そのサーバー上のすべてのキュー・マネージャーの `CCDT` ファイルに同じパスが設定されます。`MQCHLLIB` 環境変数を設定しないと、このパスはキュー・マネージャーごとに違うものになります。キュー・マネージャーは、`crtmqm` コマンドまたは `strmqm` コマンドで `MQCHLLIB` が設定されていれば、その値を読み取ります。
6. 単一のサーバーに複数のキュー・マネージャーを作成する場合には、以下の理由でそれらの区別が重要となります。`MQCHLLIB` をシステム全体に設定すると、各キュー・マネージャーが同じ `CCDT` ファイルを更新します。ファイルには、そのサーバー上にあるすべてのキュー・マネージャーからのクライアント接続定義が含まれます。複数のキュー・マネージャーに同じ定義 (例えば、`SYSTEM.DEF.CLNTCONN`) が存在する場合、ファイルには最新の定義が含まれます。キュー・マネージャーの作成時に `MQCHLLIB` を設定すると、`CCDT` で `SYSTEM.DEF.CLNTCONN` が更新されます。この更新により、別のキュー・マネージャーによって作成された `SYSTEM.DEF.CLNTCONN` が上書きされます。以前の定義を変更していた場合、その変更は失われます。そのため、サーバーで `MQCHLLIB` をシステム全体の環境変数に設定する代わりに、他の方法を探すことを検討する必要があります。
7. クライアント接続定義の `MQSC` および `PCF NOREPLACE` オプションは、`CCDT` ファイルの内容を検査しません。以前に別のキュー・マネージャーによって作成された同じ名前のクライアント接続チャンネル定義は、`NOREPLACE` オプションの指定に関係なく置換されます。以前に同じキュー・マネージャーによって作成された定義の場合は、置換されません。
8. コマンド `rcrmqobj -t clchltab` は、`CCDT` ファイルを削除し、再作成します。このファイルは、そのコマンドの実行対象となるキュー・マネージャーで作成されたクライアント接続定義のみを使用して再作成されます。
9. `CCDT` を更新する他のコマンドは、同じチャンネル名を持つクライアント接続チャンネルのみを変更します。そのファイル内の他のクライアント接続チャンネルは変更されません。
10. `MQCHLLIB` のパスには、引用符は必要ありません。

例

この変数を設定するには、次のコマンドのうちの 1 つを使用します。

-  Windows の場合:

```
SET MQCHLLIB=pathname
```

以下に例を示します。

```
SET MQCHLLIB=C:\wmqtest
```

- Linux UNIX UNIX and Linux システムの場合:

```
export MQCHLLIB=pathname
```

- IBM i IBM i の場合:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE(pathname)
```

## 関連資料

79 ページの『MQCHLTAB』

MQCHLTAB は、クライアント・チャンネル定義テーブル (CCDT) を含むファイルの名前を指定します。デフォルト・ファイル名は AMQCLCHL.TAB です。

## 関連情報

[キュー・マネージャーへの IBM MQ MQI アプリケーションの接続](#)

## MQCHLTAB

MQCHLTAB は、クライアント・チャンネル定義テーブル (CCDT) を含むファイルの名前を指定します。デフォルト・ファイル名は AMQCLCHL.TAB です。

クライアント・チャンネル定義テーブルがサーバー上のどこにあるかについては、[40 ページの『クライアント・チャンネル定義テーブル』](#)を参照してください。

この変数を設定するには、次のコマンドのうちの 1 つを使用します。

- Windows Windows の場合:

```
SET MQCHLTAB=filename
```

- Linux UNIX UNIX and Linux の場合:

```
export MQCHLTAB=filename
```

- IBM i IBM i の場合:

```
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(filename)
```

以下に例を示します。

```
SET MQCHLTAB=ccdf1.tab
```

クライアントに対するのと同様に、サーバー上の MQCHLTAB 環境変数で、クライアント・チャンネル定義テーブル名を指定します。

## 関連概念

43 ページの『クライアント・チャンネル定義テーブルへの Web アドレス指定可能アクセス』

IBM MQ 9.0 以降、この製品には、MQCNO を使用してプログラミングするか、環境変数を使用するか、または mqclient.ini ファイル・スタンザを使用することにより、URL を介してクライアント・チャンネル定義テーブル (CCDT) を見つける機能が用意されています。

## 関連資料

77 ページの『MQCHLLIB』

MQCHLLIB は、クライアント・チャンネル定義テーブル (CCDT) を含むファイルへのディレクトリー・パスを指定します。ファイルはサーバー上に作成されますが、IBM MQ MQI client・ワークステーションにコピーすることができます。

75 ページの『MQCCDTURL』

MQCCDTURL は、MQCHLLIB と MQCHLTAB 環境変数の組み合わせを設定する機能と同等の機能を提供します。

## 関連情報

[キュー・マネージャーへの IBM MQ MQI アプリケーションの接続](#)

## MQIPADDRV

MQIPADDRV は、チャンネル接続に使用すべき IP プロトコルを指定します。指定できるストリング値は、「MQIPADDR\_IPv4」または「MQIPADDR\_IPv6」です。これらの値は、ALTER QMGR IPADDRV の IPv4 および IPv6 と同じ意味です。設定されていない場合は、「MQIPADDR\_IPv4」が想定されます。

この変数を設定するには、次のコマンドのうちの 1 つを使用します。

- ▶ **Windows** Windows の場合:

```
SET MQIPADDRV=MQIPADDR_IPv4|MQIPADDR_IPv6
```

- ▶ **Linux** ▶ **UNIX** UNIX and Linux の場合:

```
export MQIPADDRV=MQIPADDR_IPv4|MQIPADDR_IPv6
```

- ▶ **IBM i** IBM i の場合:

```
ADDENVVAR ENVVAR(MQIPADDRV) VALUE(MQIPADDR_IPv4|MQIPADDR_IPv6)
```

## MQNAME

MQNAME では、IBM MQ プロセスが使用できるローカル NetBIOS 名を指定します。

その詳細について、およびクライアント上とサーバー上の優先順位の規則については、[205 ページの『Windows での NetBIOS 接続の定義』](#)を参照してください。

この変数を設定するには、このコマンドを使用します。

```
SET MQNAME>Your_env_Name
```

以下に例を示します。

```
SET MQNAME=CLIENT1
```

IBM MQ MQI client 上で複数の IBM MQ アプリケーションを同時に実行する場合、一部のプラットフォームの NetBIOS では、アプリケーションごとに異なる名前にする必要があります (MQNAME で設定)。

## MQSERVER

MQSERVER 環境変数は、最小チャンネルを定義するために使用されます。MQSERVER は、IBM MQ サーバーの位置および使用する通信方式を指定します。

TLS チャンネルの定義や、チャンネル出口があるチャンネルの定義に MQSERVER を使用することはできません。TLS チャンネルの定義方法について詳しくは、[TLS を使用したチャンネルの保護](#)を参照してください。

ConnectionName は、ネットワークの完全修飾名でなければなりません。ChannelName に、スラッシュ文字 (/) を使用することはできません。チャンネル名、トランスポート・タイプ、および接続名を区切るのに使

用されるためです。クライアント・チャンネルを定義するために MQSERVER 環境変数が使用されている場合は、最大メッセージ長 (MAXMSGL) 100 MB が使用されます。そのため、チャンネルの最大メッセージ・サイズは、実際にはサーバーの SVRCONN チャンネルに指定した値となります。

この変数を設定するには、次のコマンドのうちの 1 つを使用します。

- **Windows** Windows の場合:

```
SET MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

- **Linux** **UNIX** UNIX and Linux の場合:

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

- **IBM i** IBM i の場合:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

- **z/OS** z/OS の場合

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

*TransportType* は、ご使用の IBM MQ クライアント・プラットフォームに応じて、次の値のいずれかになります。

- LU62
- TCP
- NETBIOS
- SPX

*ConnectionName* は、接続名のコンマ区切りのリストにすることができます。リスト内の接続名は、クライアント接続テーブルでの複数接続の場合と同じように使用されます。*ConnectionName* リストは、クライアントが試行する複数接続を指定するために、キュー・マネージャー・グループの代替として使用される場合があります。複数インスタンスのキュー・マネージャーを構成する場合は、*ConnectionName* リストを使用して異なるキュー・マネージャーのインスタンスを指定することができます。

## TCP/IP デフォルト・ポート

TCP/IP の場合、IBM MQ は、デフォルトではチャンネルがポート 1414 に接続されるものと想定します。

このポートは、次のようにして変更できます。

- *ConnectionName* の最後の部分としてポート番号を括弧で囲んで追加します。

- **Windows** Windows の場合:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)
```

- **Linux** **UNIX** UNIX and Linux の場合:

```
export MQSERVER='ChannelName/TransportType/ConnectionName(PortNumber)'
```

- プロトコル名に、次の例のようにポート番号を追加して、mqclient.ini ファイルを変更します。

```
TCP:  
port=2001
```

- 210 ページの『UNIX and Linux での TCP/IP リスナーの使用』の説明に従って、サービス・ファイルに IBM MQ を追加します。

## SPX デフォルト・ソケット

デフォルトでは、SPX の場合、IBM MQ は、チャンネルがソケット 5E86 に接続されることを前提としています。

このポートは、次のようにして変更できます。

- ConnectionName の最後の部分としてソケット番号を括弧に入れて追加します。

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(SocketNumber)
```

SPX 接続の場合は、ConnectionName とソケットを network.node(socket) の形式で指定します。IBM MQ クライアントとサーバーが同じネットワークにある場合は、ネットワークを指定する必要はありません。デフォルトのソケットを使用する場合は、ソケットを指定する必要はありません。

- プロトコル名に、次の例のようにポート番号を追加して、qm.ini ファイルを変更します。

```
SPX:  
socket=5E87
```

## MQSERVER の使用法

MQSERVER 環境変数を使用して、IBM MQ MQI client・マシンとサーバー・マシンの間のチャンネルを定義すると、そのチャンネルが該当のアプリケーションにとって使用可能な唯一のチャンネルになり、クライアント・チャンネル定義テーブル (CCDT) は参照されなくなります。

この場合、サーバー・マシンで稼働状態になっているリスナー・プログラムが、アプリケーションが接続するキュー・マネージャーを判別します。このキュー・マネージャーは、リスナー・プログラムの接続先のキュー・マネージャーと同じです。

リスナーの接続先のキュー・マネージャー以外のキュー・マネージャーを MQCONN 要求と MQCONNX 要求で指定している場合、または MQSERVER パラメーター *TransportType* が認識されない場合は、MQCONN と MQCONNX 要求は失敗し、戻りコード MQRC\_Q\_MGR\_NAME\_ERROR で失敗します。

**Linux** **UNIX** UNIX and Linux システムでは、以下のいずれかの例のように、MQSERVER が定義される場合があります。

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56(2002)'  
export MQSERVER=CHANNEL1/LU62/BOX99
```

MQCONNX 要求に提供された MQCNO 構造体から MQCD 構造体が参照されない限り、すべての MQCONN または MQCONNX 要求は、ユーザーが定義したチャンネルの使用を試みます。参照される場合は、MQCD 構造体によって指定されたチャンネルが、MQSERVER 環境変数によって指定されたどのチャンネルよりも優先されます。

MQCHLLIB および MQCHLTAB が示すどのクライアント・チャンネル定義よりも、MQSERVER 環境変数のほうが優先されます。

## MQSERVER の取り消し

MQSERVER を取り消して、MQCHLLIB および MQCHLTAB が示すクライアント・チャンネル定義テーブルに戻るために、次のように入力します。

- **Windows** Windows の場合:

```
SET MQSERVER=
```

- Linux UNIX UNIX and Linux の場合:

```
unset MQSERVER
```

## MQSSLCRYP

MQSSLCRYP は、システムに存在している暗号ハードウェアの構成をユーザーに許可するパラメーター・ストリングを保持しています。許可された値は、ALTER QMGR コマンドの **SSLCRYP** パラメーター用と同じです。

この変数を設定するには、次のコマンドのうちの 1 つを使用します。

- Windows Windows システムの場合:

```
SET MQSSLCRYP=string
```

- Linux UNIX UNIX and Linux システムの場合:

```
export MQSSLCRYP=string
```

### 関連情報

[ALTER QMGR command SSLCRYP パラメーター](#)

## MQSSLFIPS

MQSSLFIPS では、IBM MQ で暗号化を実行するときに、FIPS 認定のアルゴリズムだけを使用するかどうかを指定します。この値は、ALTER QMGR コマンドの **SSLFIPS** パラメーター用と同じです。

FIPS 認定のアルゴリズムの使用は、暗号ハードウェアの使用に影響されます。[MQI クライアントでの実行時に FIPS 認定の CipherSpec のみを使用するように指定するを参照してください。](#)

この変数を設定するには、次のコマンドのうちの 1 つを使用します。

- Windows Windows システムの場合:

```
SET MQSSLFIPS=YES|NO
```

- Linux UNIX UNIX and Linux システムの場合:

```
export MQSSLFIPS=YES|NO
```

- IBM i IBM i の場合:

```
ADDENVVAR ENVVAR(MQSSLFIPS) VALUE (YES|NO)
```

デフォルトは NO です。

## MQSSLKEYR

MQSSLKEYR では、ユーザーのデジタル証明書を格納するキー・リポジトリの場所を語幹形式で指定します。語幹形式とは、絶対パスと、拡張子なしのファイル名を組み込んだ形式です。

詳細については、ALTER QMGR コマンドの [SSLKEYR](#) パラメーターを参照してください。

この変数を設定するには、次のコマンドのうちの 1 つを使用します。

- **Windows** Windows システムの場合:

```
SET MQSSLKEYR=pathname
```

- **Linux** **UNIX** UNIX and Linux システムの場合:

```
export MQSSLKEYR=pathname
```

- **IBM i** IBM i の場合:

```
ADDENVVAR ENVVAR(MQSSLKEYR) VALUE(pathname)
```

デフォルト値はありません。

## MQSSLPROXY

MQSSLPROXY では、OCSP 対応の GSKit による検査で使用する HTTP プロキシ・サーバーのホスト名とポート番号を指定します。

この変数を設定するには、次のコマンドのうちの 1 つを使用します。

- **Windows** Windows システムの場合:

```
SET MQSSLPROXY= string
```

- **Linux** **UNIX** UNIX and Linux システムの場合:

```
export MQSSLPROXY="string"
```

ストリングは、OCSP 検査用の GSKit に使用される HTTP プロキシ・サーバーのホスト名またはネットワーク・アドレスです。このアドレスの後に、オプションのポート番号が括弧で囲まれて続く場合があります。ポート番号を指定しないと、デフォルトの HTTP ポートである 80 が使用されます。

**Linux** **UNIX** 例えば、UNIX and Linux システムでは、以下のコマンドのいずれかを使用できます。

- ```
export MQSSLPROXY="proxy.example.com(80)"
```

- ```
export MQSSLPROXY="127.0.0.1"
```

## MQSSLRESET

MQSSLRESET では、TLS 秘密鍵の再ネゴシエーションが行われる前に、チャンネルで送受信される非暗号化バイト数を指定します。

秘密鍵の再ネゴシエーションについて詳しくは、[TLS 秘密鍵のリセット](#)を参照してください。

これは、0 から 999 999 999 の範囲の整数に設定することができます。デフォルトは 0 で、これは秘密鍵が再ネゴシエーションされないことを示します。TLS 秘密鍵のリセット・カウントを 1 バイトから 32 KB の範囲で指定すると、TLS チャンネルは 32 KB の秘密鍵リセット・カウントを使用します。この秘密鍵リセット・カウントは、TLS の秘密鍵のリセット値が小さい場合に生じる可能性のある、鍵の過度のリセットを防ぐためにあります。

この変数を設定するには、次のコマンドのうちの 1 つを使用します。

- ▶ **Windows** Windows システムの場合:

```
SET MQSSLRESET=integer
```

- ▶ **Linux** ▶ **UNIX** UNIX and Linux システムの場合:

```
export MQSSLRESET=integer
```

- ▶ **IBM i** IBM i の場合:

```
ADDENVVAR ENVVAR(MQSSLRESET) VALUE(integer)
```

## ULW MQSUITEB

IBM MQ は、UNIX, Linux, and Windows プラットフォーム上で、NSA Suite B 規格に準拠して動作するように構成することができます。

Suite B は、確実なセキュリティのレベルを提供するために、使用可能な暗号アルゴリズムのセットを制限します。

詳しくは、[Suite B 用 IBM MQ の構成](#)を参照してください。

## MQTCPTIMEOUT

IBM MQ が TCP 接続呼び出しを待機する時間。

## IBM MQ およびキュー・マネージャー構成情報の変更

IBM MQ または個々のキュー・マネージャーの動作を、インストール済み環境の要件に合わせて変更できます。

### このタスクについて

IBM MQ 構成情報を変更するには、IBM MQ の動作を決定する一連の構成属性 (またはパラメーター) に指定されている値を変更します。

IBM MQ 構成ファイルを編集して、属性情報を変更します。構成ファイルの編集は、ノード上のキュー・マネージャーの構成を変更するコマンドを使用して自動的に行うことも、標準的なテキスト・エディターを使用して手動で行うこともできます。詳しくは、[87 ページの『構成ファイルの編集』](#)を参照してください。

▶ **Windows** ▶ **Linux** Windows および Linux (x86 および x86-64 プラットフォーム) では、IBM MQ Explorer を使用して IBM MQ 構成ファイルを編集することもできます。

▶ **Windows** Windows システムでは、[amqmdain](#) で説明されているように、**amqmdain** を使用して構成情報を変更することもできます。

### 手順

- 使用するプラットフォームで IBM MQ およびキュー・マネージャーを構成する方法について詳しくは、以下のサブトピックを参照してください。

#### 関連概念

[5 ページの『IBM MQ の構成』](#)

1 台以上のコンピューターにキュー・マネージャーを 1 つ以上作成し、それらをビジネス・データが含まれるメッセージを処理するように開発システム、テスト・システム、実動システム上で構成します。

## 関連タスク

657 ページの『z/OS でのキュー・マネージャーの構成』

IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

## 関連情報

計画

管理 IBM MQ

## ULW UNIX, Linux, and Windows での構成情報の変更

UNIX, Linux, and Windows では、構成ファイルに保持されている IBM MQ 構成属性をノードのレベルおよびキュー・マネージャーのレベルで変更できます。

### このタスクについて

UNIX, Linux, and Windows プラットフォームでは、以下のファイルの中にある IBM MQ 構成属性を変更することができます。

- IBM MQ 構成ファイル(mqs.ini)はノード全体の IBM MQ の変更を有効にします。1 ノードに1つの mqs.ini ファイルがあります。mqs.ini に含まれているスタanzas について詳しくは、[106 ページの『IBM MQ 構成情報を変更するための属性』](#)を参照してください。
- 特定のキュー・マネージャーについて変更を加えるには、キュー・マネージャー構成ファイル(qm.ini)。qm.ini ファイルは、ノード上のそれぞれのキュー・マネージャーごとに1つずつあります。qm.ini に含まれるスタanzas についての詳細は、[112 ページの『キュー・マネージャー構成情報の変更』](#)を参照してください。

クライアント構成オプションは、クライアント構成ファイル(通常はmqclient.ini)内に個別に保持されます。詳しくは、[52 ページの『構成ファイルを使用したクライアントの構成』](#)を参照してください。

構成ファイル(またはスタanzas・ファイル)には、1つ以上のスタanzas が含まれています。スタanzas は、.ini ファイル内の行のグループで、全体で1つの機能を表しているか、ログ機能、チャンネル機能、およびインストール可能サービスなどのシステムの一部を定義するものです。

IBM MQ 構成ファイルは、キュー・マネージャーに関連したデータを見つけるために使用されるので、構成ファイルが存在しなかったり、正しくなかったりすると、一部または全部の MQSC コマンドが失敗します。また、アプリケーションは、IBM MQ 構成ファイルに定義されていないキュー・マネージャーには接続できません。

**重要:** 構成ファイルに対する変更が有効になるのは、通常、次回にキュー・マネージャーを始動したときです。

### 手順

- 構成ファイルを編集する前に、必要な場合に戻ることができるコピーを取得するためにバックアップしておいてください。
- コマンドまたは標準テキスト・エディターを使用して構成ファイルを編集します。  
詳しくは、[87 ページの『構成ファイルの編集』](#)を参照してください。

Windows および Linux (x86 および x86-64) システム上で、IBM MQ Explorer を使用して、構成ファイルに変更を行います。

詳しくは、[MQ エクスプローラーを使用した IBM MQ の構成](#) を参照してください。



Windows システムでは、IBM MQ Explorer を使用する代わりに、**amqmdain** コマンドを使用して構成ファイルに変更を行うこともできます。

詳しくは、[amqmdain](#) を参照してください。

## 関連概念

5 ページの『[IBM MQ の構成](#)』

1 台以上のコンピューターにキュー・マネージャーを 1 つ以上作成し、それらをビジネス・データが含まれるメッセージを処理するように開発システム、テスト・システム、実動システム上で構成します。

## 関連タスク

85 ページの『[IBM MQ およびキュー・マネージャー構成情報の変更](#)』

IBM MQ または個々のキュー・マネージャーの動作を、インストール済み環境の要件に合わせて変更できます。

94 ページの『[IBM i での構成情報の変更](#)』

インストール済み環境の要件に合わせてキュー・マネージャーの動作を変更するには、IBM MQ の動作を決定する一連の構成属性(またはパラメーター)に指定されている値を変更します。

112 ページの『[キュー・マネージャー構成情報の変更](#)』

個々のキュー・マネージャーの構成を変更するために使用できる属性によって、IBM MQ の設定がオーバーライドされます。

## 関連資料

106 ページの『[IBM MQ 構成情報を変更するための属性](#)』

IBM MQ for Windows システムおよび IBM MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、IBM MQ Explorer を使用して構成情報を変更します。他のシステムでは、mqqs.ini 構成ファイルを編集して構成情報を変更します。

## 関連情報

[計画](#)

[管理 IBM MQ](#)

## 構成ファイルの編集

コマンドまたは標準テキスト・エディターを使用して構成ファイルを編集します。

構成ファイルを編集する前に、必要な場合に復元できるようにバックアップをとっておいてください。

構成ファイルの編集方法には次の 2 つがあります。

- 自動。これには、ノード上のキュー・マネージャーの構成を変更するコマンドを使用します。
- 手動。これには、標準テキスト・エディターを使用します。

IBM MQ 構成ファイル内のデフォルト値は、インストール後に編集することができます。

構成ファイル属性のどれかに誤った値を設定した場合、その値は無視され、問題を示すオペレーター・メッセージが表示されます(その結果、その属性をまったく指定しなかった場合と同じになります。)

新規のキュー・マネージャーを作成するときは、次の操作をしてください。

- IBM MQ 構成ファイルのバックアップをとる
- 新しいキュー・マネージャー構成ファイルのバックアップをとる

コメントは、コメント・テキストの前に";"または"#"文字を追加することによって、構成ファイルに組み込むことができます。コメントを表すことなく、";"または"#"文字を使用したい場合は、文字の前に"\\"を接頭部として使用し、構成データの一部として使用することができます。

## どのようなときに構成ファイルの編集が必要か

バックアップからのリカバリー、キュー・マネージャーの移動、デフォルトのキュー・マネージャーの変更を行うため、または IBM サポートの補助のために、構成ファイルを編集します。

構成ファイルを編集する必要があるのは、次のような場合です。

- 構成ファイルが失われる場合。(可能な場合はバックアップからリカバリーする。)
- 1 以上のキュー・マネージャーを新しいディレクトリーに移す必要がある場合。
- デフォルトのキュー・マネージャーを変更する必要がある場合。これは、既存のキュー・マネージャーを誤って削除してしまった場合などに必要です。

- IBM サポートから編集するよう勧められた場合。

## 構成ファイルの優先順位

属性の値は、複数の場所で定義されます。コマンドで設定された属性は、構成ファイルの属性よりも優先されます。

構成ファイルの属性値は、次の優先順位に従って設定されます。

- コマンド行に入力されたパラメーターは、構成ファイル内で定義されている値より優先されます。
- qm.ini ファイル内で定義されている値は、mqs.ini ファイル内で定義されている値より優先されます。

## 構成ファイルのコメント

Linux UNIX

セミコロン「;」とハッシュ「#」の文字を使用して、構成ファイル内でコメントの先頭にマークを付けることができます。これにより、行全体をコメントとしてマークしたり、設定の値に含まれることのない行末のコメントを表示したりできます。

値にこれらの文字のいずれかが必要な場合は、円記号文字「\」を使用してその文字をエスケープする必要があります。

## ULW IBM MQ 構成ファイル mqs.ini

IBM MQ 構成ファイル mqs.ini には、ノード上のすべてのキュー・マネージャーに関連する情報が入っています。これは、インストール時に自動的に作成されます。

## ディレクトリーの場所

Linux UNIX

UNIX および Linux では、データ・ディレクトリーとログ・ディレクトリーは、それぞれ必ず /var/mqm と /var/mqm/log になります。

Windows

Windows システムでは、データ・ディレクトリー mqs.ini の場所とログ・ディレクトリーの場所はレジストリーに保管されるため、どちらの場所も変更可能です。UNIX および Linux システム上の mqinst.ini に記載されているインストール構成情報は、Windows 上に mqinst.ini ファイルがないため、レジストリーにも含まれています (93 ページの『インストール構成ファイル、mqinst.ini』を参照)。

Windows

Windows システムの mqs.ini ファイルは、HKLM\SOFTWARE\IBM\WebSphere MQ キーで指定された WorkPath によって指定されます。このファイルの内容は、次のとおりです。

- キュー・マネージャーの名前
- デフォルト・キュー・マネージャーの名前
- 各キュー・マネージャーに関連したファイルの位置

## 新規インストールの LogDefaults スタンザ

新規 IBM MQ インストール用に提供された LogDefaults スタンザには、属性の明示的な値が含まれていません。属性が欠けている場合は、新規キュー・マネージャーの作成時にこの値のデフォルトが使用されます。デフォルト値は、89 ページの図 5 にある LogDefaults スタンザに示されています。

LogBufferPages 属性に値 0 を指定した場合は、512 を意味します。

デフォルト以外の値が必要な場合は、LogDefaults スタンザで明示的に値を指定する必要があります。

## mqs.ini ファイルの例

UNIX

```

#*****#
#* Module Name: mqs.ini                                     *#
#* Type       : IBM MQ Machine-wide Configuration File   *#
#* Function    : Define IBM MQ resources for an entire machine *#
#*****#
#* Notes      :                                           *#
#* 1) This is the installation time default configuration *#
#*                                                    *#
#*****#
AllQueueManagers:
#*****#
#* The path to the qmgrs directory, below which queue manager data *#
#* is stored                                                         *#
#*****#
DefaultPrefix=/var/mqm

LogDefaults:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=0
  LogDefaultPath=/var/mqm/log

QueueManager:
  Name=saturn.queue.manager
  Prefix=/var/mqm
  Directory=saturn!queue!manager
  InstallationName=Installation1

QueueManager:
  Name=pluto.queue.manager
  Prefix=/var/mqm
  Directory=pluto!queue!manager
  InstallationName=Installation2

DefaultQueueManager:
  Name=saturn.queue.manager

ApiExitTemplate:
  Name=OurPayrollQueueAuditor
  Sequence=2
  Function=EntryPoint
  Module=/usr/ABC/auditor
  Data=123

ApiExitCommon:
  Name=MQPoliceman
  Sequence=1
  Function=EntryPoint
  Module=/usr/MQPolice/tmqp
  Data=CheckEverything

```

図 5. UNIX の IBM MQ 構成ファイルの例

## ULW キュー・マネージャー構成ファイル qm.ini

キュー・マネージャー構成ファイル qm.ini には、特定のキュー・マネージャーに関係のある情報が含まれています。

各キュー・マネージャーごとに、1つのキュー・マネージャー構成ファイルがあります。qm.ini ファイルは、関連したキュー・マネージャーが作成されるときに、自動的に作成されます。

**注:** qm.ini ファイルに対して行った変更が有効になるタイミングの詳細については、[86 ページの『UNIX, Linux, and Windows での構成情報の変更』](#)を参照してください。

**V 9.0.0.2** **V 9.0.4** IBM MQ 9.0.4 および IBM MQ 9.0.0 Fix Pack 2 以降、**strmqm** コマンドは、キュー・マネージャーを完全に開始する前に qm.ini ファイル内の CHANNELS および SSL スタンザの構文をチェックします。これにより、**strmqm** によって qm.ini ファイルのエラーが検出された場合に、誤りを容易に見つけたり、それを迅速に修正したりできます。詳しくは、[strmqm](#) を参照してください。

## qm.ini ファイルの場所

**Linux** **UNIX** UNIX and Linux システムでは、qm.ini ファイルはキュー・マネージャーが占有するディレクトリー・ツリーのルートに保持されます。例えば、QMNAME という名前のキュー・マネージャーの構成ファイルのパスと名前は、次のとおりです。

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

**Windows** Windows システムでは、qm.ini ファイルの場所は、HKLM\SOFTWARE\IBM\WebSphere MQ キーで指定された WorkPath によって指定されます。例えば、QMNAME という名前のキュー・マネージャーの構成ファイルのパスと名前は、以下のとおりです。

```
C: \ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```

キュー・マネージャーの名前の長さは 48 文字までです。ただし、この名前が有効である、または固有であるかは保証されません。このため、キュー・マネージャー名に基づいてディレクトリー名が生成されます。このプロセスは名前変換と呼ばれています。説明については、[IBM MQ ファイル名についての理解を参照してください](#)。

## qm.ini ファイルの例

**Linux** **UNIX**

以下の例は、IBM MQ for UNIX システムおよび Linux システムでキュー・マネージャー構成ファイルに属性のグループを配置する方法を示しています。

**V 9.0.5** IBM MQ 9.0.5 からのキュー・マネージャー構成ファイルの例を以下に示します。

```
##* Module Name: qm.ini                                ##*
##* Type       : IBM MQ queue manager configuration file ##*
##* Function   : Define the configuration of a single queue manager ##*
##*           ##*
##* Notes     :                                       ##*
##* 1) This file defines the configuration of the queue manager ##*
##*           ##*
##*           ##*
ExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

Service:
  Name=AuthorizationService
  EntryPoints=14

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module=amqzfu
  ComponentDataSize=0

Log:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=0 1
  LogPath=/var/mqm/log/saturn!queue!manager/

XAResourceManager:
  Name=DB2 Resource Manager Bank
  SwitchFile=/usr/bin/db2swit
  XAOpenString=MQBankDB
  XACloseString=
  ThreadOfControl=THREAD

Channels: 2
```

```

MaxChannels=200
MaxActiveChannels=100
MQIBindType=STANDARD

TCP:
  SndBuffSize=0
  RcvBuffSize=0
  RcvSndBuffSize=0
  RcvRcvBuffSize=0
  ClntSndBuffSize=0
  ClntRcvBuffSize=0
  SvrSndBuffSize=0
  SvrRcvBuffSize=0

QMErrorLog:
  ErrorLogSize=262144
  ExcludeMessage=7234
  SuppressMessage=9001,9002,9202
  SuppressInterval=30

ApiExitLocal:
  Name=ClientApplicationAPIchecker
  Sequence=3
  Function=EntryPoint
  Module=/usr/Dev/ClientAppChecker
  Data=9.20.176.20

TuningParameters:
  ImplSyncOpenOutput=2

```

IBM MQ 9.0.4 以前と LTS の構成ファイルの例を以下に示します。

```

## Module Name: qm.ini                                ##
## Type       : IBM MQ queue manager configuration file ##
## Function   : Define the configuration of a single queue manager ##
##           ##                                       ##
##*****##
## Notes     :                                       ##
## 1) This file defines the configuration of the queue manager ##
##           ##                                       ##
##*****##

ExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

Service:
  Name=AuthorizationService
  EntryPoints=14

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module=amqzfu
  ComponentDataSize=0

Log:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=0 1
  LogPath=/var/mqm/log/saturn!queue!manager/

XAResourceManager:
  Name=DB2 Resource Manager Bank
  SwitchFile=/usr/bin/db2swit
  XAOpenString=MQBankDB
  XACloseString=
  ThreadOfControl=THREAD

Channels: 2
  MaxChannels=200
  MaxActiveChannels=100
  MQIBindType=STANDARD

TCP:

```

```
SndBuffSize=0
RcvBuffSize=0
RcvSndBuffSize=0
RcvRcvBuffSize=0
ClntSndBuffSize=0
ClntRcvBuffSize=0
SvrSndBuffSize=0
SvrRcvBuffSize=0

QMErrorLog:
  ErrorLogSize=262144
  ExcludeMessage=7234
  SuppressMessage=9001,9002,9202
  SuppressInterval=30

ApiExitLocal:
  Name=ClientApplicationAPIChecker
  Sequence=3
  Function=EntryPoint
  Module=/usr/Dev/ClientAppChecker
  Data=9.20.176.20
```

#### 注:

1. LogBufferPages の値を 0 にすると、値 512 が使用されます。
2. Channel スタンザの詳細については、[187 ページの『初期設定および構成ファイル』](#)を参照してください。
3. XAResourceManager スタンザの最大数は 255 に制限されています。ただし、トランザクションの性能低下を防ぐため、少数のスタンザのみを使用してください。
4.  IBM MQ 8.0 から、マルチプラットフォーム上の新規キュー・マネージャーの場合、qm.ini file の TCP スタンザ内のデフォルトの TCP 送信および受信バッファ・サイズは、オペレーティング・システムによって管理されるように設定されています。新規キュー・マネージャーは、自動的に送信受信バッファが以下の値にデフォルト設定された状態で作成されます。

```
TCP:
SndBuffSize=0
RcvBuffSize=0
RcvSndBuffSize=0
RcvRcvBuffSize=0
ClntSndBuffSize=0
ClntRcvBuffSize=0
SvrSndBuffSize=0
SvrRcvBuffSize=0
```

IBM MQ 8.0 より前には、手動チューニングを行わない場合、これらの値は固定サイズの 32Kb バッファにデフォルト設定されていました。

この変更は、新規キュー・マネージャーにのみ適用されます。以前のリリースからマイグレーションされるキュー・マネージャーの TCP 送信/受信バッファ設定は保持されます。

マイグレーションされるキュー・マネージャーに関してオペレーティング・システムのデフォルト動作を使用するよう TCP バッファ・サイズを手動で設定する方法については、[TCP](#)、[LU62](#)、[NETBIOS](#)、および [SPX](#) を参照してください。

## AccessMode スタンザ

### Windows

Windows の qm.ini ファイルには、追加の AccessMode スタンザが含まれています。

```
AccessMode:
SecurityGroup=wmq\wmq
```

## APIExitLocal スタンザ

ApiExitLocal スタンザでは、単一の Module のみを指定できます。また、以下のように 4 つのモジュールを提供する必要があります。

- 32 ビット、スレッドなし
- 32 ビット、スレッドあり
- 64 ビット、スレッドなし
- 64 ビット、スレッドあり

IBM MQ は、提供されたモジュール名に `_r` を付加して出口のスレッド・バージョンを識別しますが、IBM MQ は 32 ビットと 64 ビットのバリエーションに直接同等のメカニズムを提供しないことに注意してください。

非修飾のモジュール名が提供されている場合は、`/var/mqm/exits` と `/var/mqm/exits64` で IBM MQ 32 ビットバリエーションと 64 ビットバリエーションを検索します。

例えば、`module=amqsaxe` は以下を暗黙に示します。

```
/var/mqm/exits/amqsaxe - 32 bit unthreaded variant
/var/mqm/exits/amqsaxe_r - 32 bit threaded variant
/var/mqm/exits64/amqsaxe - 64 bit unthreaded variant
/var/mqm/exits64/amqsaxe_r - 64 bit threaded variant
```

`prefix/mqm/samp/bin` に同梱されている `amqsaxe0` および `amqsaxe0_r` のバージョンは、作成されたプラットフォーム上のキュー・マネージャーのネイティブ・サイズ (現在はすべて 64 ビット) 用に作成され、同じネイティブ・サイズで実行されているアプリケーションでのみ使用できます。

### 関連資料

124 ページの『TCP、LU62、および NETBIOS』

ネットワーク・プロトコル構成パラメータを指定するには、以下のキュー・マネージャー・プロパティ・ページ、または `qm.ini` ファイル内のスタンザを使用します。これらのスタンザは、チャンネルのデフォルトの属性を指定変更します。

## ULW インストール構成ファイル、mqinst.ini

UNIX または Linux では、インストール構成ファイル `mqinst.ini` には、すべての IBM MQ インストールに関する情報が含まれています。Windows では、インストール構成情報はレジストリーにあります。

### mqinst.ini ファイルの場所

Linux → UNIX

UNIX and Linux システムでは、`mqinst.ini` ファイルは `/etc/opt/mqm` ディレクトリーにあります。これには、どのインストール済み環境がプライマリーのインストール済み環境 (もしあれば) であるかが、各インストール済み環境に関する以下の情報とともに書き込まれています。

- インストール名
- インストールの説明
- インストール ID
- インストール・パス

**重要:** `mqinst.ini` ファイルの形式は固定されておらず、変更される可能性があるため、このファイルは編集したり、直接参照したりしないでください。

内部でのみ使用されるインストール ID は、自動的に設定され、変更することはできません。

`mqinst.ini` ファイルを直接編集する代わりに、以下のコマンドを使用して、ファイル内の値を作成、削除、照会、および変更する必要があります。

`crtmqinst` は、項目を作成します。

[dlmqinst](#) は、項目を削除します。  
[dspmqinst](#) は、項目を表示します。  
[setmqinst](#) は、項目を設定します。

## Windows でのインストール構成情報

### Windows

Windows 上に `mqinst.ini` ファイルがありません。インストール構成情報はレジストリーにあり、以下のキーで保持されます。

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName
```

**重要:** このキーの形式は固定されておらず、変更される可能性があるため、このキーは編集したり、直接参照したりしないでください。

その代わりに、以下のコマンドを使用して、レジストリーの値を照会および変更する必要があります。

[dspmqinst](#) は、項目を表示します。  
[setmqinst](#) は、項目を設定します。

Windows では、**crtmqinst** コマンドおよび **dlmqinst** コマンドは使用できません。必要なレジストリー項目の作成および削除は、インストール・プロセスおよびアンインストール・プロセスによって処理されます。

### Windows

## IBM i IBM i での構成情報の変更

インストール済み環境の要件に合わせてキュー・マネージャーの動作を変更するには、IBM MQ の動作を決定する一連の構成属性 (またはパラメーター) に指定されている値を変更します。

### このタスクについて

構成属性は、IBM MQ 構成ファイルを編集して変更します。

### 手順

- IBM i での構成値の変更についての詳細は、以下のトピックを参照してください。
  - [95 ページの『IBM i の IBM MQ 構成ファイル』](#)
  - [96 ページの『IBM i での構成情報を変更するための属性』](#)
  - [99 ページの『IBM i でのキュー・マネージャー構成情報の変更』](#)
  - [104 ページの『IBM i 用の mqsc.ini ファイルと qm.ini ファイルの例』](#)

### 関連概念

[5 ページの『IBM MQ の構成』](#)

1 台以上のコンピューターにキュー・マネージャーを 1 つ以上作成し、それらをビジネス・データが含まれるメッセージを処理するように開発システム、テスト・システム、実動システム上で構成します。

### 関連タスク

[85 ページの『IBM MQ およびキュー・マネージャー構成情報の変更』](#)

IBM MQ または個々のキュー・マネージャーの動作を、インストール済み環境の要件に合わせて変更できます。

[86 ページの『UNIX, Linux, and Windows での構成情報の変更』](#)

UNIX, Linux, and Windows では、構成ファイルに保持されている IBM MQ 構成属性をノードのレベルおよびキュー・マネージャーのレベルで変更できます。

[112 ページの『キュー・マネージャー構成情報の変更』](#)

個々のキュー・マネージャーの構成を変更するために使用できる属性によって、IBM MQ の設定がオーバーライドされます。

## 関連資料

106 ページの『[IBM MQ 構成情報を変更するための属性](#)』

IBM MQ for Windows システムおよび IBM MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、IBM MQ Explorer を使用して構成情報を変更します。他のシステムでは、mqqs.ini 構成ファイルを編集して構成情報を変更します。

## 関連情報

[計画](#)

[管理 IBM MQ](#)

## IBM i の IBM MQ 構成ファイル

この情報は、IBM MQ for IBM i の構成方法を理解するために使用します。

IBM i では、IBM MQ 構成属性を以下で変更します。

- IBM MQ 構成ファイル mqqs.ini は、ノード全体での変更に影響します。mqqs.ini ファイルは、IBM MQ のインストール済み環境ごとに 1 つあります。
- キュー・マネージャー構成ファイル qm.ini は、特定のキュー・マネージャーの変更に影響します。qm.ini ファイルは、ノード上のそれぞれのキュー・マネージャーごとに 1 つずつあります。

.ini ファイルは、IFS にあるストリーム・ファイルであることに注意してください。

構成ファイル (スタンザ・ファイルとも呼ばれます) には、1 つ以上のスタンザが格納されています。スタンザは .ini ファイル内の行のグループで、全体で共通の機能を持つか、または (ログ機能、チャンネル機能など) システムの一部を定義します。構成ファイルに対する変更は、次回にキュー・マネージャーを始動するまで有効になりません。

## 構成ファイルの編集

構成ファイルを編集する前に、必要な場合に復元できるようにバックアップをとっておいてください。

構成ファイルの編集方法には次の 2 つがあります。

- 自動。これには、ノード上のキュー・マネージャーの構成を変更するコマンドを使用します。
- 手動。これには EDTF CL エディターを使用します。

IBM MQ 構成ファイル内のデフォルト値は、インストール後に編集することができます。構成ファイル属性のどれかに誤った値を設定した場合、その値は無視され、問題を示すオペレーター・メッセージが表示されます (その結果、その属性をまったく指定しなかった場合と同じになります)。

新規のキュー・マネージャーを作成するときは、次の操作をしてください。

- IBM MQ 構成ファイルのバックアップをとる。
- 新しいキュー・マネージャー構成ファイルのバックアップをとる。

## どのようなときに構成ファイルの編集が必要か

構成ファイルを編集する必要があるのは、次のような場合です。

- 構成ファイルが失われた場合。バックアップがある場合は、それから回復します。
- 1 以上のキュー・マネージャーを新しいディレクトリーに移す必要がある場合。
- デフォルトのキュー・マネージャーを変更する必要がある場合。これは、既存のキュー・マネージャーを誤って削除してしまった場合などに必要です。
- IBM サポートから編集するよう勧められた場合。

## 構成ファイルの優先順位

構成ファイルの属性値は、次の優先順位に従って設定されます。

- コマンド行に入力されたパラメーターは、構成ファイル内で定義されている値より優先されます。
- qm.ini ファイル内で定義されている値は、mqs.ini ファイル内で定義されている値より優先されます。

## IBM MQ 構成ファイル mqs.ini

IBM MQ 構成ファイル mqs.ini には、IBM MQ インストール済み環境のすべてのキュー・マネージャーに関連する情報が含まれています。これは、インストール時に自動的に作成されます。特に、mqs.ini 構成ファイルは、各キュー・マネージャーに関連したデータを見つけるために使用されます。

mqs.ini ファイルは、/QIBM/UserData/mqm に保管されます。

mqs.ini ファイルには、次のものが含まれています。

- キュー・マネージャーの名前。
- デフォルト・キュー・マネージャーの名前。
- 各キュー・マネージャーに関連したファイルの位置。
- API 出口を識別する情報 (詳細については、[API 出口の構成](#)を参照)。

## キュー・マネージャー構成ファイル qm.ini

キュー・マネージャー構成ファイル qm.ini には、特定のキュー・マネージャーに関係のある情報が含まれています。各キュー・マネージャーごとに、1つのキュー・マネージャー構成ファイルがあります。qm.ini ファイルは、関連したキュー・マネージャーが作成される時に、自動的に作成されます。

qm.ini ファイルは、mqmdata directory/QMNAME/qm.ini に保持されます。ここで、mqmdata directory はデフォルトで /QIBM/UserData/mqm であり、QMNAME は初期設定ファイルが適用されるキュー・マネージャーの名前です。

注:

1. mqmdata directory は、mqs.ini ファイル内で変更できます。
2. キュー・マネージャーの名前の長さは 48 文字までです。ただし、この名前が有効である、または固有であるかは保証されません。このため、キュー・マネージャー名に基づいてディレクトリー名が生成されます。このプロセスは名前変換と呼ばれています。詳細については、[IBM MQ for IBM i キュー・マネージャー・ライブラリー名についての理解](#)を参照してください。

## IBM i IBM i での構成情報を変更するための属性

この情報は、構成情報のスタンザを理解するために使用します。

次の属性のグループが、mqs.ini にあります。

- [96 ページの『AllQueueManagers スタンザ』](#)
- [98 ページの『DefaultQueueManager スタンザ』](#)
- [98 ページの『ExitProperties スタンザ』](#)
- [98 ページの『QueueManager スタンザ』](#)

API 出口 (ApiExitCommon および ApiExitTemplate) に関連する 2 つのスタンザもあります。これらの使用について詳しくは、[API 出口の構成](#)を参照してください。

## AllQueueManagers スタンザ

AllQueueManagers スタンザには次のものを指定できます。

- キュー・マネージャーに関連したファイルが保管されている qmgrs ディレクトリーへのパス

- 実行可能ライブラリーへのパス
  - EBCDIC フォーマットのデータを ASCII フォーマットに変換する方式
- スタンザの説明で、下線が引かれているのがデフォルト値で、|記号は *or* (または) を表します。

#### DefaultPrefix=directory\_name

キュー・マネージャー・データが保管されている qmgrs ディレクトリーへのパス。キュー・マネージャーのデフォルトの接頭部を変更する場合は、インストール時に作成されているディレクトリー構造を複製する必要があります。特に、qmgrs 構造を作成する必要があります。デフォルト接頭部を変更する前に IBM MQ を停止してください。IBM MQ の再始動は、構造が新しい場所に移され、デフォルト接頭部が変更された後に行ってください。

デフォルト接頭部を変更する代わりに、環境変数 MQSPREFIX を使用して、CRTMQM コマンドの DefaultPrefix を指定変更することもできます。

#### ConvEBCDICNewline=NL\_TO\_LF|TABLE|ISO

EBCDIC コード・ページには、ASCII コード・ページではサポートされない改行 (NL) 文字が含まれています。ただし、ISO 仕様の ASCII には、この改行文字と同等の文字が含まれているものもいくつかあります。

ConvEBCDICNewline 属性には、EBCDIC の NL 文字を ASCII フォーマットに変換するときに IBM MQ が使用する方式を指定します。

#### NL\_TO\_LF

EBCDIC から ASCII へのすべての変換について、EBCDIC NL 文字 (X'15') を ASCII 改行 (LF) 文字 (X'0A') に変換します。

NL\_TO\_LF はデフォルトです。

#### 表

EBCDIC から ASCII へのすべての変換用に IBM i で使用される変換テーブルに従って、EBCDIC NL 文字を変換します。

この種の変換の結果は、言語により異なります。

#### ISO

次の場合は ISO を指定します。

- TABLE メソッドを使用して ISO CCSID を変換したい場合
- NL\_TO\_CF メソッドを使用してその他のすべての CCSID を変換したい場合

使用できる ISO CCSID は、[97 ページの表 10](#) に示されています。

表 10. 使用できる ISO CCSID のリスト	
CCSID	コード・セット
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

ASCII CCSID が ISO サブセットの 1 つではない場合は、デフォルトにより、ConvEBCDICNewline は NL\_TO\_LF に設定されます。

## DefaultQueueManager スタンザ

DefaultQueueManager スタンザには、ノードのデフォルト・キュー・マネージャーを指定します。

### Name=default\_queue\_manager

デフォルト・キュー・マネージャーは、キュー・マネージャー名が明示的に指定されていないすべてのコマンドを処理します。DefaultQueueManager 属性は、新規のデフォルト・キュー・マネージャーを作成すると、自動的に更新されます。誤って新規のデフォルト・キュー・マネージャーを作成した場合、それを元のプログラムに戻したいときは、DefaultQueueManager 属性を手動で変更する必要があります。

## ExitProperties スタンザ

ExitProperties スタンザには、キュー・マネージャーの出口プログラムが使用する構成オプションを指定します。

スタンザの説明で、下線が引かれているのがデフォルト値で、| 記号は *or* (または) を表します。

### CLWLMode=SAFE|FAST

クラスター・ワークロード出口 CLWL では、MQI 呼び出し (例えば、MQOPEN または MQPUT) に応じて、クラスター内のどのクラスター・キューをオープンするかを指定できます。CLWL 出口は、CLWLMode 属性に指定する値に応じて、FAST モードまたは SAFE モードで実行されます。CLWLMode 属性を省略する場合は、クラスター・ワークロード出口は SAFE モードで実行されます。

#### SAFE

CLWL 出口は、キュー・マネージャーとは別のプロセスで実行します。これがデフォルトです。

SAFE モードでの実行中にユーザー作成の CLWL に問題が起きた場合は、次のような結果が生じます。

- CLWL サーバー・プロセス (amqzlw0) は失敗します。
- キュー・マネージャーは CLWL サーバー・プロセスを再始動します。
- エラーがエラー・ログに記録されます。進行中の MQI 呼び出しがある場合は、不成功を示す戻りコードが戻されます。

キュー・マネージャーの健全性は維持されます。

注: CLWL 出口を別個のプロセスで実行すると、パフォーマンスに悪影響を及ぼすおそれがあります。

#### FAST

クラスター出口は、キュー・マネージャー・プロセスでインラインで実行します。

このオプションを指定すると、SAFE モードで実行した場合のオーバーヘッドを回避できるのでパフォーマンスは向上しますが、代わりにキュー・マネージャーの健全性が保証されないこととなります。CLWL 出口を FAST モードで実行するのは、CLWL 出口にまったく問題がないという確信があり、パフォーマンス・オーバーヘッドが特に重要な要素である場合のみにしてください。

CLWL 出口を FAST モードで実行中に問題が発生すると、キュー・マネージャーは失敗し、キュー・マネージャーの健全性が損なわれることがあります。

## QueueManager スタンザ

キュー・マネージャーごとに QueueManager スタンザが 1 つずつあります。これらの属性は、キュー・マネージャーの名前と、そのキュー・マネージャーに関連したファイルが含まれているディレクトリーの名前を指定します。ディレクトリーの名前はキュー・マネージャー名に基づいていますが、キュー・マネージャー名が有効なファイル名でない場合は、変換されます。

名前変換について詳しくは、[IBM MQ for IBM i キュー・マネージャー・ライブラリー名についての理解を参照してください](#)。

### Name=queue\_manager\_name

キュー・マネージャーの名前。

### **Prefix=prefix**

キュー・マネージャーのファイルが保管される場所。デフォルトでは、mqs.ini ファイル内の AllQueueManager スタンザの DefaultPrefix 属性に指定されている値と同じです。

### **Directory=name**

キュー・マネージャー・ファイルが保管されている prefix\QMGRS ディレクトリーの下にあるサブディレクトリーの名前。この名前はキュー・マネージャー名に基づくものですが、重複する名前がある場合やキュー・マネージャー名が無効なファイル名の場合でも、変換されることがあります。

### **Library=name**

このキュー・マネージャーに関連する IBM i オブジェクト (例えばジャーナルやジャーナル・レシーバー) を保管するライブラリーの名前。この名前はキュー・マネージャー名に基づくものですが、重複する名前がある場合、またはキュー・マネージャー名が無効なライブラリー名の場合は、変換される場合があります。

## **IBM i IBM i でのキュー・マネージャー構成情報の変更**

この情報は、キュー・マネージャー構成スタンザを理解するために使用します。

API 出口 (ApiExitCommon および ApiExitTemplate) に関連する 2 つのスタンザがあります。これらのスタンザの使用の詳細については、[API 出口の構成](#)を参照してください。

次の各属性グループは、特定のキュー・マネージャーの qm.ini ファイルにあり、mqs.ini に設定されている値を指定変更するために使用されます。

特定のオプションの構成情報の変更については、以下のトピックを参照してください。

- [99 ページの『Log スタンザ \(IBM i\)』](#)
- [99 ページの『Channels スタンザ \(IBM i\)』](#)
- [101 ページの『QMErrorLog スタンザ \(IBM i\)』](#)
- [102 ページの『TCP スタンザ \(IBM i\)』](#)
- [68 ページの『クライアント構成ファイルの PreConnect スタンザ』](#)

## **IBM i Log スタンザ (IBM i)**

ログ・ファイルを構成するパラメーター。

Log スタンザには、特定のキュー・マネージャーのログ属性を指定します。デフォルトでは、これらの属性は、キュー・マネージャーの作成時に mqs.ini ファイル内の LogDefaults スタンザで指定されている設定から継承されます。

他とは異なるキュー・マネージャーを構成する場合は、このスタンザの属性のみを変更します。

qm.ini ファイル内の属性に指定されている値は、キュー・マネージャーの始動時に読み取られます。この構成ファイルは、キュー・マネージャーの作成時に作成されます。

### **LogBufferSize**

ジャーナル・バッファー・サイズ (バイト)。32 000 から 15 761 440 までの範囲の数値を入力します。デフォルトは 32 000 です。

### **LogPath=library\_name**

このキュー・マネージャーのジャーナルおよびジャーナル・レシーバーを保管するために使用するライブラリーの名前。

### **LogReceiverSize**

ジャーナル・レシーバーのサイズ (キロバイト)。デフォルトは 100 000 です。

## **IBM i Channels スタンザ (IBM i)**

Channels スタンザにはチャンネルに関する情報が入っています。

### **MaxChannels=100|number**

使用できる現行チャンネルの最大数。z/OS では、値は 1 から 9999 までの範囲でなければなりません。デフォルト値は 200 です。その他のすべてのプラットフォームのデフォルト値は 100 です。

**MaxActiveChannels=MaxChannels\_value**

任意のある時点でアクティブにできるチャンネルの最大数。デフォルトは、MaxChannels 属性に指定されている値です。

**MaxInitiators=3|number**

開始プログラムの最大数。デフォルト値と最大値は 3 です。

**MQIBINDTYPE=FASTPATH|STANDARD**

アプリケーションのバインディング。

**FASTPATH**

チャンネルは MQCONNX FASTPATH を使用して接続されます。つまり、エージェント・プロセスはありません。

**STANDARD**

チャンネルは STANDARD を使用して接続されます。

**ThreadedListener=NO|YES**

リスナーとして RUNMQLSR (YES) かまたは AMQCLMAA (NO) を始動する。

ThreadedListener=YES を指定すると、すべてのチャンネルは単一ジョブのスレッドとして実行されます。この場合単一ジョブで使用可能なリソースへの接続数が制限されます。

ThreadedListener=NO を指定すると、非スレッド化リスナー (AMQCLMAA) はインバウンドの各 TCP/IP チャンネルで新しいレスポンス・ジョブ (AMQCRSTA) を開始します。この手法の欠点は、新しい AMQCRSTA ジョブを開始するのが、RUNMQLSR ジョブ内でスレッドを開始するほどには速くないことです。そのため、非スレッド化リスナーの接続時間はスレッド化リスナーのものより遅くなります。

**AdoptNewMCA=NO|SVR|SNDR|RCVR|CLUSRCVR|ALL|FASTPATH**

IBM MQ がチャンネルの開始要求を受け取ったときに、同じチャンネルに対して amqcrsta プロセスが存在している場合、新しいプロセスを開始するには、その既存プロセスを停止する必要があります。AdoptNewMCA 属性を使用すると、指定したチャンネル・タイプについて、既存のプロセスの終了と新規プロセスの開始を制御することができます。

特定のチャンネル・タイプについて AdoptNewMCA 属性を指定してあるときに、そのチャンネルがすでに実行中であるために新規チャンネルが始動できない場合は、次のようになります。

1. 新規チャンネルは既存のチャンネルの終了を試みます。
2. 既存のチャンネル・サーバーが、AdoptNewMCATimeout に指定されている待ち時間が満了するまでに終了しない場合、既存のチャンネル・サーバーのプロセス (またはスレッド) は終了します。
3. ステップ 2 の実行後も既存のチャンネル・サーバーが終了していない場合は、AdoptNewMCATimeout の待ち時間が 2 回目に満了した時点で、IBM MQ はそのチャンネルを終了し、CHANNEL IN USE エラーを返します。

下記のリストから 1 以上の値を選択し、コンマまたはブランクで区切って指定します。

**NO**

AdoptNewMCA 機能が不要であることを意味します。これがデフォルトです。

**SVR**

サーバー・チャンネルが採用されます。

**SNDR**

送信側チャンネルが採用されます。

**RCVR**

受信側チャンネルが採用されます。

**CLUSRCVR**

クラスター受信側チャンネルが採用されます。

**ALL**

FASTPATH チャンネルを除くすべてのチャンネル・タイプが採用されます。

## FASTPATH

該当チャンネルが FASTPATH チャンネルのときは、そのチャンネルが採用されます。これが起きるのは、AdoptNewMCA=RCVR,SVR,FASTPATH などのように該当のチャンネル・タイプも指定されている場合だけです。

**重要!** FASTPATH チャンネルの場合は、キュー・マネージャーの内部設計が原因で、AdoptNewMCA 属性が予期しない動作をすることができます。FASTPATH チャンネルについて AdoptNewMCA 属性を使用可能にするときは、十分な注意が必要です。

## AdoptNewMCATimeout=60|1-3600

新規プロセスが旧プロセスの終了を待つ時間を秒数で指定します。1 から 3600 秒の範囲内の値を指定してください。デフォルト値は 60 です。

## AdoptNewMCACheck=QM|ADDRESS|NAME|ALL

AdoptNewMCACheck 属性には、AdoptNewMCA 属性を使用可能にするときに必要なタイプ検査を指定します。故意または不注意によりチャンネルが遮断されるのを防ぐためには、可能であれば次の 3 種類の検査をすべて行うことが重要です。最低限、チャンネル名が一致することだけは確認してください。

下記から 1 以上の値を選択し、コンマまたはブランクで区切って指定します。

### QM

リスナー・プロセスは、キュー・マネージャー名が一致することを検査します。

### ADDRESS

リスナー・プロセスは、通信アドレス (例えば、TCP/IP アドレス) を検査します。

### 名前

リスナー・プロセスは、チャンネル名が一致することを検査します。

### ALL

リスナー・プロセスは、キュー・マネージャー名の一致、通信アドレス、およびチャンネル名の一致を検査します。

デフォルトは AdoptNewMCACheck=NAME, ADDRESS, QM です。

## 関連概念

### 171 ページの『チャンネルの状態』

チャンネルは、いつでも多くの状態のうちの一つであることができます。いくつかの状態には、副状態もあります。チャンネルは、指定された状態から他の状態に移動することができます。

## IBM i QMErrorLog スタンザ (IBM i)

qm.ini ファイルの QMErrorLog スタンザを使用して、キュー・マネージャーのエラー・ログの操作と内容を調整します。

### V 9.0.4 ErrorLogSize=maxsize

キュー・マネージャーのエラー・ログがバックアップにコピーされる際のサイズを指定します。*maxsize* は、32768 から 2147483648 バイトの範囲でなければなりません。**ErrorLogSize** が指定されていないと、デフォルト値である 33554432 バイト (32 MB) が使用されます。

この属性を使用して、最大サイズを必要に応じて以前の最大サイズである 2 MB に戻すことができます。

**重要:** IBM MQ 9.0.4 から、**ErrorLogSize** 属性のデフォルト・サイズが大きくなりました。これは、IBM MQ 9.0.3 からの変更です。

MQMAXERRORLOGSIZE 環境変数を使用して、ログのサイズを設定することができます。

## ExcludeMessage= msgIds

キュー・マネージャーのエラー・ログに書き込まれないメッセージを指定します。*msgIds* には、次のメッセージ ID のコンマ区切りリストが含まれています。

7163 - ジョブ開始のメッセージ (IBM i のみ)

7234 - Number of messages loaded (いくつかのメッセージが読み込まれました)

8245

9001 - Channel program ended normally (チャンネル・プログラムが正常に終了しました)  
9002 - Channel program started (チャンネル・プログラムが開始されました)  
9202 - Remote host not available (リモート・ホストは使用できません)  
9208 - ホストからの受信エラー  
9209 - 接続の閉止  
9228 - チャンネル・レスポンスの開始不能  
9508 - キュー・マネージャーに接続不能  
9524 - Remote queue manager unavailable (リモート・キュー・マネージャーは使用できません)  
9528 - User requested closure of channel (ユーザーがチャンネルの閉止を要求しました)  
9558 - リモート・チャンネルは使用不可  
9776 - ユーザー ID によるチャンネルのブロック  
9777 - NOACCESS マップによるチャンネルのブロック  
9782 - アドレスによる接続のブロック  
9999 - Channel program ended abnormally (チャンネル・プログラムが異常終了しました)

### **SuppressMessage= msgIds**

指定した時間間隔内で 1 回だけキュー・マネージャーのエラー・ログに書き込まれるメッセージを指定します。時間間隔は、**SuppressInterval** で指定します。*msgIds* には、次のメッセージ ID のコンマ区切りリストが含まれています。

7163 - ジョブ開始のメッセージ (IBM i のみ)  
7234 - Number of messages loaded (いくつかのメッセージが読み込まれました)  
8245  
9001 - Channel program ended normally (チャンネル・プログラムが正常に終了しました)  
9002 - Channel program started (チャンネル・プログラムが開始されました)  
9202 - Remote host not available (リモート・ホストは使用できません)  
9208 - ホストからの受信エラー  
9209 - 接続の閉止  
9228 - チャンネル・レスポンスの開始不能  
9508 - キュー・マネージャーに接続不能  
9524 - Remote queue manager unavailable (リモート・キュー・マネージャーは使用できません)  
9528 - User requested closure of channel (ユーザーがチャンネルの閉止を要求しました)  
9558 - リモート・チャンネルは使用不可  
9776 - ユーザー ID によるチャンネルのブロック  
9777 - NOACCESS マップによるチャンネルのブロック  
9782 - アドレスによる接続のブロック  
9999 - Channel program ended abnormally (チャンネル・プログラムが異常終了しました)

同一のメッセージ ID が **SuppressMessage** と **ExcludeMessage** の両方で指定されていると、そのメッセージは除外されます。

### **SuppressInterval=length**

**SuppressMessage** に指定したメッセージを 1 回だけキュー・マネージャーのエラー・ログに書き込む時間間隔を秒単位で指定します。*length* は、1 から 86400 秒の範囲でなければなりません。**SuppressInterval** が指定されていないと、デフォルト値の 30 秒が使用されます。

## **IBM i TCP スタンザ (IBM i)**

キュー・マネージャーのプロパティ・ページ、または *qm.ini* ファイル内のスタンザを使用して、ネットワーク・プロトコル構成パラメーターを指定します。これらのスタンザは、チャンネルのデフォルトの属性を指定変更します。

注: 指定する必要があるのは、デフォルト値に対する変更を表す属性だけです。

### **TCP**

次の属性を指定できます。

**Port=1414|port\_number**

TCP/IP セッション用の 10 進表記のデフォルトのポート番号。IBM MQ 8.0 以降のデフォルト・ポート番号は 1414 です。

**KeepAlive=NO|YES**

KeepAlive 機能のオン/オフを切り替えます。KeepAlive=YES を指定すると、TCP/IP は、接続の相手側がまだ使用可能かどうかを定期的に検査します。有効でない場合、チャンネルは閉じた状態です。

**ListenerBacklog=number**

TCP/IP で受信するときに、未解決の接続要求の最大数を設定します。これは、TCP/IP ポート上で listener からの受け入れを待っている要求のバックログと考えることができます。IBM i のデフォルト・リスナー・バックログ値は 255 です。最大値は 512 です。バックログが値 512 に達すると、TCP/IP 接続は拒否され、チャンネルは開始できなくなります。

MCA チャンネルの場合は、チャンネルが RETRY 状態になり、後で接続が再試行されます。

クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC\_Q\_MGR\_NOT\_AVAILABLE を受け取り、後で接続を再試行することが必要になります。

ListenerBacklog 属性を使用すると、TCP/IP listener の場合のデフォルトの未解決要求数を指定変更することができます。

**Connect\_Timeout=number|0**

ソケットへの接続試行がタイムアウトになるまでの秒数。デフォルト値である 0 は、接続のタイムアウトがないことを指定します。

以下のプロパティのグループは、TCP/IP によって使用されるバッファのサイズを制御するために使用できます。値は、オペレーティング・システムの TCP/IP 層に直接渡されます。これらのプロパティを使用する際は、十分注意する必要があります。値が正しく設定されないと、TCP/IP のパフォーマンスに悪影響が及ぶ可能性があります。値が正しく設定されない場合にパフォーマンスに及ぶ影響について詳しくは、ご使用の環境の TCP/IP の資料を参照してください。値 0 は、IBM MQ によってバッファ・サイズが修正されるのではなく、オペレーティング・システムがバッファ・サイズを管理することを意味します。

**SndBuffSize=number|0**

チャンネルの送信側で使用される TCP/IP 送信バッファのサイズ (バイト)。このスタンザ値は、このチャンネル・タイプをさらに具体的に指定するスタンザ (例えば RcvSndBuffSize) でオーバーライドされる場合があります。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**RcvBuffSize=number|0**

チャンネルの受信側で使用される TCP/IP 受信バッファのサイズ (バイト)。このスタンザ値は、このチャンネル・タイプをさらに具体的に指定するスタンザ (例えば RcvRcvBuffSize) でオーバーライドされる場合があります。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**RcvSndBuffSize=number|0**

受信側チャンネルの送信側で使用される TCP/IP 送信バッファのサイズ (バイト)。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**RcvRcvBuffSize=number|0**

受信側チャンネルの受信側で使用される TCP/IP 受信バッファのサイズ (バイト)。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**SvrSndBuffSize=number|0**

クライアント接続チャンネルとサーバー接続チャンネルのサーバー側で使用される TCP/IP 送信バッファのサイズ (バイト)。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

### SvrRcvBuffSize=number|0

クライアント接続チャンネルとサーバー接続チャンネルのサーバー側で使用される TCP/IP 受信バッファのサイズ (バイト)。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

## IBM i V 9.0.5 チューニング・パラメーター・スタンザ (IBM i)

qm.ini ファイルの TuningParameters スタンザを使用して、キュー・マネージャーを調整するためのオプションを指定します。

### ImplSyncOpenOutput=value

**ImplSyncOpenOutput** は、書き込みのためにキューを開いておくアプリケーションの最小数です。その数に達すると、同期点以外で永続書き込みのための暗黙的同期点が有効になることがあります。

**ImplSyncOpenOutput** のデフォルト値は 2 です。

PUT 操作のためにキューを開いておくアプリケーションが 1 つだけの場合は、**ImplSyncOpenOutput** がオフに切り替えられます。

**ImplSyncOpenOutput=1** を指定すると、暗黙的同期点の可能性が常に存在することになります。

任意の正整数値を設定できます。暗黙的同期点を追加したくない場合は、ImplSyncOpenOutput=OFF を設定してください。

### 関連情報

暗黙の同期点

## IBM i IBM i 用の mqs.ini ファイルと qm.ini ファイルの例

### IBM MQ 構成ファイルの例

以下の例は、IBM i の mqs.ini ファイルを示しています:

```
#####  
#* Module Name: mqs.ini                                     *#  
#* Type       : IBM MQ Configuration File                 *#  
#* Function    : Define IBM MQ resources for the node     *#  
#*                                                    *#  
#####  
#* Notes      :                                           *#  
#* 1) This is an example IBM MQ configuration file        *#  
#*                                                    *#  
#####  
AllQueueManagers:  
#####  
#* The path to the qmgrs directory, within which queue manager data *#  
#* is stored                                               *#  
#####  
DefaultPrefix=/QIBM/UserData/mqm  
  
QueueManager:  
Name=saturn.queue.manager  
Prefix=/QIBM/UserData/mqm  
Library=QMSATURN.Q  
Directory=saturn!queue!manager  
  
QueueManager:  
Name=pluto.queue.manager  
Prefix=/QIBM/UserData/mqm  
Library=QMPLUTO.QU  
Directory=pluto!queue!manager  
  
DefaultQueueManager:  
Name=saturn.queue.manager
```

## キュー・マネージャー構成ファイルの例

**V 9.0.5** IBM MQ 9.0.5 以降、以下の例は、IBM i のキュー・マネージャー構成ファイルで属性のグループがどのように配置されるかを示しています。

```
#####  
#* Module Name: qm.ini                                     *#  
#* Type       : IBM MQ queue manager configuration file   *#  
# Function    : Define the configuration of a single queue manager *#  
#*          *#  
#####  
#* Notes      :                                           *#  
#* 1) This file defines the configuration of the queue manager *#  
#*          *#  
#####  
Log:  
LogPath=QMSATURN.Q  
LogReceiverSize=65536  
  
CHANNELS:  
MaxChannels = 20      ; Maximum number of channels allowed.  
                    ; Default is 100.  
MaxActiveChannels = 10 ; Maximum number of channels allowed to be  
                    ; active at any time. The default is the  
                    ; value of MaxChannels.  
  
TCP:                  ; TCP/IP entries.  
KeepAlive = Yes      ; Switch KeepAlive on.  
SvrSndBuffSize=20000 ; Size in bytes of the TCP/IP send buffer for each  
                    ; channel instance. Default is 32768.  
SvrRcvBuffSize=20000 ; Size in bytes of the TCP/IP receive buffer for each  
                    ; channel instance. Default is 32768.  
Connect_Timeout=10000 ; Number of seconds before an attempt to connect the  
                    ; channel instance times out. Default is zero (no timeout).  
  
QMErrorLog:  
ErrorLogSize = 262144  
ExcludeMessage = 7234  
SuppressMessage = 9001,9002,9202  
SuppressInterval = 30  
  
TuningParameters:  
ImplSyncOpenOutput=2
```

IBM MQ 9.0.4 以前と LTS の構成ファイルの例を以下に示します。

```
#####  
#* Module Name: qm.ini                                     *#  
#* Type       : IBM MQ queue manager configuration file   *#  
# Function    : Define the configuration of a single queue manager *#  
#*          *#  
#####  
#* Notes      :                                           *#  
#* 1) This file defines the configuration of the queue manager *#  
#*          *#  
#####  
Log:  
LogPath=QMSATURN.Q  
LogReceiverSize=65536  
  
CHANNELS:  
MaxChannels = 20      ; Maximum number of channels allowed.  
                    ; Default is 100.  
MaxActiveChannels = 10 ; Maximum number of channels allowed to be  
                    ; active at any time. The default is the  
                    ; value of MaxChannels.  
  
TCP:                  ; TCP/IP entries.  
KeepAlive = Yes      ; Switch KeepAlive on.  
SvrSndBuffSize=20000 ; Size in bytes of the TCP/IP send buffer for each  
                    ; channel instance. Default is 32768.  
SvrRcvBuffSize=20000 ; Size in bytes of the TCP/IP receive buffer for each  
                    ; channel instance. Default is 32768.  
Connect_Timeout=10000 ; Number of seconds before an attempt to connect the  
                    ; channel instance times out. Default is zero (no timeout).
```

```
QMErrorLog:
ErrorLogSize = 262144
ExcludeMessage = 7234
SuppressMessage = 9001,9002,9202
SuppressInterval = 30
```

#### 注:

1. ノード上の IBM MQ は、キュー・マネージャー用とジャーナル用にデフォルト・ロケーションを使用します。
2. キュー・マネージャー saturn.queue.manager は、ノードでのデフォルト・キュー・マネージャーです。このキュー・マネージャーに関連したファイルのディレクトリーは、ファイル・システムで有効なファイル名に自動的に変換されています。
3. IBM MQ 構成ファイルは、キュー・マネージャーに関連するデータを見つけるために使用されるので、構成ファイルが存在しなかったり、正しくなかったりすると、一部または全部の IBM MQ コマンドが失敗します。また、アプリケーションは、IBM MQ 構成ファイルに定義されていないキュー・マネージャーには接続できません。

## IBM MQ 構成情報を変更するための属性

IBM MQ for Windows システムおよび IBM MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、IBM MQ Explorer を使用して構成情報を変更します。他のシステムでは、mqqs.ini 構成ファイルを編集して構成情報を変更します。

特定のコンポーネントの属性については、以下のサブトピックを参照してください。

### 関連概念

#### 5 ページの『IBM MQ の構成』

1 台以上のコンピューターにキュー・マネージャーを 1 つ以上作成し、それらをビジネス・データが含まれるメッセージを処理するように開発システム、テスト・システム、実動システム上で構成します。

### 関連タスク

#### 85 ページの『IBM MQ およびキュー・マネージャー構成情報の変更』

IBM MQ または個々のキュー・マネージャーの動作を、インストール済み環境の要件に合わせて変更できます。

#### 112 ページの『キュー・マネージャー構成情報の変更』

個々のキュー・マネージャーの構成を変更するために使用できる属性によって、IBM MQ の設定がオーバーライドされます。

### 関連情報

#### 計画

#### 管理 IBM MQ

## すべてのキュー・マネージャー

IBM MQ Explorer の General プロパティ・ページおよび Extended IBM MQ プロパティ・ページ、または mqqs.ini ファイルの AllQueueManagers スタンザを使用して、すべてのキュー・マネージャーに関する以下の情報を指定します。

### DefaultPrefix=directory\_name

この属性には、キュー・マネージャー・データが保管されている qmgrs ディレクトリーへのパスを指定します。

キュー・マネージャーのデフォルト・プレフィックスを変更した場合、インストール時に作成されたディレクトリー構造を複製してください。

特に、qmgers 構造を作成する必要があります。デフォルト接頭部を変更する前に IBM MQ を停止してください。IBM MQ の再始動は、構造を新しい位置に移し、デフォルト接頭部を変更した後に行ってください。

注: UNIX and Linux システムの場合は /var/mqm/errors ディレクトリー、Windows システムの場合は \errors ディレクトリーを削除しないでください。

デフォルト接頭部を変更する代わりに、環境変数 MQSPREFIX を使用して、crtmqm コマンドの **DefaultPrefix** を指定変更することもできます。

オペレーティング・システムの制約があるため、指定するパスをできるだけ短くして、パス長とすべてのキュー・マネージャー名の長さの合計が最大 70 文字になるようにしてください。

### Multi **ConvEBCDICNewline=NL\_TO\_LF|TABLE|ISO**

EBCDIC コード・ページには、ASCII コード・ページではサポートされない改行 (NL) 文字が含まれています (ただし、ISO 仕様の ASCII には、この改行文字と同等の文字が含まれているものもいくつかあります)。**ConvEBCDICNewline** 属性を使用して、IBM MQ が EBCDIC の NL 文字を ASCII フォーマットに変換する方法を指定します。

**IBM i** IBM MQ for IBM i では CCSID 1253 は ISO CCSID と見なされ、NL\_TO\_LF は ISO と ASCII の両方の変換に影響を及ぼします。

**z/OS** **ConvEBCDICNewline** 属性は、z/OS では使用できません。z/OS での動作は、ConvEBCDICNewline=TABLE と同等です。他のプラットフォームでのデフォルトは異なる場合がありますので注意してください。

### **NL\_TO\_LF**

EBCDIC から ASCII へのすべての変換について、EBCDIC NL 文字 (X'15') を ASCII 改行 (LF) 文字 (X'0A') に変換します。

NL\_TO\_LF はデフォルトです。

### 表

EBCDIC から ASCII への変換に使用している変換テーブルに従って、EBCDIC の NL 文字を変換します。

このタイプの変換の結果は、プラットフォームと言語によって異なる場合があります。同じプラットフォーム上でも、使用する CCSID が異なれば動作が違ってくる場合があります。

### **ISO**

次のものを変換します。

- ISO CCSID (TABLE メソッドを使用)
- その他のすべての CCSID (NL\_TO\_CF メソッドを使用)

使用できる ISO CCSID は、[107 ページの表 11](#) に示されています。

CCSID	コード・セット
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

ASCII CCSID が ISO サブセットの 1 つではない場合は、デフォルトにより、**ConvEBCDICNewline** は NL\_TO\_LF に設定されます。

**V 9.0.0.6** IBM MQ 9.0.0 Fix Pack 6 以降では、**ConvEBCDICNewline** スタンザ属性の代わりに **AMQ\_CONVBCDICNEWLIN** 環境変数を使用できます。例えば、mqs.ini ファイルを使用できない状

況で、クライアント・サイドで **ConvEBCDICNewline** 機能を提供することができます。この環境変数では、**ConvEBCDICNewline** 属性と同じ値 (NL\_TO\_LF、TABLE、ISO) を設定できます。スタanzas属性と環境変数の両方が設定されている場合は、スタanzas属性の方が優先されます。

## デフォルト・キュー・マネージャー

IBM MQ Explorer の General IBM MQ プロパティ・ページ、または mqs.ini ファイルの DefaultQueueManager stanzasを使用して、デフォルトのキュー・マネージャーを指定します。

### Name=default\_queue\_manager

デフォルト・キュー・マネージャーは、キュー・マネージャー名が明示的に指定されていないすべてのコマンドを処理します。DefaultQueueManager 属性は、新規のデフォルト・キュー・マネージャーを作成すると、自動的に更新されます。誤って新規のデフォルト・キュー・マネージャー作成した場合、それを元のプログラムに戻すときは、DefaultQueueManager 属性を手動で変更してください。

## 出口のプロパティ

IBM MQ Explorer の Extended IBM MQ プロパティ・ページ、または mqs.ini ファイルの ExitProperties stanzasを使用して、キュー・マネージャー出口プログラムで使用される構成オプションを指定します。

### CLWLMode=SAFE|FAST

クラスター・ワークロード (CLWL) 出口では、MQI 呼び出し (例えば、MQOPEN、MQPUT) に応じて、クラスター内のどのクラスター・キューをオープンするかを指定できます。CLWL 出口は、CLWLMode 属性に指定する値に応じて、FAST モードまたは SAFE モードで実行されます。CLWLMode 属性を省略する場合は、クラスター・ワークロード出口は SAFE モードで実行されます。

#### SAFE

CLWL 出口をキュー・マネージャーから独立したプロセスで実行します。これがデフォルトです。

SAFE モードでの実行中にユーザー作成の CLWL に問題が起きた場合は、次のような結果が生じます。

- CLWL サーバー・プロセス (amqzlw0) は失敗します。
- キュー・マネージャーは CLWL サーバー・プロセスを再始動します。
- エラーがエラー・ログに記録されます。進行中の MQI 呼び出しがある場合は、戻りコードが戻されます。

キュー・マネージャーの健全性は維持されます。

注: CLWL 出口を独立したプロセスで実行すると、パフォーマンスが影響を受けることがあります。

#### FAST

クラスター出口は、キュー・マネージャー・プロセスでインラインで実行します。

このオプションを指定すると、SAFE モードで実行した場合のプロセスの切り替えコストを回避できるのでパフォーマンスは向上しますが、代わりにキュー・マネージャーの健全性が保証されないこととなります。CLWL 出口を FAST モードで実行するのは、CLWL 出口にまったく問題がないという確信があり、パフォーマンスが特に重要な要素である場合のみにしてください。

CLWL 出口を FAST モードで実行しているときに問題が発生すると、キュー・マネージャーは失敗し、キュー・マネージャーの健全性が損なわれる恐れがあります。

## IBM MQ のログのデフォルト

IBM MQ Explorer の Default log settings IBM MQ プロパティ・ページ、または mqs.ini ファイルの LogDefaults stanzasを使用して、すべてのキュー・マネージャーのログ・デフォルトに関する情報を指定します。

スタanzasが存在しない場合、MQ デフォルトが使用されます。ログ属性は、ユーザーがキュー・マネージャーを作成するときにデフォルト値として使用されますが、crtmqm コマンドにログ属性を指定することにより、指定変更することができます。このコマンドの詳細については、[crtmqm](#) を参照してください。

キュー・マネージャーが作成された後は、そのキュー・マネージャーのログ属性は、[116 ページの『キュー・マネージャーのログ』](#)に説明されている設定値から取られます。

デフォルト接頭部 ([106 ページの『すべてのキュー・マネージャー』](#)で指定) および特定のキュー・マネージャーに指定されたログ・パス ([116 ページの『キュー・マネージャーのログ』](#)で指定) により、キュー・マネージャーとそのログを別々の物理ドライブに置くことができます。デフォルトではこの2つは同じドライブに入りますが、別々のドライブを使用することをお勧めします。

ログ・サイズの計算方法については、[535 ページの『ログのサイズの計算』](#)を参照してください。

**注:** 次のパラメーター・リストに示す制限は、IBM MQ により設定される制限です。オペレーティング・システムの制限により、最大可能ログ・サイズがさらに減少することもあります。

#### **LogPrimaryFiles=3|2-254 (Windows)|2-510 (UNIX and Linux システム)**

キュー・マネージャーの作成時に割り振られるログ・ファイル。

1 次ログ・ファイルの最小数は 2 であり、最大数は Windows では 254、UNIX および Linux では 510 です。デフォルトは 3 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 3 以上にする必要があり、かつ Windows では 255 以下、UNIX および Linux では 511 以下にする必要があります。

この値は、キュー・マネージャーの作成時または開始時に調べられます。キュー・マネージャーが作成された後に、この値を変更することができます。ただし、この変更された値は、キュー・マネージャーが再始動されるまで有効にならないので、効果はただちに現れません。

#### **LogSecondaryFiles= 2|1-253 (Windows)|1-509 (UNIX および Linux)**

1 次ファイルが足りなくなったときに割り振られるログ・ファイル。

2 次ログ・ファイルの最小数は 1 であり、最大数は Windows では 253、UNIX および Linux では 509 です。デフォルトの数は 2 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 3 以上にする必要があり、かつ Windows では 255 以下、UNIX および Linux では 511 以下にする必要があります。

この値は、キュー・マネージャーの始動時に検査されます。この値は変更することができます。ただし、変更された値は、キュー・マネージャーが再始動されるまでは有効にはなりません。有効になった場合でも効果がただちに現れるとは限りません。

#### **LogFilePages=number**

ログ・データは、ログ・ファイルと呼ばれる一連のファイルに保持されます。ログ・ファイル・サイズは、4 KB ページ単位で指定します。

ログ・ファイルのデフォルトのページ数は 4096 です。これは、16 MB のログ・ファイル・サイズと同じです。

UNIX と Linux では、ログ・ファイルの最小ページ数は 64 です。Windows では、ログ・ファイルの最小ページ数は 32 です。最大ページ数はどちらも 65 535 です。

**注:** キュー・マネージャーの作成時に指定したログ・ファイルのサイズを、個々のキュー・マネージャーについて変更することはできません。

#### **LogType=CIRCULAR|LINEAR**

使用するログのタイプ。デフォルトは CIRCULAR です。

##### **CIRCULAR**

システムの停止時に進行中だったトランザクションをロールバックするログを使用して、再始動リカバリーを開始します。

循環ロギングの詳細については、[530 ページの『ログのタイプ』](#)を参照してください。

##### **LINEAR**

再始動リカバリーと、メディアまたは順方向リカバリー (ログの内容を再生することにより、消失または損傷したデータを作成する) の両方を可能にします。

リニア・ロギングの詳細については、[530 ページの『ログのタイプ』](#)を参照してください。

デフォルトを変更する場合は、LogType 属性を編集するか、crtmqm コマンドでリニア・ロギングを指定します。

**V 9.0.4** IBM MQ 9.0.4 から、キュー・マネージャーの作成後にロギング方式を変更できるようになりました。詳しくは、[migmqlog](#) を参照してください。

### LogBufferPages=0|0-4096

書き込み用としてバッファ・レコードに割り振るメモリーの量。バッファのサイズは 4 KB ページ単位で指定します。

バッファの最小ページ数は 18 であり、最大ページ数は 4096 です。バッファが大きいとスループットが高くなりますが、これは特に大きなメッセージに適用されます。

0 (デフォルト) を指定すると、サイズはキュー・マネージャーによって選択されます。IBM WebSphere MQ 7.1 では、これは 512 (2048 KB) になります。

1 から 17 の範囲の数を指定すると、キュー・マネージャーはデフォルトで 18 (72 KB) になります。18 から 4096 の範囲の数を指定すると、キュー・マネージャーは指定された数を使用して割り振るメモリーを設定します。

### LogDefaultPath=directory\_name

キュー・マネージャーのログ・ファイルが置かれるディレクトリー。ディレクトリーは、キュー・マネージャーが書き込みを行える ローカル装置に置くか、できればメッセージ・キューとは異なるドライブに置きます。異なるドライブを指定すると、システム障害時の保護機能が加わります。

デフォルトは次のとおりです。

- *DefaultPrefix*\log for IBM MQ for Windows。ここで、*DefaultPrefix* は、All Queue Managers IBM MQ プロパティー・ページの *DefaultPrefix* 属性で指定された値です。この値は、インストール時に設定されます。
- /var/mqm/log (IBM MQ for UNIX および Linux システムの場合)

代わりに、crtmqm コマンドで -ld フラグを使用して、ディレクトリーの名前を指定することもできます。キュー・マネージャーが作成されると、キュー・マネージャー・ディレクトリーの下にもう 1 つディレクトリーが作成され、ログ・ファイルの保存に使用されます。このディレクトリーの名前はキュー・マネージャー名に基づいて付けられます。これによってログ・ファイル・パスが確実に固有になり、ディレクトリー名の長さに関する制限にも抵触しないことになります。

crtmqm コマンドで -ld を指定しなかった場合は、mqs.ini ファイル内の LogDefaultPath 属性の値が使用されます。

複数のキュー・マネージャーで別々のログ・ディレクトリーが使用されるように、キュー・マネージャー名がディレクトリー名に付加されます。

キュー・マネージャーが作成されると、構成情報内のログ属性の中に LogPath 値が作成されて、キュー・マネージャーのログに完全ディレクトリー名が与えられます。この値は、キュー・マネージャーの開始時または削除時に、ログを見つけるために使用されます。

### LogWriteIntegrity=SingleWrite|DoubleWrite|TripleWrite

高信頼ログ・レコードを書き込むためにロガーが使用する方式。

#### TripleWrite

これはデフォルトの方式です。

なお、**DoubleWrite** を選択可能ですが、そのようにした場合、システムでは **TripleWrite** と解釈されます。

#### SingleWrite

**SingleWrite** は、IBM MQ リカバリー・ログをホスティングしているファイル・システムおよびデバイスが、最小単位 4KB の書き込みを明示的に保証している場合にのみ使用してください。

すなわち、何らかの原因で 4KB ページの書き込みが失敗した場合、変更前イメージまたは変更後イメージの 2 つの状態しかありません。中間の状態は考えられません。

注: 持続するワークロードにおける並行性が十分であるなら、デフォルト値 **TripleWrite** 以外の値に設定することで得られる利点はほとんどありません。

## Advanced Configuration and Power Interface (ACPI)

IBM MQ Explorer の ACPI IBM MQ プロパティ・ページを使用して、システムが中断要求を受信したときの IBM MQ の動作を指定します。

Windows は、Advanced Configuration and Power Interface (ACPI) 規格をサポートしています。この規格により、ACPI 対応のハードウェアを使用する Windows ユーザーは、システムがサスペンド・モードに入るとき、サスペンド・モードからレジュームするときに、チャンネルを停止および再始動することができます。

ACPI IBM MQ のプロパティ・ページで指定されている設定は、アラート・モニターの実行時にのみ適用されることに注意してください。アラート・モニター・アイコンは、アラート・モニターが実行中である場合にタスクバーに表示されます。

### DoDialog=Y | N

サスペンド要求があったときにダイアログを表示します。

### DenySuspend=Y | N

サスペンド要求を拒否します。これは、DoDialog=N の場合、または DoDialog=Y を指定していても、ノートブックのふたが閉じられているなどの理由でダイアログを表示できない場合に使用されます。

### CheckChannelsRunning=Y | N

実行中のチャンネルがないか確認します。この結果によって、他の設定の結果が分かります。

次の表は、これらのパラメーターのそれぞれの組み合わせの効果を示しています。

DoDialog	DenySuspend	CheckChannels Running	アクション
N	N	N	サスペンド要求を受け入れます。
N	N	Y	サスペンド要求を受け入れます。
N	Y	N	サスペンド要求を拒否します。
N	Y	Y	チャンネルが実行中の場合はサスペンド要求を拒否し、チャンネルが実行中でない場合はサスペンド要求を受け入れます。
Y	N	N	ダイアログを表示します (注を参照。サスペンド要求を受け入れます)。これがデフォルトです。
Y	N	Y	実行中のチャンネルがない場合はサスペンド要求を受け入れ、チャンネルが実行中の場合はダイアログを表示します (注を参照。サスペンド要求を受け入れます)。
Y	Y	N	ダイアログを表示します (注を参照。サスペンド要求を拒否します)。
Y	Y	Y	実行中のチャンネルがない場合はサスペンド要求を受け入れ、チャンネルが実行中の場合はダイアログを表示します (注を参照。サスペンド要求を拒否します)。

注: アクションがダイアログを表示するようになっている場合で、ダイアログを表示できない場合 (ノートブックのふたが閉じられているなどの理由で) は、サスペンド要求を受け入れるか拒否するかを判別するために、DenySuspend オプションが使用されます。

## API 出口

IBM MQ Explorer または amqmdain コマンドを使用して、API 出口の項目を変更します。

IBM MQ Explorer の Exits IBM MQ プロパティ・ページ、または mqs.ini ファイルの ApiExitTemplate および ApiExitCommon スタンザを使用して、すべてのキュー・マネージャーの API 出口ルーチンを識別します。Windows システムでは、API 出口の項目を変更するために、amqmdain コマンドを使用することもできます。(個々のキュー・マネージャーの API 出口ルーチンを識別するには、89 ページの『キュー・マネージャー構成ファイル qm.ini』に説明されているとおりに、ApiExitLocal スタンザを使用します。)

これらのスタンザの属性の詳細な説明については、[API 出口の構成](#)を参照してください。

## キュー・マネージャー

キュー・マネージャーごとに QueueManager スタンザが 1 つずつあります。このスタンザを使用して、キュー・マネージャーのディレクトリーの位置を指定します。

Windows、UNIX and Linux システムでは、キュー・マネージャーごとに、QueueManager スタンザが 1 つずつあります。これらの属性は、キュー・マネージャーの名前と、そのキュー・マネージャーに関連したファイルが含まれているディレクトリーの名前を指定します。ディレクトリーの名前はキュー・マネージャー名に基づいていますが、キュー・マネージャー名が有効なファイル名でない場合は、変換されます。名前変換について詳しくは、[IBM MQ ファイル名についての理解](#)を参照してください。

### Name=queue\_manager\_name

キュー・マネージャーの名前。

### Prefix=prefix

キュー・マネージャーのファイルが保管される場所。デフォルトでは、この値は「All Queue Managers (すべてのキュー・マネージャー)」情報の DefaultPrefix 属性に指定されている値と同じです。

### Directory=name

キュー・マネージャー・ファイルが保管されている prefix\QMGRS ディレクトリーの下にあるサブディレクトリーの名前。この名前はキュー・マネージャー名に基づいていますが、重複する名前がある場合、またはキュー・マネージャー名が無効なファイル名の場合は、変換される場合があります。

### DataPath=path

キュー・マネージャーの作成時に指定された明示的なデータ・パス。このパスは、キュー・マネージャー・データのパスとして設定された Prefix と Directory をオーバーライドします。

### InstallationName=name

このキュー・マネージャーに関連付けられている IBM MQ インストールの名前。このキュー・マネージャーと対話するときには、このインストールからコマンドを使用する必要があります。

InstallationName の値が存在しない場合は、キュー・マネージャーが IBM WebSphere MQ 7.1 より前の製品インストールに関連付けられます。

## 関連タスク

395 ページの『[キュー・マネージャーとインストールの関連付け](#)』

キュー・マネージャーを作成すると、そのキュー・マネージャーは、crtmqm コマンドを実行したインストールに自動的に関連付けられます。UNIX, Linux, and Windows では、setmqm コマンドを使用して、キュー・マネージャーに関連付けられているインストールを変更できます。

## キュー・マネージャー構成情報の変更

個々のキュー・マネージャーの構成を変更するために使用できる属性によって、IBM MQ の設定がオーバーライドされます。

## このタスクについて

UNIX and Linux システムでは、qm.ini 構成ファイルを編集してキュー・マネージャーの構成情報を変更します。qm.ini でスタンザを定義しているとき、各項目を新しい行で始める必要はありません。コメントを入れる場合は、セミコロン (;) またはハッシュ文字 (#) を使用できます。

Windows システムと Linux x86-64 システムでは、IBM MQ Explorer を使用して一部の構成情報を変更できます。ただし、インストール可能サービスとそのコンポーネントを変更すると大きな影響が生じるので、IBM MQ Explorer では、インストール可能サービスは読み取り専用になっています。したがって、インス

ツール可能サービスを変更する場合は、Windows では **regedit** を使用する必要があり、UNIX and Linux では `qm.ini` ファイルを編集する必要があります。

## 手順

- キュー・マネージャー構成情報の変更について詳しくは、以下のサブトピックを参照してください。

### 関連概念

[5 ページの『IBM MQ の構成』](#)

1 台以上のコンピューターにキュー・マネージャーを 1 つ以上作成し、それらをビジネス・データが含まれるメッセージを処理するように開発システム、テスト・システム、実動システム上で構成します。

### 関連タスク

[85 ページの『IBM MQ およびキュー・マネージャー構成情報の変更』](#)

IBM MQ または個々のキュー・マネージャーの動作を、インストール済み環境の要件に合わせて変更できます。

### 関連資料

[106 ページの『IBM MQ 構成情報を変更するための属性』](#)

IBM MQ for Windows システムおよび IBM MQ for Linux (x86 および x86-64 プラットフォーム) システムでは、IBM MQ Explorer を使用して構成情報を変更します。他のシステムでは、`mqs.ini` 構成ファイルを編集して構成情報を変更します。

### 関連情報

[計画](#)

[管理 IBM MQ](#)

## Windows アクセス・モード

**Access Mode** は、Windows サーバーにのみ適用されます。AccessMode スタンザは、`crtmqm` コマンドの `-a [r]` オプションで設定します。キュー・マネージャーの作成後には、AccessMode スタンザを変更しないでください。

`crtmqm` コマンドのアクセス・グループ (`-a [r]`) オプションを使用して、Windows セキュリティー・グループを指定します。このグループのメンバーは、すべてのキュー・マネージャー・データ・ファイルへの全アクセス権限を付与されます。このグループは、使用する構文に応じて、ローカル・グループかグローバル・グループのいずれかになります。グループ名の有効な構文は次のとおりです。

*LocalGroup*

*Domain name &#xa5; GlobalGroup name*

*GlobalGroup name@Domain name*

`-a [r]` オプションを指定して `crtmqm` コマンドを実行するには、まず追加のアクセス・グループを定義しておく必要があります。

`-a` の代わりに `-ar` を使用してグループを指定すると、ローカル `mqm` グループはキュー・マネージャー・データ・ファイルへのアクセス権を付与されません。キュー・マネージャー・データ・ファイルをホストするファイル・システムが、ローカルに定義されたグループのアクセス制御項目をサポートしていない場合は、このオプションを使用してください。

このグループは通常はグローバル・セキュリティ・グループです。グローバル・セキュリティ・グループは、複数インスタンス・キュー・マネージャーに、キュー・マネージャーの共有データ・フォルダーと共有ログ・フォルダーに対するアクセス権を付与するために使用されます。このような追加のセキュリティ・アクセス・グループを使用すれば、キュー・マネージャーのデータ・ファイルとログ・ファイルが含まれているフォルダーや共有フォルダーに対する読み取り/書き込み権限を設定できます。

キュー・マネージャーのデータとログが含まれているフォルダーに対する権限を設定するために、`mqm` という名前のローカル・グループを使用することもできますが、追加のセキュリティ・アクセス・グループは、その代替手段になります。ローカル・グループ `mqm` の場合とは異なり、追加のセキュリティ・アクセス・グループは、ローカル・グループでもグローバル・グループでもかまいません。複数インスタン

ス・キュー・マネージャーが使用するデータ・ファイルとログ・ファイルが含まれている共有フォルダーに対する権限を設定する場合は、グローバル・グループを使用する必要があります。

Windows オペレーティング・システムは、キュー・マネージャーのデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限を検査します。検査の対象になるのは、キュー・マネージャーのプロセスを実行しているユーザー ID の権限です。検査対象になるユーザー ID は、キュー・マネージャーをサービスとして開始したか、それとも対話式に開始したかによって異なります。キュー・マネージャーをサービスとして開始した場合は、「準備 IBM MQ」ウィザードで構成したユーザー ID が Windows システムによって検査されます。キュー・マネージャーを対話式に開始した場合、Windows システムによって検査されるユーザー ID は、**strmqm** コマンドを実行したユーザー ID です。

キュー・マネージャーを開始するユーザー ID は、ローカル mqm グループのメンバーでなければなりません。そのユーザー ID が追加のセキュリティー・アクセス・グループのメンバーになっていれば、キュー・マネージャーで、そのグループに基づいて権限が与えられているファイルを読み書きすることが可能になります。

**制約事項:** 追加のセキュリティー・アクセス・グループを指定できるのは、Windows オペレーティング・システムだけです。他のオペレーティング・システムで追加のセキュリティー・アクセス・グループを指定すると、**crtmqm** コマンドでエラーが返されます。

### 関連概念

[473 ページの『Windows でキュー・マネージャーのデータとログの非共有ディレクトリーおよびファイルを保護する操作』](#)

[469 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 \(Windows\)』](#)

### 関連タスク

[443 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 \(Windows\)』](#)

### 関連情報

crtmqm

## ULW インストール可能サービスの構成

インストール可能サービスを変更する場合は、Windows では **regedit** を使用し、UNIX and Linux では **qm.ini** ファイルの Service スタンザを使用します。

**注:** インストール可能サービスとそのコンポーネントを変更することには、大きな影響があります。このため IBM MQ Explorer では、インストール可能サービスを読み取り専用にしています。

Windows システムでインストール可能サービスを変更するには **regedit** を使用し、UNIX and Linux システムの場合は **qm.ini** ファイルの Service スタンザを使用します。1つのサービス内の各コンポーネントについて、そのコンポーネントのコードを含むモジュールの名前とパスも指定する必要があります。UNIX and Linux システムでは、このために ServiceComponent スタンザを使用します。

### **Name=AuthorizationService|NameService**

必要なサービスの名前。

#### **AuthorizationService**

IBM MQ の場合は、Authorization Service コンポーネントはオブジェクト権限マネージャー (OAM) と呼ばれています。キュー・マネージャーの作成時に、AuthorizationService スタンザとそれに関連した ServiceComponent スタンザが自動的に追加されます。その他の ServiceComponent スタンザは、手動で追加します。

#### **NameService**

デフォルトではネーム・サービスが提供されていません。ネーム・サービスが必要な場合は、NameService スタンザを手動で追加する必要があります。

### **EntryPoints=number-of-entries**

このサービス用として定義するエントリー・ポイントの数。

これには、初期化エントリー・ポイントと終了エントリー・ポイントも含まれています。

## Windows SecurityPolicy= Default|NTSIDsRequired

Windows システムの場合、SecurityPolicy 属性が適用されるのは、指定されているサービスがデフォルトの許可サービス、つまり OAM である場合のみです。SecurityPolicy 属性は、各キュー・マネージャーのセキュリティ・ポリシーを指定するために使用できます。

指定できる値は以下のとおりです。

### デフォルト

有効にするデフォルトのセキュリティ・ポリシーを使用します。ある特定のユーザー ID の Windows セキュリティー ID (NT SID) が OAM に渡されなかった場合は、関連のセキュリティ・データベース内の検索によって、該当する SID が取得されます。

### NTSIDsRequired

セキュリティ検査を行うときに、NT SID を OAM に渡します。

詳しくは、[Windows セキュリティー ID \(SID\)](#) を参照してください。

許可サービス・スタンザーの構成: [Windows システム](#) も参照してください。

## Linux UNIX SecurityPolicy=user|group|default

UNIX and Linux システムの場合、この値で、キュー・マネージャーがユーザー・ベースとグループ・ベースのどちらの許可を使用するかが指定されます。値には大/小文字の区別はありません。

この属性を含めない場合は、グループ・ベース許可を使用する default が使用されます。変更内容を有効にするには、キュー・マネージャーを再始動します。[許可サービス・スタンザーの構成: UNIX and Linux システム](#) も参照してください。

### SharedBindingsUserId=user-type

SharedBindingsUserId 属性が適用されるのは、指定されているサービスがデフォルトの許可サービス、つまり OAM である場合のみです。SharedBindingsUserId 属性は、共有バインディングが関係する場合にのみ使用されます。この値により、MQZ\_AUTHENTICATE\_USER 関数から、IdentityContext 構造体の UserIdentifier フィールドを有効ユーザー ID と実ユーザー ID のどちらかに指定できます。

MQZ\_AUTHENTICATE\_USER 関数については、[MQZ\\_AUTHENTICATE\\_USER - ユーザーの認証](#)を参照してください。

指定できる値は以下のとおりです。

### デフォルト

UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

### Real

UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

### Effective

UserIdentifier フィールドの値は、有効ユーザー ID として設定されます。

### FastpathBindingsUserId=user-type

FastpathBindingsUserId 属性が適用されるのは、指定されているサービスがデフォルトの許可サービス、つまり OAM である場合のみです。FastpathBindingsUserId 属性は、ファースト・パス・バインディングが関係する場合にのみ使用されます。この値により、MQZ\_AUTHENTICATE\_USER 関数から、IdentityContext 構造体の UserIdentifier フィールドを有効ユーザー ID と実ユーザー ID のどちらかに指定できます。

MQZ\_AUTHENTICATE\_USER 関数については、[MQZ\\_AUTHENTICATE\\_USER - ユーザーの認証](#)を参照してください。

指定できる値は以下のとおりです。

### デフォルト

UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

### Real

UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

### Effective

UserIdentifier フィールドの値は、有効ユーザー ID として設定されます。

## IsolatedBindingsUserId=user-type

**IsolatedBindingsUserId** 属性が適用されるのは、指定されているサービスがデフォルトの許可サービス、つまり OAM である場合のみです。**IsolatedBindingsUserId** 属性は、分離されたバインディングが関係する場合にのみ使用されます。この値により、MQZ\_AUTHENTICATE\_USER 関数から、IdentityContext 構造体の UserIdentifier フィールドを有効ユーザー ID と実ユーザー ID のどちらかに指定できます。

MQZ\_AUTHENTICATE\_USER 関数については、[MQZ\\_AUTHENTICATE\\_USER - ユーザーの認証](#)を参照してください。

指定できる値は以下のとおりです。

### デフォルト

UserIdentifier フィールドの値は、有効ユーザー ID として設定されます。

### Real

UserIdentifier フィールドの値は、実ユーザー ID として設定されます。

### Effective

UserIdentifier フィールドの値は、有効ユーザー ID として設定されます。

インストール可能なサービスとコンポーネントの詳細については、[UNIX, Linux, and Windows 用のインストール可能なサービスとコンポーネント](#)を参照してください。

セキュリティー・サービス全般について詳しくは、[UNIX and Linux システムでのセキュリティーのセットアップ](#)を参照してください。

## 関連情報

[インストール可能なサービスの参照情報](#)

## サービス・コンポーネント

新しいインストール可能なサービスを追加する場合は、サービス・コンポーネント情報を指定する必要があります。Windows システムでは **regedit** を使用し、UNIX and Linux システムでは **qm.ini** ファイルの **ServiceComponent** スタンザを使用します。デフォルトでは、許可サービス・スタンザが事前に提供され、関連するコンポーネントである OAM がアクティブになっています。

以下のように、サービス・コンポーネントを指定します。

### Service=service\_name

必要なサービスの名前。これは、サービス構成情報の Name 属性に指定されている値に一致していることが必要です。

### Name=component\_name

サービス・コンポーネントの記述名を指定します。これは、固有名でなければならず、しかも IBM MQ オブジェクトの名前 (例えばキュー名) として有効な文字のみが使用されていなければなりません。この名前は、サービスで生成されるオペレーター・メッセージに表示されます。この名前は、企業の商標やそれに類似した識別性の高いストリングで始めることをお勧めします。

### Module=module\_name

このコンポーネントのコードを含むモジュールの名前。これは、絶対パス名でなければなりません。

### ComponentDataSize=size

各呼び出しでコンポーネントに渡されるコンポーネント・データ域のバイト単位のサイズ。コンポーネント・データが必要ない場合は、ゼロを指定します。

インストール可能なサービスとコンポーネントについて詳しくは、[UNIX、Linux および Windows 用のインストール可能なサービスとコンポーネント](#)を参照してください。

## キュー・マネージャーのログ

IBM MQ Explorer の Log キュー・マネージャー・プロパティー・ページ、または **qm.ini** ファイルの Log スタンザを使用して、キュー・マネージャーのログギングに関する情報を指定します。

デフォルトでは、これらの設定は、キュー・マネージャーのデフォルト・ログ設定に指定されている設定から継承されます (108 ページの『[IBM MQ のログのデフォルト](#)』を参照してください)。これらの設定を変更するのは、このキュー・マネージャーを別の方法で構成する場合のみにしてください。

ログ・サイズの計算方法については、535 ページの『[ログのサイズの計算](#)』を参照してください。

注: 次のパラメーター・リストに示す制限は、IBM MQ により設定されます。オペレーティング・システムの制限により、最大可能ログ・サイズがさらに減少することもあります。

### **LogPrimaryFiles=3|2-254 (Windows)|2-510 (UNIX and Linux システム)**

キュー・マネージャーの作成時に割り振られるログ・ファイル。

1 次ログ・ファイルの最小数は 2 であり、最大数は Windows では 254、UNIX and Linux システムでは 510 です。デフォルトは 3 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 3 以上にする必要があり、かつ Windows では 255 以下、UNIX and Linux システムでは 511 以下にする必要があります。

この値は、キュー・マネージャーの作成時または開始時に調べられます。キュー・マネージャーが作成された後に、この値を変更することができます。ただし、この変更された値は、キュー・マネージャーが再始動されるまで有効にならないので、効果はただちに現れません。

### **LogSecondaryFiles=2|1-253 (Windows)|1-509 (UNIX and Linux システム)**

1 次ファイルが足りなくなったときに割り振られるログ・ファイル。

2 次ログ・ファイルの最小数は 1 であり、最大数は Windows では 253、UNIX and Linux システムでは 509 です。デフォルトの数は 2 です。

1 次ログ・ファイルと 2 次ログ・ファイルの合計数は 3 以上にする必要があり、かつ Windows では 255 以下、UNIX and Linux システムでは 511 以下にする必要があります。

この値は、キュー・マネージャーの始動時に検査されます。この値は変更することができます。ただし、変更された値は、キュー・マネージャーが再始動されるまでは有効にはなりません。有効になった場合でも効果がただちに現れるとは限りません。

### **LogFilePages=number**

ログ・データは、ログ・ファイルと呼ばれる一連のファイルに保持されます。ログ・ファイル・サイズは、4 KB ページ単位で指定します。

ログ・ファイルのデフォルトのページ数は 4096 です。これは、16 MB のログ・ファイル・サイズと同じです。

UNIX and Linux システムの場合、ログ・ファイルの最小ページ数は 64 です。Windows の場合、ログ・ファイルの最小ページ数は 32 です。最大ページ数はどちらも 65 535 です。

注: キュー・マネージャーの作成時に指定したログ・ファイルのサイズを、個々のキュー・マネージャーについて変更することはできません。

### **LogType=CIRCULAR|LINEAR**

キュー・マネージャーで使用するロギングのタイプ。キュー・マネージャーの作成後に、使用するロギングのタイプを変更することはできません。必要とするタイプのロギングを持つキュー・マネージャーを作成する方法については、108 ページの『[IBM MQ のログのデフォルト](#)』の LogType 属性の説明を参照してください。

#### **CIRCULAR**

システムの停止時に進行中だったトランザクションをロールバックするログを使用して、再始動リカバリーを開始します。

循環ロギングの詳細については、530 ページの『[ログのタイプ](#)』を参照してください。

#### **LINEAR**

再始動リカバリーと、メディアまたは順方向リカバリー (ログの内容を再生することにより、消失または損傷したデータを作成する) の両方を可能にします。

リニア・ロギングの詳細については、530 ページの『[ログのタイプ](#)』を参照してください。

## LogBufferPages=0|0-4096

書き込み用としてバッファ・レコードに割り振るメモリーの量。バッファのサイズは 4 KB ページ単位で指定します。

バッファの最小ページ数は 18 であり、最大ページ数は 4096 です。バッファが大きいとスループットが高くなりますが、これは特に大きなメッセージに適用されます。

0 (デフォルト) を指定すると、サイズはキュー・マネージャーによって選択されます。IBM WebSphere MQ 7.1 では、これは 512 (2048 KB) になります。

1 から 17 の範囲の数を指定すると、キュー・マネージャーはデフォルトで 18 (72 KB) になります。18 から 4096 の範囲の数を指定すると、キュー・マネージャーは指定された数を使用して割り振るメモリーを設定します。

この値は、キュー・マネージャーの始動時に検査されます。値は、制限内で増減できます。ただし、この値の変更は、キュー・マネージャーが次に始動されるまで有効になりません。

## LogPath=directory\_name

キュー・マネージャーのログ・ファイルが置かれるディレクトリー。ディレクトリーは、キュー・マネージャーが書き込みを行える ローカル装置に置くか、できればメッセージ・キューとは異なるドライブに置く必要があります。異なるドライブを指定すると、システム障害時の保護機能が加わります。

デフォルトは次のとおりです。

- IBM MQ for Windows の C: \ProgramData\IBM\MQ\log。
- /var/mqm/log (IBM MQ for UNIX および Linux システム)。

ディレクトリーの名前は、-ld フラグを使用して crtmmq コマンドに指定できます。キュー・マネージャーが作成されると、キュー・マネージャー・ディレクトリーの下にもう 1 つディレクトリーが作成され、ログ・ファイルの保存に使用されます。このディレクトリーの名前はキュー・マネージャー名に基づいて付けられます。これによってログ・ファイル・パスが確実に固有になり、ディレクトリー名の長さに関する制限にも抵触しないこととなります。

crtmmq コマンドで -ld を指定しなかった場合は、LogDefaultPath 属性の値が使用されます。

IBM MQ for UNIX および Linux システムでは、ユーザー ID mmq およびグループ mmq はログ・ファイルについての全権限を持っていない限りなりません。これらのファイルの位置を変更する場合は、それらの権限を取得する必要があります。ログ・ファイルがプロダクトで提供されたデフォルト位置にある場合、これは必要ありません。

## LogWriteIntegrity=SingleWrite|DoubleWrite|TripleWrite

高信頼ログ・レコードを書き込むためにロガーが使用する方式。

### TripleWrite

これはデフォルトの方式です。

なお、**DoubleWrite** を選択可能ですが、そのようにした場合、システムでは **TripleWrite** と解釈されます。

### SingleWrite

**SingleWrite** は、IBM MQ リカバリー・ログをホスティングしているファイル・システムおよびデバイスが、最小単位 4KB の書き込みを明示的に保証している場合にのみ使用してください。

すなわち、何らかの原因で 4KB ページの書き込みが失敗した場合、変更前イメージまたは変更後イメージの 2 つの状態しかありません。中間の状態は考えられません。

**注:** 持続するワークロードにおける並行性が十分であるなら、デフォルト値 **TripleWrite** 以外の値に設定することで得られる利点はほとんどありません。

## V 9.0.2

## LogManagement= Manual | Automatic | Archive

ログ・エクステンントを手動またはキュー・マネージャーによって管理するために使用される方式。

この属性は、**LogType** が LINEAR である場合にのみ適用されます。

**LogManagement** の値を変更しても、キュー・マネージャーを再始動するまでその変更は有効になりません。

属性に認識されない値が見つかったら、その値が修正されるまでキュー・マネージャーは始動しません。

### 手動

手動でログ・エクステントを管理します。このオプションを指定すると、リカバリー用のログ・エクステントが不要になった場合でも、キュー・マネージャーは、ログ・エクステントを再使用したり削除したりしません。

### 自動

キュー・マネージャーによってログ・エクステントが自動的に管理されます。このオプションを指定すると、リカバリー用のログ・エクステントが不要になった時点で、キュー・マネージャーがログ・エクステントを再使用したり削除したりできるようになります。アーカイブ保存はできません。

### アーカイブ

キュー・マネージャーによってログ・エクステントが管理されますが、それぞれのログ・エクステントのアーカイブ保存が完了したことをキュー・マネージャーに通知する必要があります。

このオプションを指定すると、不要になったリカバリー用のログ・エクステントのアーカイブ保存が完了したことをキュー・マネージャーに通知した時点で、キュー・マネージャーがログ・エクステントを自由に再使用したり削除したりできるようになります。

この通知は、`RESET QMGR MQSC` コマンドまたは `Reset Queue Manager PCF` コマンドを使用して実行できます。

## Linux → UNIX 制限モード

このオプションは、UNIX and Linux システムのみに適用されます。RestrictedMode スタンザは、`crtmqm` コマンドの `-g` オプションにより設定します。キュー・マネージャーの作成後には、このスタンザを変更しないでください。 `-g` オプションを使用しなかった場合は、`qm.ini` ファイル内にこのスタンザは作成されません。

キュー・マネージャー・データ・ディレクトリー内には、IBM MQ アプリケーションが、キュー・マネージャーに接続している間にファイルを作成するディレクトリーがあります。これらのディレクトリー内にファイルを作成できるように、アプリケーションには、以下の world write アクセス権限が付与されます。

- `/var/mqm/sockets/QMgrName/@ipcc/ssem/hostname/`
- `/var/mqm/sockets/QMgrName/@app/ssem/hostname/`
- `/var/mqm/sockets/QMgrName/zsocketapp/hostname/`

ここで、`QMGRNAME` はキュー・マネージャーの名前で、`hostname` はホスト名です。

一部のシステムでは、すべてのユーザーにこれらのディレクトリーへの書き込みアクセス権限を付与することは望ましくありません。例えば、ユーザーによっては、キュー・マネージャーへのアクセス権限は必要ありません。制限モードでは、キュー・マネージャー・データを保管するディレクトリーに関する許可が変更されます。その後、これらのディレクトリーには指定されたアプリケーション・グループのメンバーだけがアクセスできるようになります。キュー・マネージャーとの通信に使用される System V IPC 共有メモリーに対する許可も、同様の仕方に変更されます。

アプリケーション・グループは、以下の内容を許可されるメンバーで構成されるグループの名前です。

- MQI アプリケーションの実行
- すべての IPCC リソースの更新
- 一部のキュー・マネージャー・ディレクトリーの内容の変更

キュー・マネージャーの制限モードを使用するには、次の条件を満たす必要があります。

- キュー・マネージャーの作成者は、`mqm` グループとアプリケーション・グループに含まれている必要があります。
- `mqm` ユーザー ID は、アプリケーション・グループに含まれている必要があります。
- キュー・マネージャーを管理するすべてのユーザーは、`mqm` グループとアプリケーション・グループに含まれている必要があります。

- IBM MQ アプリケーションを実行するすべてのユーザーは、アプリケーション・グループに含まれている必要があります。

アプリケーション・グループに入っていないユーザーが実行した MQCONN または MQCONNX の呼び出しは、理由コード MQRC\_Q\_MGR\_NOT\_AVAILABLE で失敗します。

**重要:** 多くのオペレーティング・システムでは、グループへのユーザーの追加をシステムに認識させるために、該当ユーザーがいったんログオフしてから再度ログオンする必要があります。

制限モードは、IBM MQ 許可サービスで作動します。そのため、ユーザーには、IBM MQ に接続し、IBM MQ 許可サービスを使用して必要なリソースにアクセスする権限も付与する必要があります。

**ULW** IBM MQ 許可サービスの構成について詳しくは、[Windows、UNIX and Linux システムでのセキュリティのセットアップ](#)をご覧ください。

IBM MQ の制限モードは、許可サービスによって提供される制御によってキュー・マネージャー・リソースが十分に分離されない場合にのみ使用してください。

## XA リソース・マネージャー

IBM MQ Explorer の XA resource manager キュー・マネージャー・プロパティ・ページ、または qm.ini ファイルの XAResourceManager スタンザを使用して、キュー・マネージャーによって調整されるグローバル作業単位に関係するリソース・マネージャーに関する以下の情報を指定します。

グローバル作業単位に関係するリソース・マネージャーのインスタンスごとに XA リソース・マネージャー構成情報を手動で追加します。デフォルト値はありません。

リソース・マネージャーの属性について詳しくは、[データベースの調整](#)を参照してください。

### Name=name (mandatory)

この属性は、リソース・マネージャーのインスタンスを識別します。

Name 値は長さが 31 文字までです。XA スイッチ構造で定義されているリソース・マネージャーの名前を使用することができます。ただし、同じリソース・マネージャーの複数のインスタンスを使用している場合は、各インスタンスごとに固有の名前を作成する必要があります。例えば、次のように、データベースの名前を Name ストリングに組み込むことによって、それぞれの名前を固有に保つことができます。

IBM MQ は、メッセージおよび dspmqtrn コマンドからの出力で Name 値を使用します。

関連するキュー・マネージャーが開始され、リソース・マネージャー名が有効になったら、リソース・マネージャーのインスタンスの名前変更、あるいは構成情報からの項目の削除は行わないでください。

### SwitchFile=name (mandatory)

リソース・マネージャーの XA スイッチ構造を収容するロード・ファイルの完全修飾名。

64 ビットのキュー・マネージャーを 32 ビット・アプリケーションと一緒に使用する場合、name 値に含まれるのは、リソース・マネージャーの XA スイッチ構造を含むロード・ファイルのベース名のみでなければなりません。

32 ビット・ファイルは、ExitsDefaultPath によって指定されたパスからアプリケーションヘロードされます。

64 ビット・ファイルは、ExitsDefaultPath64 によって指定されたパスからキュー・マネージャーヘロードされます。

### XAOpenString=string (オプション)

リソース・マネージャーの xa\_open エントリー・ポイントに渡されるデータのストリング。このストリングの内容は、リソース・マネージャーによって異なります。例えば、このリソース・マネージャーのインスタンスがアクセスするデータベースを識別します。この属性の定義方法については、以下のセクションを参照してください。

- [Db2 のリソース・マネージャー構成情報の追加](#)
- [Oracle のリソース・マネージャー構成情報の追加](#)
- [Sybase 用のリソース・マネージャー構成情報の追加](#)

- [Informix®のリソース・マネージャー構成情報の追加](#)

また、適切なストリングについては、リソース・マネージャーの資料を参照してください。

### **XACloseString=string (オプション)**

リソース・マネージャーの `xa_close` エントリ・ポイントに渡されるデータのストリング。このストリングの内容は、リソース・マネージャーによって異なります。この属性の定義方法については、以下のセクションを参照してください。

- [Db2 のリソース・マネージャー構成情報の追加](#)
- [Oracle のリソース・マネージャー構成情報の追加](#)
- [Sybase 用のリソース・マネージャー構成情報の追加](#)
- [Informix のリソース・マネージャー構成情報の追加](#)

また、適切なストリングについては、データベースの資料を参照してください。

### **ThreadOfControl=THREAD|PROCESS**

**Windows** この属性は、IBM MQ for Windows に必須です。キュー・マネージャーは、自身のマルチスレッド・プロセスの 1 つから リソース・マネージャーを呼び出す必要がある場合に、この値を使用してシリアライゼーションの処理をします。

#### **スレッド**

リソース・マネージャーは完全にスレッド認識型です。マルチスレッド IBM MQ プロセスでは、外部のリソース・マネージャーに対して複数のスレッドから XA 関数呼び出しを同時に行うことができます。

#### **PROCESS**

リソース・マネージャーはスレッド安全ではありません。マルチスレッド IBM MQ プロセスでは、リソース・マネージャーに対して行える XA 関数呼び出しは一度に 1 つのみです。

ThreadOfControl 項目は、マルチスレッド・アプリケーション・プロセスのキュー・マネージャーによって発行される XA 関数呼び出しには適用されません。通常、別々のスレッドに並行する複数の作業単位を持つアプリケーションは、各リソース・マネージャーでサポートされるためにこのモードの操作が必要となります。

## **channels スタンザの属性**

これらの属性は、チャンネルの構成を決定します。

**z/OS** ただし、この情報は IBM MQ for z/OS には適用されません。

IBM MQ Explorer の Channels キュー・マネージャー・プロパティ・ページ、または `qm.ini` ファイルの CHANNELS スタンザを使用して、チャンネルに関する情報を指定します。

### **MaxChannels=100|number**

使用できる現行チャンネルの最大数。

デフォルトは 100 です。

必要に応じて、**MaxChannels** を別の値に設定し、現行チャンネルの最大数を制限することができます。IBM MQ Appliance では、デフォルト値は 999 999 999 です。この値は変更しないでください。

### **MaxActiveChannels=MaxChannels\_value**

任意のある時点でアクティブにできるチャンネルの最大数。デフォルトは、MaxChannels 属性に指定されている値です。

### **MaxInitiators=3|number**

開始プログラムの最大数。デフォルト値と最大値は 3 です。

### **MQIBindType=FASTPATH|STANDARD**

アプリケーション用のバインディング方法。次のとおりです。

#### **FASTPATH**

チャンネルは MQCONNX FASTPATH を使用して接続されます。つまり、エージェント・プロセスはありません。

## STANDARD

チャンネルは STANDARD を使用して接続されます。

### PipeLineLength=1|number

チャンネルが使用する並行スレッドの最大数。デフォルトは、1 です。1 より大きい値は 2 として扱われます。

パイプラインを使用する場合は、PipeLineLength が 1 より大きくなるようにチャンネルの両端のキュー・マネージャーを構成してください。

注：パイプラインは、TCP/IP チャンネルでのみ有効です。

### AdoptNewMCA=NO|SVR|SDR|RCVR|CLUSRCVR|ALL|FASTPATH

IBM MQ がチャンネル開始要求を受け取ったときに、チャンネルのインスタンスが既に実行中であることが検出された場合、新しいインスタンスを開始する前に、既存のチャンネル・インスタンスを停止する必要があります。AdoptNewMCA 属性を使用すると、どの種類のチャンネルをこの方法に従って終了可能にするかを制御できます。

特定のチャンネル・タイプについて AdoptNewMCA 属性を指定すると、一致するチャンネル・インスタンスがすでに実行中であるために新規チャンネルが始動できない場合は、次のようになります。

1. 新規チャンネルは、終了要求を出して既存のチャンネルを停止しようとします。
2. 既存のチャンネル・サーバーが、AdoptNewMCATimeout に指定されている待ち時間が満了するまでにこの要求に応じない場合は、既存のチャンネル・サーバーのスレッドまたはプロセスが終了されます。
3. ステップ 2 の実行後も既存のチャンネル・サーバーが終了していない場合は、AdoptNewMCATimeout の待ち時間が 2 回目に満了した時点で、IBM MQ はそのチャンネルを終了し、CHANNEL IN USE エラーを返します。

AdoptNewMCA 機能は、サーバー、送信側、受信側、およびクラスター受信側の各チャンネルに適用されます。送信側チャンネルまたはサーバー・チャンネルの場合、受信側のキュー・マネージャーでは、特定の名前を持つチャンネルの 1 つのインスタンスだけが実行可能です。受信側チャンネルまたはクラスター受信側チャンネルの場合、受信側のキュー・マネージャーでは、特定の名前を持つチャンネルの複数のインスタンスが実行可能です。ただし、特定のリモート・キュー・マネージャーからは、ある時点で 1 つのインスタンスだけが実行可能です。

注：リクエスト・チャンネルまたはサーバー接続チャンネルに対しては、AdoptNewMCA はサポートされません。

以下のリストから 1 つ以上の値を選んで、コンマまたはブランクで区切って指定します。

#### NO

AdoptNewMCA 機能が不要であることを意味します。これがデフォルトです。

#### SVR

サーバー・チャンネルが採用されます。

#### SDR

送信側チャンネルが採用されます。

#### RCVR

受信側チャンネルが採用されます。

#### CLUSRCVR

クラスター受信側チャンネルが採用されます。

#### ALL

FASTPATH チャンネル以外のすべてのチャンネル・タイプが採用されます。

#### FASTPATH

該当チャンネルが FASTPATH チャンネルのときは、そのチャンネルが採用されます。これが起きるのは、AdoptNewMCA=RCVR,SVR,FASTPATH などのように該当するチャンネル・タイプも指定されている場合のみです。

**重要!** FASTPATH チャンネルの場合は、AdoptNewMCA 属性が予期しない動作をすることがあります。FASTPATH チャンネルの AdoptNewMCA 属性を使用可能にするときは、十分な注意が必要です。

## AdoptNewMCATimeout=60 |1 - 3600

新しいチャンネル・インスタンスが古いチャンネル・インスタンスの終了を待つ時間 (秒)。1 から 3600 までの範囲の値を指定します。デフォルト値は 60 です。

## AdoptNewMCACheck=QM|ADDRESS|NAME|ALL

AdoptNewMCA 属性を使用可能にするときに必要な検査のタイプ。可能な場合は、故意または不注意によりチャンネルがシャットダウンされるのを防ぐために、すべての検査を実行してください。最低限、チャンネル名が一致することだけは確認してください。

QM、NAME、または ALL の場合はコンマまたはブランクで区切って、以下の 1 つ以上の値を指定します。

### QM

キュー・マネージャー名が一致することを検査します。

QMID ではなく、キュー・マネージャー名自体が突き合わされることに注意してください。

### ADDRESS

通信ソースの IP アドレスを検査します。例えば、TCP/IP アドレス。

**注:** コンマ区切りの CONNAME 値は、ターゲット・アドレスに適用されるため、このオプションには関係ありません。

複数インスタンス・キュー・マネージャーが hosta から hostb にフェイルオーバーした場合、そのキュー・マネージャーのすべてのアウトバウンド・チャンネルは hostb のソース IP アドレスを使用します。これが hosta と異なる場合、AdoptNewMCACheck=ADDRESS の突き合わせは失敗します。

相互認証に SSL または TLS を使用することで、実行中の既存のチャンネルがアタッカーによって中断されるのを防止できます。別の方法として、複数インスタンス・キュー・マネージャーの代わりに、IP テイクオーバーを行う HACMP タイプのソリューションを使用する方法と、ネットワーク・ロード・バランサーを使用してソース IP アドレスをマスクする方法があります。

### 名前

チャンネル名が一致することを検査します。

### ALL

キュー・マネージャー名の一致、通信アドレス、およびチャンネル名の一致を検査します。

デフォルトは AdoptNewMCACheck=NAME, ADDRESS, QM です。

## V 9.0.4 ChlauthEarlyAdopt = Y | N

IBM MQ クライアント・アプリケーション接続のセキュリティー・コンテキストを決定するときに、接続認証規則とチャンネル認証規則の処理順序が重要な要素になります。



**重要:** **ChlauthEarlyAdopt** が qm.ini ファイルに存在しない場合のデフォルトは N です。ただし、IBM MQ 9.0.4 以降、すべてのキュー・マネージャーは **ChlauthEarlyAdopt=Y** で作成され、自動的に qm.ini ファイルに追加されます。

キュー・マネージャー上の接続認証 AUTHINFO オブジェクトに ADOPTCTX(YES) が設定されている場合、**ChlauthEarlyAdopt** は、接続認証のためにキュー・マネージャーに指定されたユーザー ID のみを採用します。

**ChlauthEarlyAdopt** の有効値は、以下の値です。

### Y

チャンネルは、チャンネル認証規則を適用する前に、キュー・マネージャー接続認証を使用して、アプリケーションから提供されたユーザー ID とパスワードの資格情報を検証し、採用します。この操作モードでは、チャンネル認証規則は、接続認証チェックの結果生成されるユーザー ID と照合されます。

### N

チャンネルは、チャンネル認証規則が適用されるまで、アプリケーションから提供されたユーザー ID とパスワードの資格情報の接続認証検証を遅らせます。この操作モードでは、チャンネル認証のプロッキング規則とマッピング規則は、ユーザー ID とパスワードの検証結果を考慮できません。

例えば、デフォルトの認証情報オブジェクトが **ADOPTCTX(YES)** に設定され、ユーザー fred がログインしているとします。次の2つの CHLAUTH 規則が構成されています。

```
SET CHLAUTH('MY.CHLAUTH') TYPE(ADDRESSMAP) DESCR('Block all access by
default') ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
SET CHLAUTH('MY.CHLAUTH') TYPE(USERMAP) DESCR('Allow user bob and force
CONNAUTH') CLNTUSER('bob') CHCKCLNT(REQUIRED) USERSRC(CHANNEL)
```

次のコマンドを、ユーザー bob のセキュリティー・コンテキストを採用してコマンドを認証する目的で発行します。

```
runmqsc -c -u bob QMGR
```

実際には、キュー・マネージャーは bob ではなく fred のセキュリティー・コンテキストを使用するため、接続に失敗します。

bob のセキュリティー・コンテキストを使用するには、**ChlauthEarlyAdopt** を Y に設定する必要があります。

### PasswordProtection = Compatible|always|optional

IBM MQ 8.0 から、TLS を使用する代わりに、保護されたパスワードを MQCSP 構造に設定します。

MQCSP パスワード保護は TLS 暗号化より設定が簡単なので、テストや開発目的に役立ちます。ただし、それほど安全ではありません。

詳しくは、[MQCSP パスワード保護](#)を参照してください。

### ChlauthIssueWarn = y

SET CHLAUTH コマンドで **WARN = YES** 属性を設定したときに、AMQ9787 を生成する場合、この属性を設定します。

### 関連概念

#### 171 ページの『チャネルの状態』

チャネルは、いつでも多くの状態のうちの一つであることができます。いくつかの状態には、副状態もあります。チャネルは、指定された状態から他の状態に移動することができます。

## TCP、LU62、および NETBIOS

ネットワーク・プロトコル構成パラメーターを指定するには、以下のキュー・マネージャー・プロパティ・ページ、または qm.ini ファイル内のスタンザを使用します。これらのスタンザは、チャネルのデフォルトの属性を指定変更します。

### TCP

IBM MQ Explorer の TCP キュー・マネージャー・プロパティ・ページ、または qm.ini ファイルの TCP スタンザを使用して、伝送制御プロトコル/Internet Protocol (TCP/IP) 構成パラメーターを指定します。

#### Port=1414|port\_number

TCP/IP セッション用の 10 進表記のデフォルトのポート番号。IBM MQ の事前割り当てポート番号は 1414 です。

#### Library1= DLLName1 ( IBM MQ for Windows のみ)

TCP/IP ソケット DLL の名前。

デフォルトは WSOCK32 です。

#### KeepAlive=NO|YES

KeepAlive 機能のオン/オフを切り替えます。KeepAlive=YES を指定すると、TCP/IP は、接続の相手側がまだ使用可能かどうかを定期的に検査します。有効でない場合、チャネルは閉じた状態です。

#### ListenerBacklog=number

TCP/IP リスナーの場合のデフォルトの未解決要求数を指定変更します。

TCP/IP で受信するときに、未解決の接続要求の最大数を設定します。これは、TCP/IP ポート上で listener からの受け入れを待っている要求のバックログと考えることができます。デフォルトのリスナー・バックログ値は、[125 ページの表 12](#) に示すとおりです。

表 12. デフォルトの未解決接続要求数 (TCP)	
プラットフォーム	ListenerBacklog のデフォルト値
Windows Server	100
Windows ワークステーション	5
Linux	100
Solaris	100
HP-UX	20
AIX 5.3 以降	100

注: オペレーティング・システムの中には、表に示されているデフォルトより大きい値をサポートするものもあります。接続限度に達しないようにするために、この表の値を使用してください。

これとは反対に、オペレーティング・システムの中には、TCP バックログのサイズを制限する可能性のあるものがあり、有効な TCP バックログは、ここで要求されるよりも小さい可能性があります。

バックログが [125 ページの表 12](#) に示されている値に達すると、TCP/IP 接続は拒否され、チャンネルは開始できなくなります。メッセージ・チャンネルの場合は、結果としてチャンネルは RETRY 状態になり、後で接続を再試行することになります。クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC\_Q\_MGR\_NOT\_AVAILABLE を受け取り、後で接続を再試行します。

以下のプロパティのグループは、TCP/IP によって使用されるバッファのサイズを制御するために使用できます。値は、オペレーティング・システムの TCP/IP 層に直接渡されます。これらのプロパティを使用する際は、十分注意する必要があります。値が正しく設定されないと、TCP/IP のパフォーマンスに悪影響が及ぶ可能性があります。値が正しく設定されない場合にパフォーマンスに及ぶ影響について詳しくは、ご使用の環境の TCP/IP の資料を参照してください。値 0 は、IBM MQ によってバッファ・サイズが修正されるのではなく、オペレーティング・システムがバッファ・サイズを管理することを意味します。

#### Connect\_Timeout=number

ソケットへの接続試行がタイムアウトになるまでの秒数。デフォルト値である 0 は、接続のタイムアウトがないことを指定します。

IBM MQ チャンネル・プロセスは非ブロッキング・ソケットを介して接続します。このため、ソケットのもう一方の側の準備ができていない場合、connect() は即時に EINPROGRESS または EWOULDBLOCK を返します。これに続いて接続が最大 20 回まで再試行された後、通信エラーがレポートされます。

Connect\_Timeout がゼロ以外の値に設定されている場合、IBM MQ は、ソケットの準備ができるまで、select() 呼び出しに対して規定された期間待機します。これにより、後続の connect() 呼び出しが成功する確率は上昇します。このオプションは、ネットワークの負荷が高くなっていて、接続に多少の待機時間が必要なときに役に立つ場合があります。

#### SndBuffSize=number|0

チャンネルの送信側で使用される TCP/IP 送信バッファのサイズ (バイト)。このスタンザ値は、このチャンネル・タイプをさらに具体的に指定するスタンザ (例えば RcvSndBuffSize) でオーバーライドされる場合があります。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**Multi** IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます ([89 ページの『キュー・マネージャー構成ファイル qm.ini』](#) を参照してください)。

#### RcvBuffSize=number|0

チャンネルの受信側で使用される TCP/IP 受信バッファのサイズ (バイト)。このスタンザ値は、このチャンネル・タイプをさらに具体的に指定するスタンザ (例えば RcvRcvBuffSize) でオーバーライド

される場合があります。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**Multi** IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (89 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください)。

#### **RcvSndBuffSize=number|0**

受信側チャンネルの送信側で使用される TCP/IP 送信バッファのサイズ (バイト)。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**Multi** IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (89 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください)。

#### **RcvRcvBuffSize=number|0**

受信側チャンネルの受信側で使用される TCP/IP 受信バッファのサイズ (バイト)。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**Multi** IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (89 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください)。

#### **SvrSndBuffSize=number|0**

クライアント接続チャンネルとサーバー接続チャンネルのサーバー側で使用される TCP/IP 送信バッファのサイズ (バイト)。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**Multi** IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (89 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください)。

#### **SvrRcvBuffSize=number|0**

クライアント接続チャンネルとサーバー接続チャンネルのサーバー側で使用される TCP/IP 受信バッファのサイズ (バイト)。値がゼロとして設定されている場合、オペレーティング・システムのデフォルトが使用されます。値が設定されていない場合、IBM MQ は、デフォルトの 32768 が使用されます。

**Multi** IBM MQ 8.0 以降、新規キュー・マネージャーは自動的に 0 にデフォルト設定された状態で作成されます (89 ページの『キュー・マネージャー構成ファイル qm.ini』を参照してください)。

### **Windows LU62 (IBM MQ for Windows のみ)**

IBM MQ Explorer の LU6.2 キュー・マネージャー・プロパティ・ページ、または qm.ini ファイルの LU62 スタンザを使用して、SNA LU 6.2 プロトコル構成パラメーターを指定します。

#### **TPName**

リモート・サイトで始動する TP 名。

#### **Library1=DLLName 1**

APPC DLL の名前。

デフォルト値は WCPIC32 です。

#### **Library2= DLLName2**

Library1 と同様に、コードが 2 つの別個のライブラリーに格納されている場合に使用されます。

デフォルト値は WCPIC32 です。

## Windows NETBIOS (IBM MQ for Windows のみ)

IBM MQ Explorer の Netbios キュー・マネージャー・プロパティ・ページ、または qm.ini ファイルの NETBIOS スタンザを使用して、NetBIOS プロトコル構成パラメーターを指定します。

### LocalName=*name*

このマシンが LAN 上で識別される名前。

### AdapterNum=*0*|*adapter\_number*

LAN アダプターの番号。デフォルトはアダプター 0 です。

### NumSess=*1*|*number\_of\_sessions*

割り振るセッションの数。デフォルトは、1 です。

### NumCmds=*1*|*number\_of\_commands*

割り振るコマンドの数。デフォルトは、1 です。

### NumNames=*1*|*number\_of\_names*

割り振る名前の数。デフォルトは、1 です。

### Library1= *DLLName1*

NetBIOS DLL の名前。

デフォルト値は NETAPI32 です。

## Windows SPX (IBM MQ for Windows のみ)

SPX プロトコル構成パラメーターを指定するには、IBM MQ Explorer の SPX キュー・マネージャー・プロパティ・ページ、または qm.ini ファイルの SPX スタンザを使用します。

### Socket=*5E86*|*socket\_number*

16 進表記の SPX ソケット番号。デフォルトは X'5E86' です。

### BoardNum=*0*|*adapter\_number*

LAN アダプターの番号。デフォルトはアダプター 0 です。

### KeepAlive=*NO*|*YES*

KeepAlive 機能のオン/オフを切り替えます。

KeepAlive=YES を指定すると、SPX は、接続の相手側がまだ使用可能かどうかを定期的に検査します。有効でない場合、チャンネルは閉じた状態です。

### Library1= *DLLName1*

SPX DLL の名前。

デフォルトは WSOCK32.DLL です。

### Library2= *DLLName2*

LibraryName1 と同じもので、コードが 2 つの異なるライブラリーに保管されている場合に使用されます。

デフォルトは WSOCK32.DLL です。

### ListenerBacklog=*number*

SPX リスナーの場合のデフォルトの未解決要求数を指定変更します。

SPX で受信するときに、未解決の接続要求の最大数を設定します。これは、SPX ソケット上でリスナーからの受け入れを待っている要求のバックログと考えることができます。デフォルトのリスナー・バックログ値は、127 ページの表 13 に示すとおりです。

表 13. デフォルトの未解決接続要求数 (SPX)	
プラットフォーム	ListenerBacklog のデフォルト値
Windows Server	100
Windows ワークステーション	5

注：オペレーティング・システムの中には、表に示されているデフォルトより大きい値をサポートするものもあります。接続限度に達しないようにするために、この表の値を使用してください。

これとは反対に、オペレーティング・システムの中には、SPX バックログのサイズを制限する可能性のあるものがあり、有効な SPX バックログは、ここで要求されるよりも小さい可能性があります。

バックログが [127 ページの表 13](#) に示されている値に達すると、SPX 接続は拒否され、チャンネルは開始できなくなります。メッセージ・チャンネルの場合は、結果としてチャンネルは RETRY 状態になり、後で接続を再試行することになります。クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC\_Q\_MGR\_NOT\_AVAILABLE を受け取り、後で接続を再試行することが必要になります。

## 出口パス

IBM MQ Explorer の Exits キュー・マネージャー・プロパティ・ページ、または qm.ini ファイルの ExitPath スタンザを使用して、キュー・マネージャー・システム上のユーザー出口プログラムのパスを指定します。

### ExitsDefaultPath=string

ExitsDefaultPath 属性は、次の出口の位置を指定します。

- クライアントの 32 ビット・チャンネル出口
- サーバーの 32 ビット・チャンネル出口およびデータ変換出口
- 非修飾 XA スイッチ・ロード・ファイル

### ExitsDefaultPath64= string

ExitsDefaultPath64 属性は、次の出口の位置を指定します。

- クライアントの 64 ビット・チャンネル出口
- サーバーの 64 ビット・チャンネル出口およびデータ変換出口
- 非修飾 XA スイッチ・ロード・ファイル

## API 出口

サーバーの場合、IBM MQ Explorer の Exits キュー・マネージャー・プロパティ・ページ、または qm.ini ファイルの ApiExitLocal スタンザを使用して、キュー・マネージャーの API 出口ルーチンを識別します。クライアントの場合、キュー・マネージャーの API 出口ルーチンを識別するには、mqclient.ini ファイルの ApiExitLocal スタンザを変更します。

Windows システムでは、API 出口の項目を変更するために、amqmdain コマンドを使用することもできます。(すべてのキュー・マネージャーの API 出口ルーチンを識別するには、[111 ページの『API 出口』](#)に説明されているとおりに、ApiExitCommon および ApiExitTemplate スタンザを使用します。)

API 出口が正しく機能するためには、サーバーからのメッセージを変換されていない状態でクライアントに送信する必要がありますので、注意してください。そのメッセージは、API 出口がメッセージを処理した後に、クライアントで変換される必要があります。そのため、すべての変換出口がクライアントにインストールされている必要があります。

これらのスタンザの属性の詳細な説明については、[API 出口の構成](#)を参照してください。

## V 9.0.5 診断メッセージ・ロギング

IBM MQ の診断メッセージ・ログは、IBM MQ システムのさまざまなコンポーネントを使用して IBM MQ 構成やランタイムの状態変更および問題に関連する診断メッセージを報告するメカニズムです。

これらのログは、IBM MQ エラー・ログと呼ばれることもありますが、エラー・メッセージだけでなく、IBM MQ 情報メッセージや警告メッセージも常に含まれています。これらのログに報告を行う 3 つの主な IBM MQ コンポーネントは以下のとおりです。

- キュー・マネージャー
- IBM MQ クライアント
- IBM MQ システムの残り

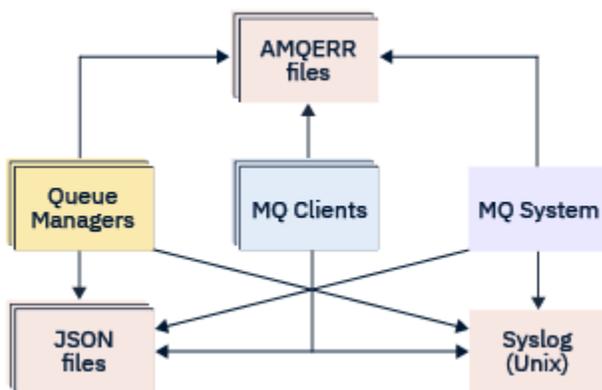
IBM MQ では、診断メッセージ・サービス と呼ばれるさまざまな方式を介した診断メッセージの報告がサポートされているため、以下の情報を記録および消費するためのアプローチを調整できます。

- AMQERRnn ログ・ファイル
- JSON 形式のログ・ファイル
- **UNIX** JSON 形式の Syslog

IBM MQ による JSON 出力は、1 行の JSON オブジェクトとしてフォーマットされます。つまり、JSON ログまたは Syslog レコードの各行は、有効な JSON オブジェクトを表します。ログの全体は、単一の JSON オブジェクトとしてカプセル化されません。

以下の図は、キュー・マネージャー、IBM MQ クライアント、および IBM MQ システムのすべてが、説明される方式を使用して診断メッセージを報告できることを示しています。

図 6. IBM MQ の各部分が診断メッセージを報告する方法



## IBM MQ 診断ログの構成方法

診断ログは、qm.ini ファイル内のスタanzasを使用して、それを必要とする IBM MQ コンポーネントに応じて、スタanzasを使用して定義およびカスタマイズされます。それぞれ固有のロギング・エンドポイントが、ini ファイル内の独自のスタanzas・ヘッディングの下で、カスタマイズと共に定義されます。カスタマイズには以下のことが含まれます。

- ロールオーバーが発生する前に循環させるログ・ファイルのサイズ (Syslog には適用されない)
- ログ・メッセージの重大度に基づくフィルター処理
- 特定のメッセージ・コードの抑止

IBM MQ は 3 つのタイプのロギング・エンドポイントのいずれかまたはすべてに書き込むように構成できるため、特定のログ・スタanzasが特定の役割を果たすようにできます。同様に、複数のファイル・サービスも定義できます。以下に例を示します。

- JSON 形式を使用すると、ローカル環境および Cloud 環境内で自動化ツールを介して容易に解析できるようになります。
- Syslog 出力を使用すると、IBM MQ コンポーネントで、システムの他の製品に合わせた共通 OS ロギングの場所に診断情報を統合できます。
- 重大度に基づいてフィルター処理されるログ・エンドポイントでは、例えば、システム内の重大エラーのみを特定のログ・ファイルに記録できます。

構成された診断ロギングのスタイルに関係なく、IBM MQ システム・ログ・ディレクトリー (/var/mqm/errors/AMQERRnn.log) および特定のキュー・マネージャー・ログ・ディレクトリー (/var/mqm/qmgrs/<qmgr\_name>/errors/AMQERRnn.log) の下に保持される従来型の診断ファイルには、使用されるその他のロギング構成に加えて、常に書き込まれます。

キュー・マネージャーの場合のみ、131 ページの『診断メッセージ・サービス・スタanzas』の属性を指定して、これらの必須ログのオプション構成を実行できます。

## さまざまなスタンプ領域

IBM MQ のさまざまな領域に追加スタンプを適用できます。

### Qmgr(qm.ini)

キュー・マネージャーによって生成されるログ・メッセージに適用されます。

### System(mqs.ini)

システムによって生成されるログ・メッセージに適用されます。このオプションは、キュー・マネージャーがそれ自体のログにアクセスまたは書き込みできない場合を除き、キュー・マネージャーに固有ではありません。

### Templates(mqs.ini)

テンプレートとして使用される 1 つ以上のスタンプであり、キュー・マネージャーが作成されるときに qm.ini にコピーされます。

### Client(mqclient.ini)

クライアント操作に適用されます (例えば、リモート・キュー・マネージャーへのクライアント・モードでの `runmqsc`)。

## JSON 形式のログと従来形式のログの間の変換

`mqrc` コマンドが拡張され、JSON 形式のログと従来形式のログの間、および異なる言語間での多くの変換が可能になりました。

### 関連資料

#### 131 ページの『診断メッセージ・サービス・スタンプ』

使用可能な診断メッセージ・サービス・オプションによって IBM MQ 診断ロギングをカスタマイズできるので、さまざまな IBM MQ コンポーネントのさまざまなログ・エンドポイントにログ出力を送信できます。

#### 130 ページの『QMErrorLog スタンプ (UNIX, Linux, and Windows)』

QMErrorLog サービスは、キュー・マネージャーに関する診断メッセージの出力に使用される従来型の IBM MQ 診断ロギング・サービスです。QMErrorLog サービスは、継続的に実行され、オフにすることはできませんが、ある程度カスタマイズすることができます。

#### 134 ページの『診断メッセージ・サービス』

以下の診断メッセージ・サービスおよびそれらのサービス固有の属性 (構成ファイルの `DiagnosticSystemMessages`、`DiagnosticMessages` および `DiagnosticMessagesTemplate` の各スタンプの下で指定される) を定義できます。

ULW

### QMErrorLog スタンプ (UNIX, Linux, and Windows)

QMErrorLog サービスは、キュー・マネージャーに関する診断メッセージの出力に使用される従来型の IBM MQ 診断ロギング・サービスです。QMErrorLog サービスは、継続的に実行され、オフにすることはできませんが、ある程度カスタマイズすることができます。

このスタンプは、IBM MQ システムやクライアントの ini ファイル構成には適用されません。

IBM MQ Explorer の拡張キュー・マネージャー・プロパティ・ページ、または `qm.ini` ファイルの QMErrorLog スタンプを使用して、IBM MQ エラー・ログの操作と内容を調整します。



**重要:** IBM MQ Explorer を使用して変更を行うことができるのは、Windows プラットフォームでローカル・キュー・マネージャーを使用している場合のみです。

V 9.0.4

### ErrorLogSize=maxsize

キュー・マネージャーのエラー・ログがバックアップにコピーされるときにサイズを指定します。`maxsize` は、32768 から 2147483648 バイトの範囲でなければなりません。**ErrorLogSize** が指定されていないと、デフォルト値である 33554432 バイト (32 MB) が使用されます。

この属性を使用して、最大サイズを必要に応じて以前の最大サイズである 2 MB に戻すことができます。

**重要:** IBM MQ 9.0.4 から、**ErrorLogSize** 属性のデフォルト・サイズが大きくなりました。これは、IBM MQ 9.0.3 からの変更です。

`MQMAXERRORLOGSIZE` 環境変数を使用して、ログのサイズを設定することができます。

### **ExcludeMessage= msgIds**

キュー・マネージャーのエラー・ログに書き込まれないメッセージを指定します。

詳しくは、[131 ページの『診断メッセージ・サービス・スタンザ』](#)の **ExcludeMessage** を参照してください。

### **SuppressMessage= msgIds**

指定した時間間隔内で 1 回だけキュー・マネージャーのエラー・ログに書き込まれるメッセージを指定します。同一のメッセージ ID が **SuppressMessage** と **ExcludeMessage** の両方で指定されていると、そのメッセージは除外されます。

このオプションは、MQ `client.ini` で定義される診断メッセージ・サービスには適用されません。

詳しくは、[131 ページの『診断メッセージ・サービス・スタンザ』](#)の **SuppressMessage** を参照してください。

### **SuppressInterval=length**

**SuppressMessage** に指定したメッセージを 1 回だけキュー・マネージャーのエラー・ログに書き込む時間間隔を秒単位で指定します。*length* は、1 から 86400 秒の範囲でなければなりません。

**SuppressInterval** が指定されていないと、デフォルト値の 30 秒が使用されます。

### **関連資料**

#### [131 ページの『診断メッセージ・サービス・スタンザ』](#)

使用可能な診断メッセージ・サービス・オプションによって IBM MQ 診断ロギングをカスタマイズできるので、さまざまな IBM MQ コンポーネントのさまざまなログ・エンドポイントにログ出力を送信できます。

#### [134 ページの『診断メッセージ・サービス』](#)

以下の診断メッセージ・サービスおよびそれらのサービス固有の属性 (構成ファイルの **DiagnosticSystemMessages**、**DiagnosticMessages** および **DiagnosticMessagesTemplate** の各スタンザの下で指定される) を定義できます。

## **V 9.0.5 診断メッセージ・サービス・スタンザ**

使用可能な診断メッセージ・サービス・オプションによって IBM MQ 診断ロギングをカスタマイズできるので、さまざまな IBM MQ コンポーネントのさまざまなログ・エンドポイントにログ出力を送信できます。

以下のいずれかの名前のスタンザを使用して、追加の診断メッセージ・サービスを有効にします。

### **• DiagnosticSystemMessages**

システム・エラー・ログに送信される診断メッセージの生成時に使用されるサービスを定義します。`mqs.ini` または `mqclient.ini` ファイルで有効です。

クライアント・アプリケーションは **DiagnosticSystemMessages** スタンザを `mqclient.ini` および `mqs.ini` ファイルで使用し、**DiagnosticSystemMessages** スタンザはキュー・マネージャー・コンテキストがないサーバー・アプリケーションのメッセージを制御します。

すべてのメッセージを `syslog` サービスに追加で書き込むキュー・マネージャーとアプリケーションを構成できます。

### **• DiagnosticMessages**

キュー・マネージャー・エラー・ログに送信される診断メッセージの生成時に使用されるサービスを定義します。`qm.ini` ファイルでのみ有効です。

### **• DiagnosticMessagesTemplate**

キュー・マネージャーの作成時に `mqs.ini` ファイルから `qm.ini` ファイルの **DiagnosticMessages** にコピーされるスタンザです。

診断メッセージを表示するには、`mqrcc` コマンドを使用します。

## **スタンザの属性**



**重要:** Service およびスタンザ名は必須です。

**name=<stanzaname>**

スタンプの名前。値は ini ファイル内で一意である必要があります。

**Service= type of service**

この属性では、このスタンプで有効にするサービスを定義します。サービスの名前は、大/小文字を区別しません。

例えば、syslog を追加サービスとして有効にするには、以下のように入力します。

```
Service=syslog
```

134 ページの『[診断メッセージ・サービス](#)』と、診断メッセージ・サービス・スタンプで使用可能なそれらのサービス固有の属性を参照してください。

以下のオプション属性をスタンプに追加できます。

- [ExcludeMessage](#)
- [SuppressMessage](#)
- [SuppressInterval](#)
- [133 ページの『Severities』](#)

**ExcludeMessage= msgIds**

キュー・マネージャーのエラー・ログに書き込まれないメッセージを指定します。IBM MQ システムが大量に使用されている場合、多数のチャンネルが停止して開始している場合は、多数の情報メッセージが z/OS コンソールおよびハードコピー・ログに送信されます。IBM MQ-IMS ブリッジとバッファ管理プログラムも、大量の通知メッセージを生成する場合がありますので、必要な場合に、メッセージを除外すると、大量のメッセージは受信されません。msgIds には、次のメッセージ ID のコンマ区切りリストが含まれています。

- 5211 - プロパティ名の最大長を超えました。
- 5973 - 分散パブリッシュ/サブスクライブのサブスクリプションは禁止されています
- 5974 - 分散パブリッシュ/サブスクライブのパブリッシュは禁止されています
- 6254 - システムは共有ライブラリーを動的にロードできませんでした
- 7234 - Number of messages loaded (いくつかのメッセージが読み込まれました)
- 8245 - エンティティーには、オブジェクトを表示するための十分な権限がありません (Entity has insufficient authority to display object)
- 9001 - Channel program ended normally (チャンネル・プログラムが正常に終了しました)
- 9002 - Channel program started (チャンネル・プログラムが開始されました)
- 9202 - Remote host not available (リモート・ホストは使用できません)
- 9208 - ホストからの受信エラー
- 9209 - 接続の閉止
- 9228 - チャンネル・レスポンスの開始不能
- 9489 - SVRCONN インスタンス数の最大限度を超えました
- 9490 - クライアントあたりの SVRCONN インスタンス数の最大限度を超えました
- 9508 - キュー・マネージャーに接続不能
- 9524 - Remote queue manager unavailable (リモート・キュー・マネージャーは使用できません)
- 9528 - User requested closure of channel (ユーザーがチャンネルの閉止を要求しました)
- 9545 - 切断間隔の有効期限切れ
- 9558 - リモート・チャンネルは使用不可
- 9637 - チャンネルに証明書がありません。
- 9776 - ユーザー ID によるチャンネルのブロック
- 9777 - NOACCESS マップによるチャンネルのブロック
- 9782 - アドレスによる接続のブロック
- 9999 - Channel program ended abnormally (チャンネル・プログラムが異常終了しました)

### SuppressMessage= msgIds

指定した時間間隔内で 1 回だけキュー・マネージャーのエラー・ログに書き込まれるメッセージを指定します。IBM MQ システムが大量に使用されている場合、多数のチャンネルが停止して開始している場合は、多数の情報メッセージが z/OS コンソールおよびハードコピー・ログに送信されます。IBM MQ-IMS ブリッジとバッファ管理プログラムも、大量の通知メッセージを生成する場合がありますので、必要な場合に、メッセージを抑止すると、大量の反復メッセージは受信されません。時間間隔は、SuppressInterval で指定します。msgIds には、次のメッセージ ID のコンマ区切りリストが含まれています。

- 5211 - プロパティ名の最大長を超えました。
- 5973 - 分散パブリッシュ/サブスクライブのサブスクリプションは禁止されています
- 5974 - 分散パブリッシュ/サブスクライブのパブリッシュは禁止されています
- 6254 - システムは共有ライブラリーを動的にロードできませんでした
- 7234 - Number of messages loaded (いくつかのメッセージが読み込まれました)
- 8245 - エンティティーには、オブジェクトを表示するための十分な権限がありません (Entity has insufficient authority to display object)
- 9001 - Channel program ended normally (チャンネル・プログラムが正常に終了しました)
- 9002 - Channel program started (チャンネル・プログラムが開始されました)
- 9202 - Remote host not available (リモート・ホストは使用できません)
- 9208 - ホストからの受信エラー
- 9209 - 接続の閉止
- 9228 - チャンネル・レスポnderの開始不能
- 9489 - SVRCONN インスタンス数の最大限度を超えました
- 9490 - クライアントあたりの SVRCONN インスタンス数の最大限度を超えました
- 9508 - キュー・マネージャーに接続不能
- 9524 - Remote queue manager unavailable (リモート・キュー・マネージャーは使用できません)
- 9528 - User requested closure of channel (ユーザーがチャンネルの閉止を要求しました)
- 9545 - 切断間隔の有効期限切れ
- 9558 - リモート・チャンネルは使用不可
- 9637 - チャンネルに証明書がありません。
- 9776 - ユーザー ID によるチャンネルのブロック
- 9777 - NOACCESS マップによるチャンネルのブロック
- 9782 - アドレスによる接続のブロック
- 9999 - Channel program ended abnormally (チャンネル・プログラムが異常終了しました)

同一のメッセージ ID が SuppressMessage と ExcludeMessage の両方で指定されていると、そのメッセージは除外されます。

このオプションは、MQ client.ini で定義される診断メッセージ・サービスには適用されません。

### SuppressInterval=length

**SuppressMessage** に指定したメッセージを 1 回だけキュー・マネージャーのエラー・ログに書き込む時間間隔を秒単位で指定します。length は、1 から 86400 秒の範囲でなければなりません。

**SuppressInterval** が指定されていないと、デフォルト値の 30 秒が使用されます。

### Severities

重大度レベルのコンマ区切りリスト。重大度レベルの名前は大/小文字を区別しません。許容値は以下のとおりです。

- I (または、Information か 0)
- W (または、Warning か 10)
- E (または、Error か 20 と 30)
- S (または、Stop か 40)
- T (または、System か 50)

注:

1. デフォルト値は all です。
2. 選択した重大度レベルのメッセージのみがサービスに提示されます。  
または、プラス文字 (+) を使用して、指定したエラー・レベル以上のメッセージを表示することもできます。例えば、すべてのエラーを表示するには、以下のように入力します。

Severities=E+

## 関連資料

### 130 ページの『QMErrorLog スタンザ (UNIX, Linux, and Windows)』

QMErrorLog サービスは、キュー・マネージャーに関する診断メッセージの出力に使用される従来型の IBM MQ 診断ロギング・サービスです。QMErrorLog サービスは、継続的に実行され、オフにすることはできませんが、ある程度カスタマイズすることができます。

### 134 ページの『診断メッセージ・サービス』

以下の診断メッセージ・サービスおよびそれらのサービス固有の属性 (構成ファイルの DiagnosticSystemMessages、DiagnosticMessages および DiagnosticMessagesTemplate の各スタンザの下で指定される) を定義できます。

### V9.0.5 診断メッセージ・サービス

以下の診断メッセージ・サービスおよびそれらのサービス固有の属性 (構成ファイルの DiagnosticSystemMessages、DiagnosticMessages および DiagnosticMessagesTemplate の各スタンザの下で指定される) を定義できます。

以下の診断メッセージ・サービスが定義されています。

## ファイル

このサービスでは、QMErrorLog サービスと同様の方法でフィルター処理されていないメッセージをファイルに送信します。Format の指定値に基づいて、指定されている既存のテキスト形式か JSON 形式のどちらかが使用されます。デフォルトでは、Format プロパティに応じて AMQERR01.LOG、AMQERR02.LOG、および AMQERR03.LOG または AMQERR01.json、AMQERR02.json、および AMQERR03.json という 3 つのファイルがあり、これらのロールオーバーは構成されたサイズに基づいています。

以下の属性は、File スタンザだけでサポートされています。

### FilePath

ログ・ファイルの書き込み先のパス。デフォルトは AMQERR01.log ファイルと同じ場所 (システムまたはキュー・マネージャー) です。このパスは絶対パスである必要がありますが、置き換え可能な挿入を含めることができます。以下に例を示します。

#### +MQ\_Q\_MGR\_DATA\_PATH+

キュー・マネージャー診断メッセージ・ディレクトリーの親への絶対パス。デフォルトでは、UNIX プラットフォームの場合は /var/mqm/qmgrs/<QM\_name>、Windows の場合は C:\Program Data\IBM\MQ\qmgrs\<QM\_name> です。

#### +MQ\_DATA\_PATH+

システム診断メッセージ・ディレクトリーの親への絶対パス。デフォルトでは、UNIX プラットフォームの場合は /var/mqm、Windows の場合は C:\Program Data\IBM\MQ です。

既存のエラー・ディレクトリーを使用しない場合は、適切な権限を使用してこのパスを作成する必要があります。

### FilePrefix

ログ・ファイルの接頭部。デフォルトは AMQERR です。

### FileSize

ログがロールオーバーするサイズ。デフォルトは、130 ページの『QMErrorLog スタンザ (UNIX, Linux, and Windows)』の ErrorLogSize プロパティと同様に 32 MB であり、意味的には同じです。

注: ErrorLogSize プロパティは、カスタム診断サービスではなく、デフォルトのエラー・ログ・サービスにのみ適用されます。

IBM MQ 9.0.5 以降では、**MQMAXERRORLOGSIZE** 環境変数を使用してログのサイズを設定できます。

### Format

ファイルの形式。値は、*text* (追加の QMErrorLog スタイル・サービスの場合) または *json* (デフォルト) のどちらかです。

ファイルの接尾部は、この属性の設定に基づいて、*.LOG* と *.json* のどちらかになります。

例えば、キュー・マネージャーの *qm.ini* ファイルを編集し、以下のスタンザを追加します。

```
DiagnosticMessages:  
  Service = File  
  Name = JSONLogs  
  Format = json  
  FilePrefix = AMQERR
```

再始動後、キュー・マネージャーの ERRORS ディレクトリーには *AMQERR0x.json* ファイルが配置されます。

複数の File サービスを定義できます。これにより、以下の例に示す構成が可能になります。異なるタグのメッセージは、異なるログのセットに分割されます。

```
DiagnosticMessages:  
  Name=ErrorsToFile  
  Service=File  
  Severities=E+  
  FilePrefix=OnlyErrors  
  
DiagnosticMessages:  
  Name=NonErrorstoFile  
  Service=File  
  Severities=1 W  
  FilePrefix=Information
```

### システム・ログ

Syslog サービスは、Windows または IBM i では使用できません。

定義できる Syslog サービスは 1 つのみであり、Syslog サービスでは、**JSON 形式**の診断メッセージ仕様を使用して、フィルター処理されていないメッセージが *syslog* に送信されます。情報は表に示されている順序で *syslog* に追加され、*msgID* と挿入で開始されます。

メッセージの重大度は、以下のように *syslog* レベルにマップされます。

重大度	レベル
0	LOG_INFO
10	LOG_WARNING
20	LOG_ERR
30	LOG_ERR
40	LOG_ALERT
50	LOG_ALERT

以下の属性は、*syslog* スタンザだけでサポートされています。

### Ident

*syslog* 項目に関連付けられた **ident** 値を定義します。デフォルト値は *ibm-mq* です。

*Syslog* に送信されるエラー・メッセージの例を以下に示します。

```
DiagnosticMessages:  
  Name=ErrorsToSyslog  
  Service=Syslog
```

スタンザの一般的な属性について詳しくは、[131 ページの『診断メッセージ・サービス・スタンザ』](#)を参照してください。

注:

1. File サービスの場合のみ、それぞれ名前が異なる複数のスタンザを持つことができます。シーケンス内の最後の名前を使用した定義のみが有効になります。
2. スタンザの値に加えた変更は、キュー・マネージャーを再始動した場合にのみ有効になります。

## キュー・マネージャーのデフォルトのバインド・タイプ

IBM MQ Explorer の Extended キュー・マネージャー・プロパティ・ページ、または qm.ini ファイルの Connection スタンザを使用して、デフォルトのバインド・タイプを指定します。なお、Connection スタンザが必要な場合は、作成する必要があります。

### DefaultBindType=SHARED|ISOLATED

DefaultBindType を ISOLATED に設定すると、アプリケーションとキュー・マネージャーは別個のプロセスで実行され、両者間でリソースは共有されません。

DefaultBindType を SHARED に設定すると、アプリケーションとキュー・マネージャーは別個のプロセスで実行されますが、いくつかのリソースは両者間で共有されます。

デフォルト値 SHARED です。



**重要: DefaultBindType** は、すべての MQCONN 呼び出し、および [MQCNO STANDARD BINDING](#) を指定して MQCONNX を使用するすべての呼び出しに適用されます。

**DefaultBindType** を変更すると、一部のアプリケーションでパフォーマンスが低下する場合があります。

## キュー・マネージャー構成ファイルの SSL スタンザ

キュー・マネージャー構成ファイルの SSL スタンザを使用して、キュー・マネージャーの TLS チャンネルを構成します。

### Online Certificate Status Protocol (OCSP)

証明書には、AuthorityInfoAccess 拡張を含めることができます。この拡張は、サーバーへの通信に Online Certificate Status Protocol (OCSP) を使用することを指定します。ご使用のキュー・マネージャーの SSL チャンネルまたは TLS チャンネルが AuthorityInfoAccess 拡張を使用できるようにするには、それらの拡張で指定されている OCSP サーバーが使用可能で、正しく構成されており、ネットワークを通じてアクセスできることを確認してください。詳しくは、[失効した証明書の取り扱い](#)を参照してください。

### CrlDistributionPoint (CDP)

証明書には、CrlDistributionPoint 拡張を含めることができます。この拡張には、証明書取り消しリスト (CRL) のダウンロードに使用されるプロトコルと通信先のサーバーの両方を識別する URL が含まれます。

ご使用のキュー・マネージャーの SSL チャンネルまたは TLS チャンネルが CrlDistributionPoint 拡張を使用できるようにするには、それらの拡張で指定されている CDP サーバーが使用可能で、正しく構成されており、ネットワークを通じてアクセスできることを確認してください。

## SSL スタンザ

qm.ini ファイルの SSL スタンザを使用して、キュー・マネージャーの TLS チャンネルが以下の機能の使用をどのように試行するか、またその使用中に問題が発生した場合にどのように対応するかを構成します。

以下の各ケースでは、指定された値がリストされている有効な値のいずれでもない場合、デフォルトの値が使用されます。無効な値が指定されたことを示すエラー・メッセージは書き込まれません。

#### **CDPCheckExtensions=YES|NO**

CDPCheckExtensions は、このキュー・マネージャーの TLS チャンネルまたは TLS チャンネルが、CrlDistributionPoint 証明書拡張で指定されている CDP サーバーの検査を試行するかどうかを指定します。

- YES: TLS チャンネルまたは TLS チャンネルは、CDP サーバーを検査して、デジタル証明書が取り消されているかどうかを判別しようとします。
- NO: TLS チャンネルまたは TLS チャンネルは、CDP サーバーの検査を試行しません。この値はデフォルトです。

#### **OCSPAAuthentication=REQUIRED|WARN|OPTIONAL**

OCSPAAuthentication は、取り消し状況を OCSP サーバーから判別できない場合に実行されるアクションを指定します。

OCSP 検査が使用可能になっている場合、TLS のチャンネル・プログラムは、OCSP サーバーとの通信を試行します。

チャンネル・プログラムがどの OCSP サーバーとも通信できない場合、あるいは証明書の取り消し状況を提供できるサーバーがない場合は、OCSPAAuthentication パラメーターの値が使用されます。

- REQUIRED: 取り消し状況の判別に失敗すると、エラーを出して接続を閉じます。この値はデフォルトです。
- WARN: 取り消し状況の判別に失敗すると、キュー・マネージャーのエラー・ログに警告メッセージを書き込みますが、接続の続行は許可します。
- OPTIONAL: 取り消し状況の判別に失敗しても、通知なしで接続の続行を許可します。警告やエラーは生成されません。

#### **OCSPCheckExtensions= YES|NO**

OCSPCheckExtensions は、このキュー・マネージャーの TLS チャンネルおよび TLS チャンネルが、AuthorityInfoAccess 証明書拡張で指定されている OCSP サーバーの検査を試行するかどうかを指定します。

- YES: TLS チャンネルおよび TLS チャンネルは、OCSP サーバーを検査して、デジタル証明書が取り消されているかどうかを判別しようとします。この値はデフォルトです。
- NO: TLS チャンネルおよび TLS チャンネルは、OCSP サーバーの検査を試行しません。

#### **SSLHTTPProxyName=string**

ストリングは、OCSP チェックのために GSKit で使用される HTTP プロキシ・サーバーのホスト名またはネットワーク・アドレスのいずれかです。このアドレスの後に、オプションのポート番号が括弧で囲まれて続く場合もあります。ポート番号を指定しないと、デフォルトの HTTP ポートである 80 が使用されます。

 AIX の 32 ビット・クライアント、Solaris SPARC プラットフォーム、HP-UX PA-RISC プラットフォームの場合、ネットワーク・アドレスには IPv4 アドレスのみを使用できます。

その他のプラットフォームでは、ネットワーク・アドレスには IPv4 または IPv6 アドレスを使用できません。

例えば、ファイアウォールが OCSP 応答側の URL へのアクセスを回避する場合など、この属性が必要になる場合があります。

## **出口のプロパティ**

キュー・マネージャーの出口プロパティについての情報を指定するには、IBM MQ Explorer からクラスター・キュー・マネージャー・プロパティ・ページを使用するか、または qm.ini ファイルの

ExitPropertiesLocal スタンザを使用します。あるいは、**amqmdain** コマンドを使用して設定することもできます。

デフォルトでは、この設定は、マシン全体の構成の ExitProperties スタンザに含まれている CLWLMode 属性から継承されます (108 ページの『[出口のプロパティ](#)』を参照してください)。この設定を変更するのは、このキュー・マネージャーを別の方法で構成する場合のみにしてください。この値は、クラスター・キュー・マネージャー・プロパティ・ページのクラスター・ワークロード・モード属性を使用して、キュー・マネージャーごとに個別にオーバーライドできます。

### CLWLMode=SAFE|FAST

クラスター・ワークロード (CLWL) 出口では、MQI 呼び出し (例えば、MQOPEN、MQPUT) に応えて、クラスター内のどのクラスター・キューをオープンするかを指定できます。CLWL 出口は、CLWLMode 属性に指定する値に応じて、FAST モードまたは SAFE モードで実行されます。CLWLMode 属性を省略する場合は、クラスター・ワークロード出口は SAFE モードで実行されます。

#### SAFE

CLWL 出口をキュー・マネージャーから独立したプロセスで実行します。これがデフォルトです。

SAFE モードでの実行中にユーザー作成の CLWL に問題が起きた場合は、次のような結果が生じます。

- CLWL サーバー・プロセス (amqzlw0) は失敗します。
- キュー・マネージャーは CLWL サーバー・プロセスを再始動します。
- エラーがエラー・ログに記録されます。進行中の MQI 呼び出しがある場合は、戻りコードが戻されます。

キュー・マネージャーの健全性は維持されます。

注: CLWL 出口を独立したプロセスで実行すると、パフォーマンスが影響を受けることがあります。

#### FAST

クラスター出口は、キュー・マネージャー・プロセスでインラインで実行します。

このオプションを指定すると、SAFE モードで実行した場合のプロセスの切り替えコストを回避できるのでパフォーマンスは向上しますが、代わりにキュー・マネージャーの健全性が保証されないこととなります。CLWL 出口を FAST モードで実行するのは、CLWL 出口にまったく問題がないという確信があり、パフォーマンスが特に重要な要素である場合のみにしてください。

CLWL 出口を FAST モードで実行しているときに問題が発生すると、キュー・マネージャーは失敗し、キュー・マネージャーの健全性が損なわれる恐れがあります。

## サブプール

このスタンザは、IBM MQ によって作成されます。変更はしないでください。

スタンザ Subpool およびスタンザ内の属性 ShortSubpoolName は、キュー・マネージャーを作成するときに、IBM MQ によって自動的に作成されます。IBM MQ は ShortSubpoolName の値を選択します。この値は変更しないでください。

名前は、/var/mqm/sockets ディレクトリー内に作成されるディレクトリーおよびシンボリック・リンクに対応しています。これは、IBM MQ が実行中のプロセス間の内部通信に使用します。

IBM i

Linux

UNIX

## qm.ini ファイルの Filesystem スタンザ

エラー・ログ・ファイルに設定されたデフォルトの権限は、ほとんどの環境で役に立つことが予想されます。したがって、ほとんどの IBM MQ 管理者は、これらを変更する必要はありません。

ただし、IBM MQ 管理者は、エラー・ログ・ファイルの許可を変更したい場合があります。その場合は、ファイル・システム・スタンザ・オプション **ValidateAuth=No** を設定する必要があります。これにより、キュー・マネージャーは許可を変更しないままになります。

デフォルトの動作 (**ValidateAuth=No** を指定しない場合) では、キュー・マネージャーはキュー・マネージャーのエラー・ログのファイル許可を検査し、それらをデフォルト値に戻します。検査は、キュー・マネージャーの終了時や操作開始時など、いつでも発生することがあります。

## 機密保護

qm.ini ファイルの Security スタンザを使用して、オブジェクト権限マネージャー (OAM) のオプションを指定します。

### ClusterQueueAccessControl=RQMName|Xmitq

クラスター・キュー・マネージャー上でホストされるクラスター・キューまたは完全修飾キューのアクセス制御を検査するには、この属性を設定します。

#### RQMName

リモート側でホストされるキューのアクセス制御について検査されるプロファイルは、指定されたキューのプロファイルまたは指定されたキュー・マネージャーのプロファイルです。

#### Xmitq

リモート側でホストされるキューのアクセス制御について検査されるプロファイルは、SYSTEM.CLUSTER.TRANSMIT.QUEUE に解決されます。

Xmitq はデフォルト値です。

### GroupModel=GlobalGroups

この属性は、Windows 上でユーザーのグループ・メンバーシップを判別する際に、OAM がグローバル・グループを検査するかどうかを決定します。

デフォルトでは、グローバル・グループを検査しません。

#### GlobalGroups

OAM はグローバル・グループを検査します。

GlobalGroups を設定すると、許可コマンド **setmqaut**、**dspmqaut**、および **dmpmqaut** はグローバル・グループ名を受け入れます。**setmqaut -g** パラメーターを参照してください。

注: ClusterQueueAccessControl=RQMName を設定したときに、許可サービスのカスタム実装が MQZAS\_VERSION\_6 未満である場合は、キュー・マネージャーは開始しません。この場合には、ClusterQueueAccessControl=Xmitq を設定するか、またはカスタム許可サービスを MQZAS\_VERSION\_6 以上にアップグレードしてください。

## V 9.0.5 チューニング・パラメーター

qm.ini ファイルの TuningParameters スタンザを使用して、キュー・マネージャーを調整するためのオプションを指定します。

### ImplSyncOpenOutput=value

**ImplSyncOpenOutput** は、書き込みのためにキューを開いておくアプリケーションの最小数です。その数に達すると、同期点以外で永続書き込みのための暗黙的同期点が有効になることがあります。

**ImplSyncOpenOutput** のデフォルト値は 2 です。

PUT 操作のためにキューを開いておくアプリケーションが 1 つだけの場合は、**ImplSyncOpenOutput** がオフに切り替えられます。

**ImplSyncOpenOutput=1** を指定すると、暗黙的同期点の可能性が常に存在することになります。

任意の正整数値を設定できます。暗黙的同期点を追加したくない場合は、**ImplSyncOpenOutput=OFF** を設定してください。

### V 9.0.0.12 OAMLdapConnectTimeout=最大時間

LDAP クライアントがサーバーへの TCP 接続を確立するために待機する最大時間(秒単位)。接続名前リストを介して、複数の LDAP サーバーを提供している場合、タイムアウトは個々の接続試行に適用されるため、このタイムアウトに達すると、名前リスト内の次のエントリーへの接続が試行されます。

### V 9.0.0.12 OAMLdapQueryTimeLimit=最大時間

接続が確立され、LDAP 要求が送信された後、LDAP クライアントがサーバーからの LDAP 要求への応答を受信するのを待機する最大時間を秒単位で指定します。

## 分散キューイングの構成

このセクションでは、キュー定義、チャンネル定義、トリガー操作および同期点プロシージャを含む、IBM MQ インストール間の相互通信の詳細について説明します。

### 始める前に

このセクションを読む前に、チャンネル、キュー、および分散キューイングおよびクラスターで導入されたその他の概念について理解しておく役に立ちます。

### 手順

- 以下のサブトピックにある情報を使用し、分散キューイングを使用してアプリケーションを接続します。
  - [140 ページの『IBM MQ 分散キューイング技法』](#)
  - [161 ページの『分散キュー管理の概要』](#)
  - [163 ページの『別のキュー・マネージャーへのメッセージの送信方法』](#)
  - [184 ページの『チャンネルのトリガー操作』](#)
  - [182 ページの『メッセージの安全性』](#)
  -  [191 ページの『UNIX, Linux, and Windows 上のチャンネルのモニターおよび制御』](#)
  -  [214 ページの『IBM i 上のチャンネルのモニターおよび制御』](#)

### 関連概念

[661 ページの『IBM MQ for z/OS のセットアップ』](#)

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

[728 ページの『他のキュー・マネージャーとの通信のセットアップ』](#)

このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

### 関連タスク

[16 ページの『サーバーとクライアント間の接続の構成』](#)

IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャンネルを定義します。

[236 ページの『キュー・マネージャー・クラスターの構成』](#)

クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

[85 ページの『IBM MQ およびキュー・マネージャー構成情報の変更』](#)

IBM MQ または個々のキュー・マネージャーの動作を、インストール済み環境の要件に合わせて変更できます。

[657 ページの『z/OS でのキュー・マネージャーの構成』](#)

IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

## IBM MQ 分散キューイング技法

このセクションのサブトピックでは、チャンネルを計画する際に役立つ技法について説明します。これらのサブトピックでは、キュー・マネージャー同士を接続して、アプリケーション間のメッセージのフローを管理する方法を計画するのに役立つ技法について説明します。

メッセージ・チャンネルの計画の例については、以下を参照してください。

- **ULW** [UNIX, Linux, and Windows におけるメッセージ・チャネルの計画例](#)
- **IBM i** [IBM i におけるメッセージ・チャネルの計画例](#)
- **z/OS** [z/OS におけるメッセージ・チャネルの計画例](#)
- **z/OS** [キュー共有グループを使用する z/OS におけるメッセージ・チャネルの計画例](#)

## 関連タスク

140 ページの『[分散キューイングの構成](#)』

このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャを含む、IBM MQ インストール間の相互通信の詳細について説明します。

## 関連情報

[チャネル](#)

[メッセージ・キューイングの概要](#)

[分散キューイングとクラスター](#)

[構成情報の例](#)

## メッセージ・フロー制御

メッセージ・フロー制御は、キュー・マネージャーを結ぶメッセージ経路を準備し、維持するタスクです。これは、多数のキュー・マネージャーを介するマルチ・ホップ経路の場合には重要です。このセクションでは、運用システム上のキュー、別名キュー定義、およびメッセージ・チャネルを使用してメッセージ・フロー制御を行う方法を説明します。

メッセージ・フローは、140 ページの『[分散キューイングの構成](#)』で紹介したいくつかの技法で制御します。キュー・マネージャーがクラスター内にある場合は、メッセージ・フローは、141 ページの『[メッセージ・フロー制御](#)』で説明されているように、別の方法を使用して制御されます。**z/OS** キュー・マネージャーがキュー共有グループ内にあり、グループ内キューイング (IGQ) が使用可能である場合は、メッセージ・フローを IGQ エージェントで制御することができます。これらのエージェントについては、[グループ内キューイング](#)で説明します。

以下のオブジェクトを使用して、メッセージ・フロー制御を行うことができます。

- 伝送キュー
- メッセージ・チャネル
- リモート・キュー定義
- キュー・マネージャー別名定義
- 応答先キュー別名定義

キュー・マネージャーおよびキュー・オブジェクトは、[オブジェクト・タイプ](#)で説明されています。メッセージ・チャネルについては、[分散キューイング・コンポーネント](#)に説明します。以下の技法では、このようなオブジェクトを使用して運用システム内のメッセージ・フローを作成します。

- リモート・キューへのメッセージの書き込み
- 特定の伝送キューを介した転送
- メッセージの受信
- メッセージのユーザー・システム通過
- メッセージ・フローの分離
- 別の宛先へのメッセージ・フローの切り替え
- 応答先キュー名から別名への解決

## 注記

このセクションで説明するすべての概念は、ネットワーク内のすべてのノードに関連するものであり、メッセージ・チャンネルの送信側と受信側に関与します。したがって、ほとんどの例ではノードは1つだけが示されています。例外は、メッセージ・チャンネルの反対側で管理者が明示的な協調を行う必要がある例です。

個々の技法の説明に進む前に、解決の概念、およびリモート・キュー定義の3つの使用法を復習しておく役に立ちます。[分散キューイングおよびクラスター](#)を参照してください。

### 関連概念

#### 142 ページの『[伝送ヘッダー内のキュー名](#)』

宛先キュー名は、宛先キューに到達するまで、伝送ヘッダーをメッセージと共に移動します。

#### 142 ページの『[キュー・マネージャーと応答先別名の作成方法](#)』

このトピックでは、リモート・キュー定義を作成できる3つの方法について説明しています。

### 伝送ヘッダー内のキュー名

宛先キュー名は、宛先キューに到達するまで、伝送ヘッダーをメッセージと共に移動します。

アプリケーションによって使用されるキュー名(すなわち論理キュー名)は、キュー・マネージャーによって、宛先キュー名として解決されます。すなわち、物理キュー名です。この宛先キュー名は、宛先キューに到達するまで、別個のデータ域(伝送ヘッダー)をメッセージと共に移動します。その後、伝送ヘッダーは除去されます。

並列のサービス・クラスを作成するときには、このキュー名のキュー・マネージャー部分を変更します。サービス・クラスの迂回経路の終わりに達したときには、キュー・マネージャー名を必ず元の名前に戻してください。

### キュー・マネージャーと応答先別名の作成方法

このトピックでは、リモート・キュー定義を作成できる3つの方法について説明しています。

リモート・キュー定義オブジェクトは、三つの異なる方法で使用されます。[143 ページの表 14](#)では、以下の三つの方法のそれぞれを定義する方法:

- リモート・キュー定義を使用してローカル・キュー名を再定義する。

アプリケーションはキューのオープン時にキュー名だけを提供します。このキュー名は、リモート・キュー定義の名前です。

リモート・キュー定義にはターゲット・キューとキュー・マネージャーの名前が含まれています。オプションで、この定義には、使用される伝送キューの名前を含めることができます。伝送キュー名が指定されないと、キュー・マネージャーは、リモート・キュー定義からとったキュー・マネージャー名を伝送キュー名として使用します。この名前の伝送キューが定義されていないが、デフォルト伝送キューが定義されている場合は、デフォルト伝送キューを使用します。

- リモート・キュー定義を使用してキュー・マネージャー名を再定義する。

アプリケーションまたはチャンネル・プログラムは、キューのオープン時にリモート・キュー・マネージャー名と共にキュー名を指定します。

すでに指定されているリモート・キュー定義がキュー・マネージャー名と同じ名前になっている場合、定義内のキュー名がブランクのままになっていると、キュー・マネージャーは、オープン・コールで指定されたキュー・マネージャー名を定義内のキュー・マネージャー名で置き換えます。

さらにこの定義には、使用される伝送キューの名前を指定することができます。伝送キュー名が指定されないと、キュー・マネージャーは、リモート・キュー定義からとったキュー・マネージャー名を伝送キュー名としてとります。この名前の伝送キューが定義されていないが、デフォルト伝送キューが定義されている場合は、デフォルト伝送キューを使用します。

- リモート・キュー定義を使用して応答先キュー名を再定義する。

アプリケーションは、キューにメッセージを入れるたびに、応答メッセージ用に、キュー・マネージャー名をブランクにしたまま応答先キューの名前を指定することがあります。

応答先キューと同じ名前のリモート・キュー定義を指定すると、ローカル・キュー・マネージャーは、その応答先キュー名を定義で指定されたキュー名によって置き換えます。

キュー・マネージャー名を定義で指定することはできますが、伝送キュー名は指定できません。

表 14. リモート・キュー定義オブジェクトの 3 つの使用法			
使用法	キュー・マネージャー名	キュー名	Transmission queue name
1. リモート・キュー定義 (OPEN 呼び出しで)			
呼び出しで指定	ブランクまたはローカル QM	(*) 必須	適用外
定義で指定	必須	必須	オプション
2. キュー・マネージャー別名 (OPEN 呼び出しで)			
呼び出しで指定	(*) 必須かつ非ローカル QM	必須	適用外
定義で指定	必須	ブランク	オプション
3. 応答先キュー別名 (PUT 呼び出しで)			
呼び出しで指定	ブランク	(*) 必須	適用外
定義で指定	オプション	オプション	ブランク

注: (\*) は、この名前が定義オブジェクトの名前であることを示しています。

詳しくは、[キュー名の解決](#)を参照してください。

## リモート・キューへのメッセージの書き込み

リモート・キュー定義オブジェクトを使用して、キュー名を隣接するキュー・マネージャーに通じる伝送キューとして解決できます。

分散キューイング環境では、伝送キューとチャネルは、メッセージが運用しているローカル・システムのアプリケーションから出される場合にも、隣接システムからチャネルを介して到達する場合にも、あるロケーションに送られるすべてのメッセージの中心拠点になります。[144 ページの図 7](#)では、アプリケーションが 'QA\_norm' という名前の論理キューにメッセージを入れています。このネーム・レゾリューションでは、'QA\_norm' というリモート・キュー定義を使用して伝送キュー QMB を選択します。次に、'QA\_norm at QMB' というメッセージに伝送ヘッダーを追加します。

隣接システムから 'Channel\_back' に到着するメッセージには伝送ヘッダーが付いていて、例えば 'QA\_norm at QMB' などの物理キュー名を示します。これらのメッセージは、未変更のまま伝送キュー QMB に入ります。

チャネルは、このメッセージを隣接するキュー・マネージャーに送ります。

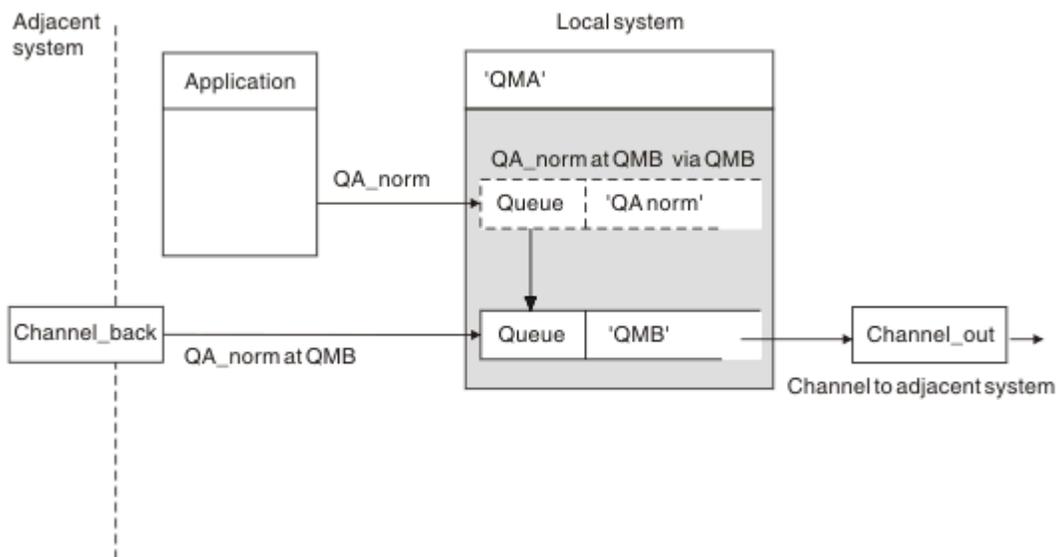


図 7. リモート・キュー定義が使用されて、キュー名が解決され、隣接するキュー・マネージャーに通じる伝送キューが得られます

IBM MQ システム管理者は、以下を行ってください。

- 隣接システムからのメッセージ・チャンネルを定義する。
- 隣接システムに向かうメッセージ・チャンネルを定義する。
- 伝送キュー QMB を作成する。
- リモート・キュー・オブジェクト 'QA\_norm' を定義し、アプリケーションによって使用されるキュー名を解決して宛先キュー名、宛先キュー・マネージャー名、および伝送キュー名が得られるようにする。

クラスター化環境では、ローカル・キュー・マネージャーにクラスター受信側チャンネルを定義するだけで十分です。伝送キューまたはリモート・キュー・オブジェクトを定義する必要はありません。[クラスター](#)を参照してください。

## 解決に関する補足説明

リモート・キュー定義により、物理的な宛先キュー名およびキュー・マネージャー名が定義されます。これらの名前は、メッセージの伝送ヘッダーに書き込まれます。

隣接システムからの着信メッセージでは、既に元のキュー・マネージャーによってこのタイプのネーム・レゾリューションが実行されています。したがって、物理的な宛先キュー名およびキュー・マネージャー名が示されている伝送ヘッダーがあります。これらのメッセージは、リモート・キュー定義による影響を受けません。

## 伝送キューの選択

リモート・キュー定義を使用すると、別の伝送キューが同じ隣接キュー・マネージャーにメッセージを送信することができます。

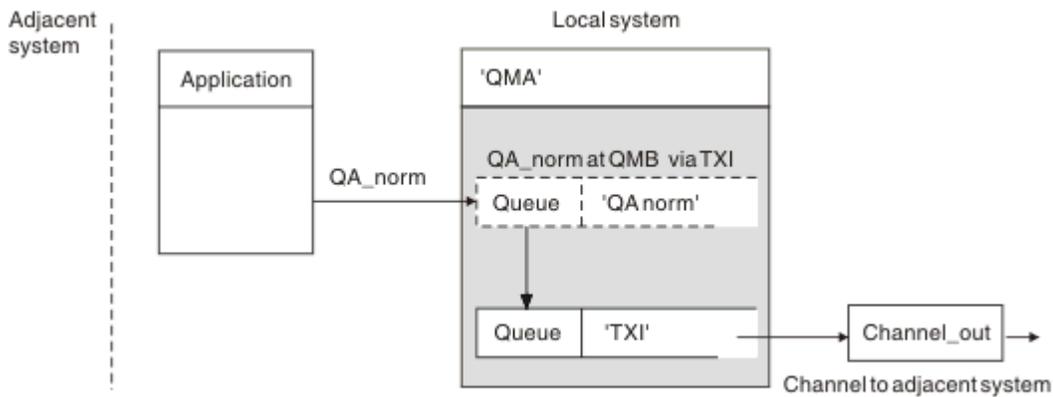


図 8. リモート・キュー定義による異なる伝送キューの使用可能化

分散キューイング環境では、あるチャンネルから別のチャンネルにメッセージ・フローを変更する必要がある場合は、同じシステム構成を 144 ページの図 7 143 ページの『リモート・キューへのメッセージの書き込み』してください。この 145 ページの図 8 では、リモート・キュー定義を使用して、異なる伝送キューを経由するメッセージを、異なるチャンネルを介して、同じ隣接キュー・マネージャーに送信する方法を示します。

145 ページの図 8 に示されている構成の場合には、リモート・キュー・オブジェクト「QA\_norm」および伝送キュー「TX1」を提供する必要があります。「QA\_norm」を提供して、リモート・キュー・マネージャーで「QA\_norm」キュー、伝送キュー「TX1」、およびキュー・マネージャー「QMB\_priority」を選択する必要があります。隣接システムへのチャンネルの定義で、「TX1」を指定します。

メッセージは、「QA\_norm at QMB\_priority」という内容の伝送ヘッダーと共に伝送キュー「TX1」に入り、チャンネルを介して隣接システムに送られます。

channel\_back には、キュー・マネージャー別名が必要なため、この図には示されていません。

クラスター化環境では、伝送キューまたはリモート・キュー定義を定義する必要はありません。詳細については、237 ページの『クラスター・キューの定義』を参照してください。

## メッセージの受信

他のキュー・マネージャーからメッセージを受信するようにキュー・マネージャーを構成できます。意図されないネーム・レゾリューションが起きないことを確認する必要があります。

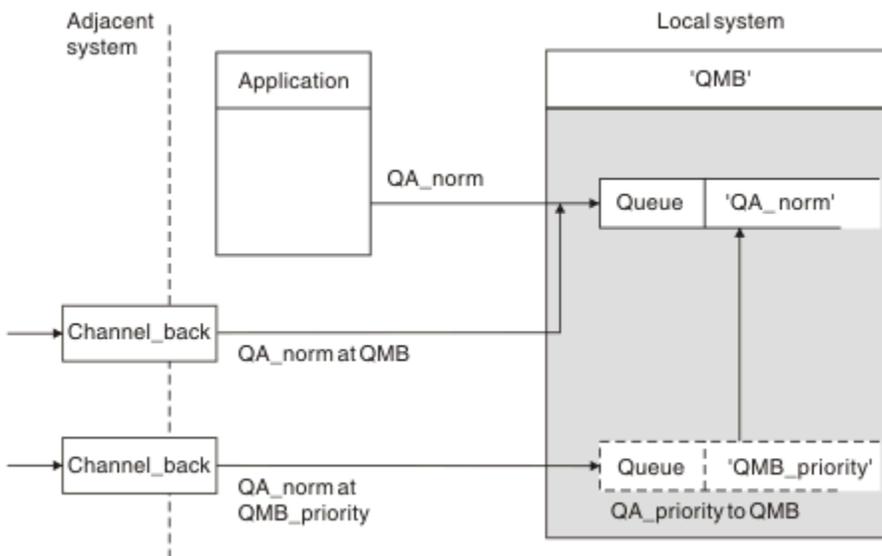


図 9. メッセージの直接受信によるキュー・マネージャー別名の解決

システム管理者は、メッセージを送信する準備のほかに、隣接キュー・マネージャーからのメッセージの受信を準備する必要があります。受信されたメッセージの伝送ヘッダーには、宛先キュー・マネージャーとキューの物理名が指定されています。これらのメッセージは、キュー・マネージャー名とキュー名の両方が指定されたローカル・アプリケーションからのメッセージと同じように扱われます。この処理のため、運用システムに入ってくるメッセージに対して意図されないネーム・レゾリューションが行われないようにしなければなりません。このシナリオについては [145 ページの図 9](#) を参照してください。

この構成では、次のものを準備する必要があります。

- 隣接キュー・マネージャーからメッセージを受信するメッセージ・チャンネル
- 着信メッセージ・フロー 'QMB\_priority' を解決してローカル・キュー・マネージャー名 'QMB' を得るための、キュー・マネージャー別名定義
- ローカル・キュー 'QA\_norm' (存在していない場合)

## キュー・マネージャー別名の受信

この図におけるキュー・マネージャー別名定義の使用では、別の宛先キュー・マネージャーは選択されていません。'QMB\_priority' とアドレッシングされて、このローカル・キュー・マネージャーを通過するメッセージは、キュー・マネージャー 'QMB' に送られます。このキュー・マネージャー別名が使用されて、別のメッセージ・フローが作成されます。

## メッセージのユーザー・システム通過

ローケーション名の使用、キュー・マネージャーの別名の使用、または伝送キューの選択の 3 つの方法で、メッセージにシステムを通過させることができます。

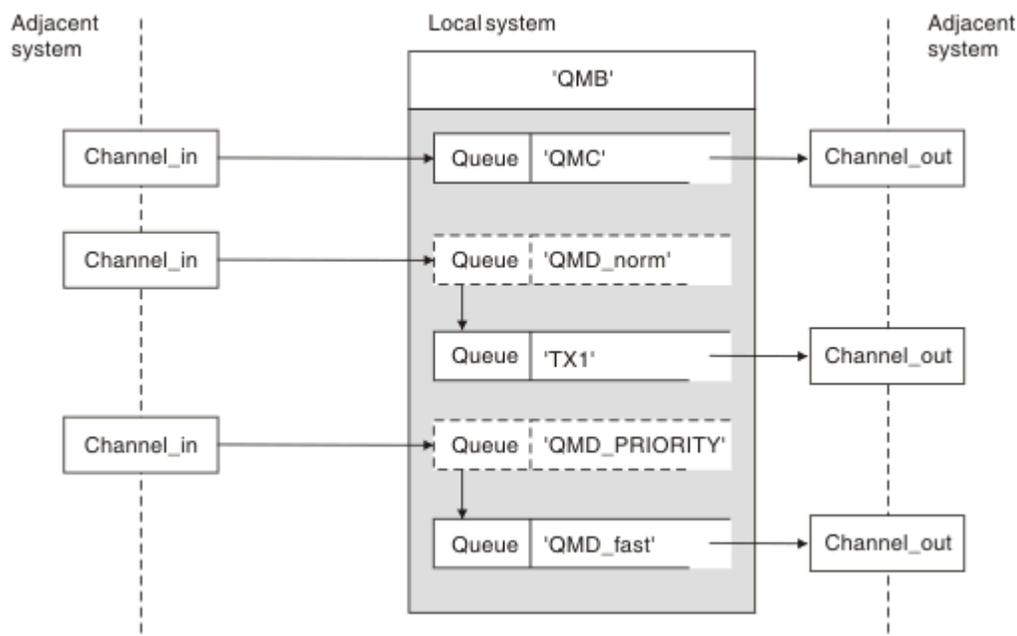


図 10. メッセージが運用システムを通過する 3 つの方法

[145 ページの『メッセージの受信』](#)の [145 ページの図 9](#) に示されている技法は、別名フローがどのようにキャプチャーされるかを示しています。[146 ページの図 10](#) は、前述の手法をまとめることによって、ネットワークが構築される方法を示しています。

この構成で示されるチャンネルは、次のようなさまざまな宛先を指定された 3 つのメッセージを送達します。

1. QMC の QB
2. QMD\_norm の QB
3. QMD\_PRIORITY の QB

最初のメッセージ・フローは変更せずにシステムを通過させる必要があります。2番目のメッセージ・フローは別の伝送キューおよびチャンネルを通過させる必要があります。2番目のメッセージ・フローでは、別名キュー・マネージャー名 `QMD_norm` のメッセージをキュー・マネージャー `QMD` に解決する必要もあります。3番目のメッセージ・フローでは、別の伝送キューが選択されるだけで、それ以外の変更は行われません。

クラスター化環境では、メッセージがクラスター伝送キューを通過します。通常、単一伝送キュー `SYSTEM.CLUSTER.TRANSMIT.QUEUE` は、メンバーになっているクラスター内のすべてのキュー・マネージャーにメッセージを転送します。キュー・マネージャーのクラスターを参照してください。キュー・マネージャーがメンバーとなっているクラスター内のすべてのキュー・マネージャー、またはその一部に使用する別個の伝送キューを定義できます。

次の方法は、分散キューイング環境に適用できる技法を説明しています。

## これらの方法の使用法

これらの構成では、次のものを準備する必要があります。

- 入力チャンネル定義
- 出力チャンネル定義
- 伝送キュー
  - `QMC`
  - `TX1`
  - `QMD_fast`
- キュー・マネージャー別名定義
  - `TX1` 経由の `QMD` への `QMD_norm` を指定した `QMD_norm`
  - `QMD_fast` 経由の `QMD_PRIORITY` への `QMD_PRIORITY` を指定した `QMD_PRIORITY`

注：この例で示されているいずれのメッセージ・フローでも、宛先キューは変更されていません。キュー・マネージャー名の別名では、メッセージ・フローの分離が行われています。

### 方法 1: 着信ロケーション名の使用

伝送ヘッダーに別のロケーション名 (例えば、`QMC`) が格納されたメッセージを受信するとします。その場合、最も単純な構成となるのは、その名前 (`QMC`) を設定した伝送キューを作成することです。この伝送キューを処理するチャンネルが、メッセージをそのままの状態での宛先に配信します。

### 方法 2: キュー・マネージャーの別名の使用

2番目の方法は、キュー・マネージャー別名オブジェクト定義を使用することです。ただし、新しいロケーション名 `QMD` および特定の伝送キュー `TX1` を指定します。この処置を行うと、

- キュー・マネージャー名の別名 `QMD_norm` でセットアップされた別名メッセージ・フローが終了します。これは、指定されたサービス・クラス `QMD_norm` です。
- メッセージの伝送ヘッダーを `QMD_norm` から `QMD` に変更します。

### 方法 3: 伝送キューの選択

3番目の方法は、宛先ロケーション `QMD_PRIORITY` と同じ名前を指定してキュー・マネージャー別名オブジェクトを定義することです。このキュー・マネージャー別名定義を使用して、特定の伝送キュー `QMD_fast`、つまり別チャンネルを選択します。これらのメッセージの伝送ヘッダーは、未変更のままになります。

## メッセージ・フローの分離

キュー・マネージャー別名を使用して、同じキュー・マネージャーにメッセージを送信するための別個のメッセージ・フローを作成できます。

分散キューイング環境では、いくつかの理由で、同じキュー・マネージャーに送られるメッセージを別のメッセージ・フローに分離する必要があります。以下に例を示します。

- 大中小のそれぞれの大きさのメッセージに合わせて別々のフローを用意する必要がある場合。この必要はクラスター化環境にも当てはまり、その場合、重なり合ったクラスターを作成することもできます。そのようにする理由は、次のようにいろいろあります。
  - 組織ごとに独自の管理ができるようにする。
  - 独立したアプリケーションを個別に管理できるようにする。
  - サービス・クラスを作成するため。例えば、STUDENTS というクラスターのサブセットとして STAFF というサブセットを作成できます。STAFF クラスター内で公示されたキューへメッセージを入れた場合は、制限されたチャンネルが使用されます。STUDENTS クラスター内で公示されたキューへメッセージを入れた場合は、一般チャンネルと制限されたチャンネルのどちらかが使用されます。
  - テスト環境または実稼働環境を作成する。
- 着信メッセージを、ローカルで生成されたメッセージのパスと異なるパスで経路指定する必要がある場合。
- インストール先で特定の時刻 (例えば、深夜) にメッセージの移動をスケジュールする必要があり、そのメッセージを、スケジュールされるまで用意されたキューに保管しておく必要がある場合。

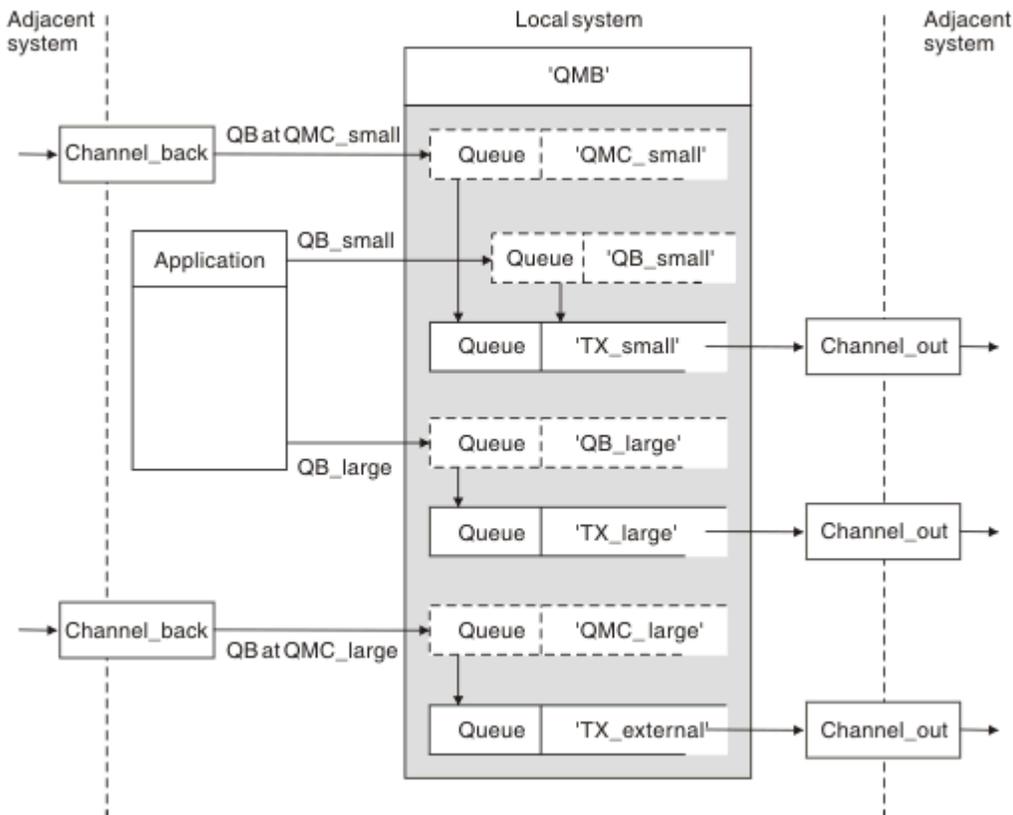


図 11. メッセージ・フローの分離

148 ページの図 11 の例で、2つの着信フローはキュー・マネージャー名の別名 'QMC\_small' および 'QMC\_large' を宛先としています。これらのフローにキュー・マネージャー別名定義を提供して、ローカル・キュー・マネージャーのためにこれらのフローを獲得できるようにしてください。2つのリモート・キューをアドレッシングするアプリケーションがあり、これらのメッセージ・フローを別個に分けておく必要があるものとします。ロケーションには同じ値 'QMC' を指定し、伝送キューには別の値を指定して、

2つのリモート・キュー定義を提供します。この定義は、フローが分離されます。伝送ヘッダーに同じ宛先キュー・マネージャー名が入っているため、相手側での追加処置は必要ありません。ローカル・システムでは次のものを指定してください。

- 着信チャンネル定義
- 2つのリモート・キュー定義 QB\_small および QB\_large
- 2つのキュー・マネージャー別名定義 QMC\_small および QMC\_large
- 3つの送信チャンネル定義
- 3つの伝送キュー TX\_small、TX\_large、および TX\_external

## 隣接システムとの調整

別個のメッセージ・フローを作成するためにキュー・マネージャー別名を使用するときには、メッセージ・チャンネルのリモート側のシステム管理者と連携してこの作業を行い、リモート側で対応するキュー・マネージャーの別名が有効になるようにする必要があります。

## さまざまなロケーションに送られるメッセージの集中

さまざまなロケーションにあてられたメッセージを単一のチャンネルに集中させます。

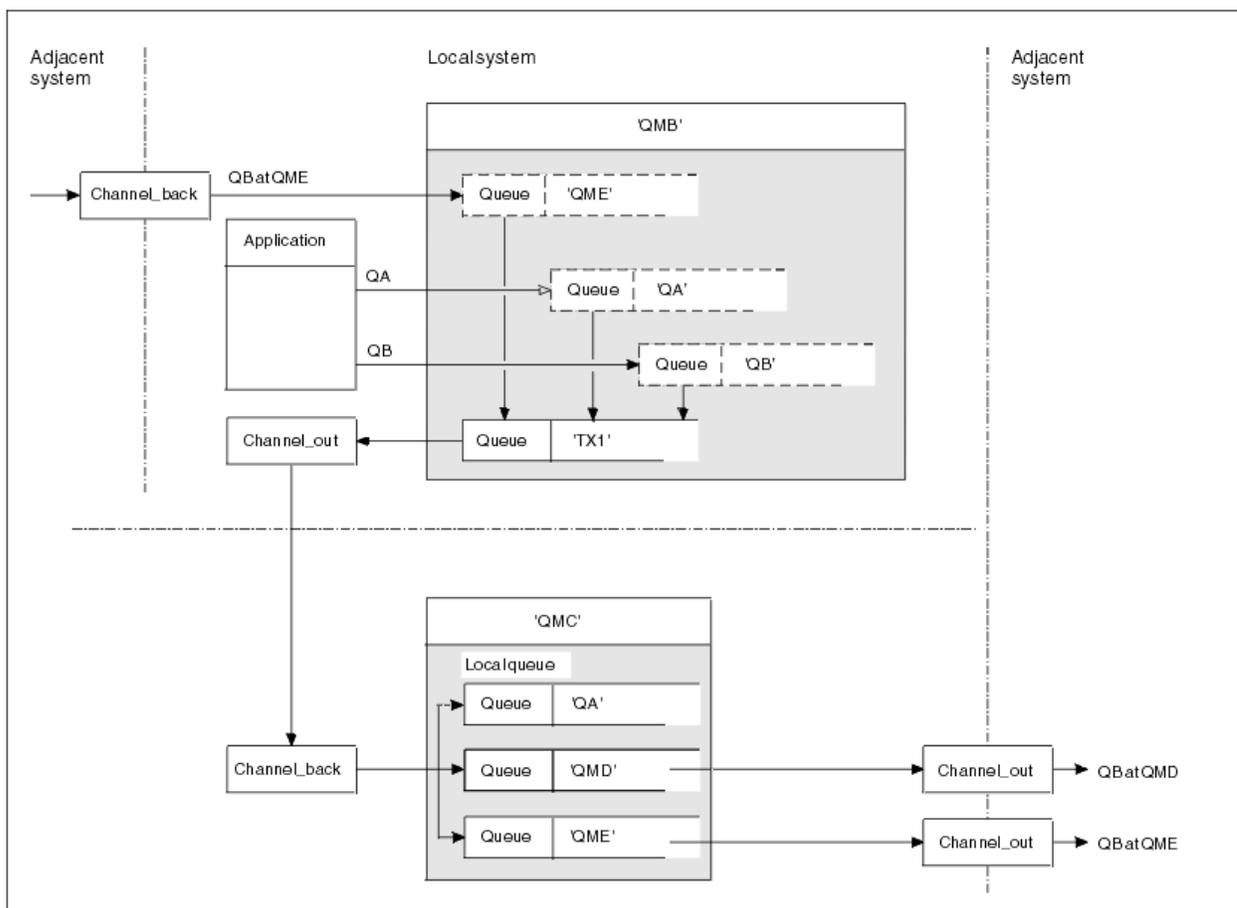


図 12. チャンネルへのメッセージ・フローの結合

149 ページの図 12 は、さまざまなロケーションにあてられたメッセージを 1 つのチャンネルに集中させる分散キューイングの技法を示しています。可能な使用法には次の 2 つがあります。

- ゲートウェイを使用してメッセージ・トラフィックを集中させる
- ノード間で通信容量の高い高速経路を使用する

この例では、ローカル・システムおよび隣接システムのさまざまなソースからメッセージが発行されます。これらメッセージは、宛先キューも取り扱うキュー・マネージャーも異なっていますが、いずれも伝送キュー 'TX1' を介してキュー・マネージャー QMC に送られています。キュー・マネージャー QMC は、宛先に応じてメッセージを送信します。あるものは伝送キュー 'QMD' に、さらにキュー・マネージャー QMD に伝送されるように設定されます。別のものは、伝送キュー 'QME' に、さらにキュー・マネージャー QME に伝送されるように設定されます。他のメッセージは、ローカル・キュー 'QA' に書き込まれます。

以下を指定してください。

- チャンネル定義
- 伝送キュー TX1
- リモート・キュー定義
  - 'TX1 経由の QMC の QA' を指定した QA
  - 'TX1 経由の QMD の QB' を指定した QB
- キュー・マネージャー別名定義
  - 'TX1 経由の QME' を指定した QME

QMC を構成する管理者アシスタントは以下を指定してください。

- 同じチャンネル名を指定した受信側チャンネル定義
- 関連する送信側チャンネル定義を指定した伝送キュー QMD
- 関連する送信側チャンネル定義を指定した伝送キュー QME
- ローカル・キュー・オブジェクト QA

## 別の宛先へのメッセージ・フローの転送

キュー・マネージャー別名および伝送キューを使用して、特定のメッセージの宛先を再定義できます。

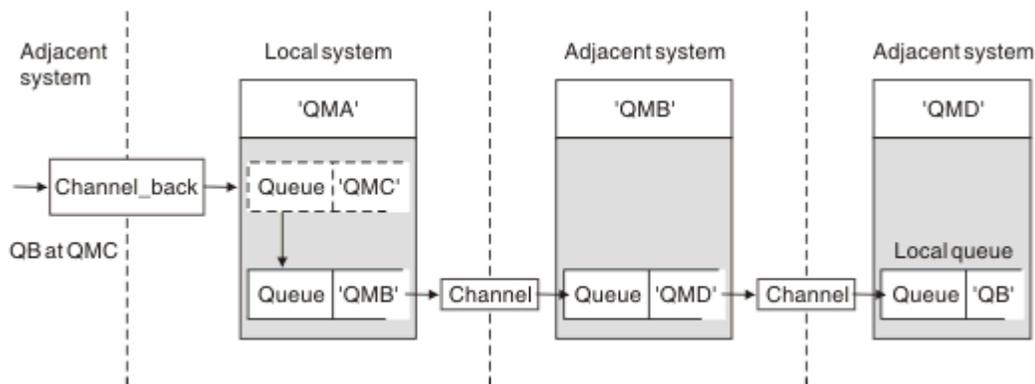


図 13. 別の宛先へのメッセージ・ストリームの転送

150 ページの図 13 では、特定のメッセージの宛先を再定義する方法を示しています。QMA への着信メッセージは 'QMC の QB' に宛てて送られます。通常は QMA に到達して、QMC に至るチャンネルの一部である QMC という伝送キューに入ります。QMA はこのメッセージを QMD に転送する必要がありますが、QMD には QMB を通らなければ到達できません。この方法は、あるロケーションから別のロケーションにサービスを移動する場合に便利であり、加入者は、メッセージが新しいアドレスに合わせて調整されるまでの間、一時的な方法でメッセージの送信を続けることができます。

特定のキュー・マネージャーに向けられた着信メッセージを別のキュー・マネージャーに再経路指定する方法では、次のものを使用します。

- 宛先キュー・マネージャーを別のキュー・マネージャーに変更して、隣接システムに至る伝送キューを選択するための、キュー・マネージャー別名
- 隣接キュー・マネージャーで使用される伝送キュー

- 宛先キュー・マネージャーに向けてさらにルーティングするための、隣接キュー・マネージャーにおける伝送キュー

以下を指定してください。

- Channel\_back 定義
- QMB 経由の QMD の QB を指定したキュー・マネージャー別名オブジェクト定義 QMC
- Channel\_out 定義
- 関連の伝送キュー QMB

QMB を構成する管理者アシスタントは以下を指定してください。

- 対応する channel\_back 定義
- 伝送キュー QMD
- QMD に通じる関連チャネル定義

クラスター化環境の中では、別名が使用できます。詳細については、[332 ページの『キュー・マネージャー別名とクラスター』](#)を参照してください。

## 配布先リストへのメッセージの送信

単一の MQPUT 呼び出しを使用して、アプリケーションが複数の宛先にメッセージを送信するようにできます。

z/OS を除くすべてプラットフォーム上の IBM MQ では、アプリケーションは 1 つの MQPUT 呼び出しで複数の宛先にメッセージを送信できます。分散キューイング環境とクラスター化環境の両方でそれを行うことができます。[配布先リスト](#)で説明されているように、宛先を配布先リストで定義する必要があります。

すべてのキュー・マネージャーが配布先リストをサポートしているとは限りません。MCA は、パートナーとの接続を確立するときに、パートナーが配布先リストをサポートしているかどうかを判別し、それに応じて伝送キューにフラグを設定します。アプリケーションが、配布先リストに指定されているメッセージを送信しようとしたところ、パートナーが配布先リストをサポートしていない場合、送信側の MCA がメッセージを代行受信して、各宛先ごとに 1 回、伝送キューに書き込みます。

受信側の MCA は、配布先リストに送信したメッセージがすべての宛先で安全に受信されたことを確認します。いずれかの宛先で受信が失敗した場合、MCA は、どれが失敗したかを特定します。その後、例外報告書を作成し、メッセージの送信の再試行を行うことができます。

## 応答先キュー

応答先キューを使用すると、完全なりモート・キュー処理ループを作成できます。

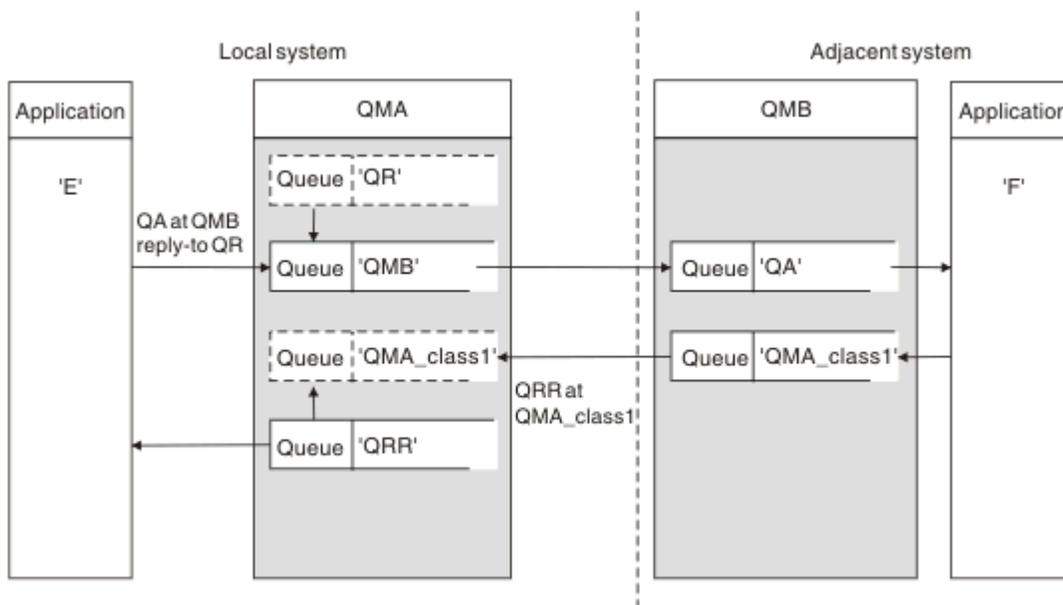


図 14. 書き込み (PUT) 呼び出し時の応答先キュー名の置換

応答先キューを使用する完全なリモート・キュー処理ループが [152 ページの図 14](#) に示されています。このループは、分散キューイング環境とクラスター化環境の両方に適用されます。詳細については、[159 ページの表 18](#) を参照してください。

アプリケーションは QMB にある QA をオープンし、そのキューにメッセージを書き込みます。このメッセージには QR という応答先キュー名が指定され、キュー・マネージャー名は指定されません。キュー・マネージャー QMA は応答先キュー・オブジェクト QR を検出し、その中から QRR という別名とキュー・マネージャー名 QMA\_class1 を取り出します。これらの名前はメッセージの応答先フィールドに書き込まれます。

QMB にあるアプリケーションから発行されたメッセージは、QMA\_class1 にある QRR に向けてアドレッシングされます。キュー・マネージャーによってキュー・マネージャー別名定義 QMA\_class1 が使用され、メッセージがそのキュー・マネージャー自体およびキュー QRR に流れるようになります。

このシナリオでは、応答メッセージのサービス・クラスを選択する機能を、アプリケーションに与える方法を示しています。クラスは、QMB 上の伝送キュー QMA\_class1 と、QMA 上のキュー・マネージャー別名定義 QMA\_class1 を併用することによって実装されています。この方法で、アプリケーションの応答先キューを変更して、アプリケーションには影響を与えずにフローを分離できます。アプリケーションでは、常に、この特定のサービス・クラスの QR を選択します。この応答先キュー定義 QR を使用して、サービス・クラスを変更することができます。

以下を作成してください。

- 応答先キュー定義 QR
- 伝送キュー・オブジェクト QMB
- Channel\_out 定義
- Channel\_back 定義
- キュー・マネージャー別名定義 QMA\_class1
- ローカル・キュー・オブジェクト QRR (まだ存在していない場合)

隣接システムの管理者アシスタントは、以下を作成してください。

- 受信チャンネル定義
- 伝送キュー・オブジェクト QMA\_class1
- 関連する送信チャンネル
- ローカル・キュー・オブジェクト QA

アプリケーション・プログラムでは以下のものを使用します。

- 書き込み呼び出しで応答先キュー名 QR
- 読み取り呼び出しでキュー名 QRR

このようにして、アプリケーションの関与なしで、必要に応じてサービス・クラスを変更することができます。応答先別名「QR」を、伝送キュー「QMA\_class1」およびキュー・マネージャー別名「QMA\_class1」と共に変更します。

メッセージがキューに書き込まれるときに応答先別名オブジェクトが検出されなかった場合は、ブランクの応答先キュー・マネージャー名フィールドにローカル・キュー・マネージャー名が挿入されます。応答先キュー名は未変更のままになります。

## 解決の制約事項

元のメッセージが書き込まれたときに「QMA」上で応答先キューのネーム・レゾリューションが行われているため、「QMB」上ではそれ以上のネーム・レゾリューションは実行できません。したがって、応答側アプリケーションは、応答先キューの物理名を指定してメッセージをキューに書き込みます。

応答先キューとして使用する名前と、戻りメッセージが検出される実際のキューの名前とは異なることを、アプリケーションが認識している必要があります。

例えば、アプリケーションが使用するための2つのサービス・クラスが、「C1\_alias」および「C2\_alias」という応答先キュー別名で指定された場合、アプリケーションはメッセージ書き込み呼び出しでこれらの名前を応答先キュー名として使用します。ただし、アプリケーションは、実際にはメッセージが、「C1\_alias」の場合はキュー「C1」に、「C2\_alias」の場合はキュー「C2」に入ることを予期します。

ただし、アプリケーションは、応答先別名キューに対して照会呼び出しを出して、応答メッセージを取得するために使用する必要のある実際のキューの名前を調べることができます。

## 関連概念

### [142 ページの『キュー・マネージャーと応答先別名の作成方法』](#)

このトピックでは、リモート・キュー定義を作成できる3つの方法について説明しています。

### [153 ページの『応答先キュー別名の例』](#)

この例では、戻されるメッセージ用に別の経路(伝送キュー)を選択するために応答先別名を使用する方法が示されています。この機能を使用するためには、アプリケーションと連携して応答先キュー名を変更する必要があります。

### [155 ページの『この例の作業内容』](#)

例およびキュー・マネージャーが応答先キュー別名を使用する方法についての説明。

### [156 ページの『応答先キュー別名の概要』](#)

アプリケーションがリモート・キューにメッセージを書き込んでから、同じアプリケーションが応答先キューの別名から応答メッセージを削除するまでのプロセスのワークスルーです。

## 応答先キュー別名の例

この例では、戻されるメッセージ用に別の経路(伝送キュー)を選択するために応答先別名を使用する方法が示されています。この機能を使用するためには、アプリケーションと連携して応答先キュー名を変更する必要があります。

[154 ページの図 15](#) に示すように、応答メッセージが、伝送キュー、チャンネル、およびキュー・マネージャー別名の入った戻り経路を利用できなければなりません。

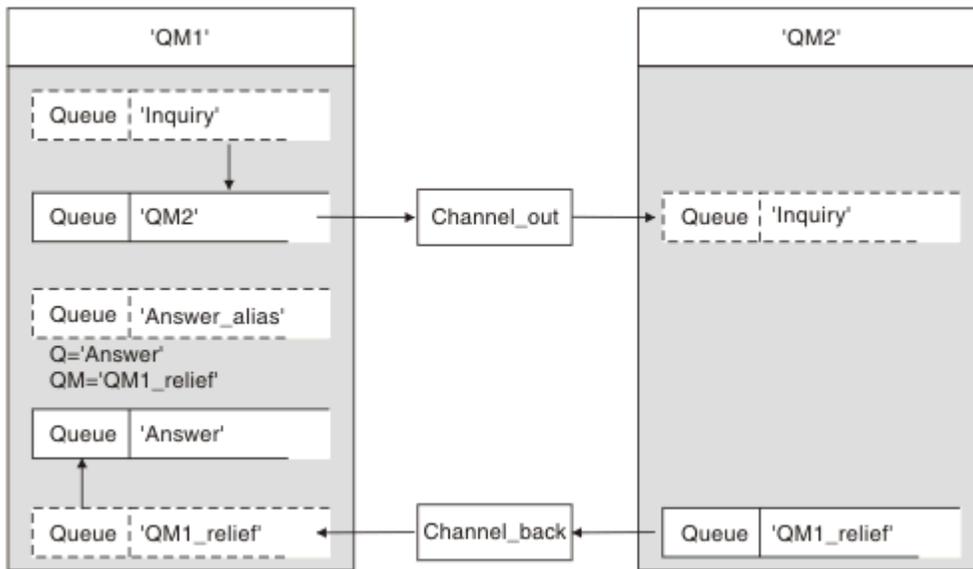


図 15. 応答先キュー別名の例

この例は、'QM2'にあるサーバー・アプリケーションにメッセージを送る、'QM1'にある要求側アプリケーションを表しています。サーバーのメッセージは、伝送キュー 'QM1\_relief' を使用して、代替チャンネルを介して返されます (デフォルトの戻りチャンネルは、伝送キュー 'QM1' によって提供されます)。

この応答先キュー別名には、'Answer\_alias' というリモート・キュー定義が特定の方法で使用されています。QM1にあるアプリケーションは、キュー 'Inquiry' に入れるすべてのメッセージの応答先フィールドに、この 'Answer\_alias' という名前を書き込みます。

応答先キュー定義 'Answer\_alias' は、'Answer at QM1\_relief' として定義されます。QM1にあるアプリケーションは、応答が 'Answer' というローカル・キューに入るものと想定します。

QM2にあるサーバー・アプリケーションは、受信したメッセージの応答先フィールドを使用して、QM1にある要求側に対する応答メッセージ用のキューおよびキュー・マネージャーの名前を取得します。

### この例の QM1 で使用されている定義

QM1のIBM MQシステム管理者は、他のオブジェクトと共に応答先キュー 'Answer' が作成されるようにしなければなりません。'\*'が付いているキュー・マネージャー別名の名前は、'\*'が付いている応答先のキュー別名定義のキュー・マネージャー名と一致している必要があります。

オブジェクト	定義	
ローカル伝送キュー	QM2	
リモート・キュー定義	オブジェクト名	照会
	リモート・キュー・マネージャー名	QM2
	リモート・キュー名	照会
	Transmission queue name	QM2 (DEFAULT)
キュー・マネージャーの別名	オブジェクト名	QM1_relief *
	キュー・マネージャー名	QM1
	キュー名	(ブランク)
応答先キュー別名	オブジェクト名	Answer_alias
	リモート・キュー・マネージャー名	QM1_relief *

オブジェクト	定義	
	リモート・キュー名	Answer

## QM1での書き込み定義

アプリケーションは応答先フィールドに応答先キュー別名を入れ、キュー・マネージャー名フィールドをブランクのままにします。

フィールド	内容
キュー名	照会
キュー・マネージャー名	(ブランク)
応答先キュー名	Answer_alias
応答先キュー・マネージャー	(ブランク)

## この例のQM2で使用されている定義

QM2のIBM MQシステム管理者は、着信メッセージのためのローカル・キューが存在していることを確認し、正しい名前の伝送キューを応答メッセージで利用できるようにしなければなりません。

オブジェクト	定義
ローカル・キュー	照会
伝送キュー	QM1_relief

## QM2での書き込み定義

QM2にあるアプリケーションは元のメッセージから応答先キュー名とキュー・マネージャー名を取り出し、応答先キューに応答メッセージを書き込むときにはその名前を使用します。

フィールド	内容
キュー名	Answer
キュー・マネージャー名	QM1_relief

## この例の作業内容

例およびキュー・マネージャーが応答先キュー別名を使用する方法についての説明。

この例で、QM1にある要求側アプリケーションは常に、書き込み呼び出しの関連フィールド内で応答先キューとして'Answer\_alias'を使用します。それらは常に'Answer'というキューからメッセージを取り出します。

応答先キュー'Answer'の名前および戻り経路'QM1\_relief'の名前を変更するために、応答先キュー別名定義をQM1のシステム管理者が使用できるようになっています。

QM1のアプリケーションは応答がキューに入ることを予想しているため、このキュー名'Answer'を変更することは通常は有効ではありません。ただし、QM1システム管理者は、必要に応じて戻り経路(サービス・クラス)を変更できます。

## キュー・マネージャーが応答先キュー別名を使用する方法

キュー・マネージャーQM1は、アプリケーションによって書き込み呼び出しで指定された応答先キュー名が応答先キュー別名と同じであって、キュー・マネージャー部分がブランクになっている場合は、応答先キュー別名から定義を取り出します。

キュー・マネージャーは、書き込み呼び出しで指定された応答先キュー名を定義内の応答先キュー名によって置き換えます。また、書き込み呼び出しでブランクになっているキュー・マネージャー名を定義内のキュー・マネージャー名によって置き換えます。

これらの名前は、メッセージ記述子に入ってメッセージと共に送られます。

表 15. 応答先キュー別名		
フィールド名	書き込み呼び出し	伝送ヘッダー
応答先キュー名	Answer_alias	Answer
応答先キュー・マネージャー名	(ブランク)	QM1_relief

## 応答先キュー別名の概要

アプリケーションがリモート・キューにメッセージを書き込んでから、同じアプリケーションが応答先キューの別名から応答メッセージを削除するまでのプロセスのワークスルーです。

この例を完成するために、プロセスについて説明します。

1. アプリケーションが 'Inquiry' というキューをオープンして、その中にメッセージを書き込みます。アプリケーションは、メッセージ記述子の応答先フィールドに次の値を入れます。

<b>応答先キュー名</b>	<b>Answer_alias</b>
応答先キュー・マネージャー名	(ブランク)

2. キュー・マネージャー 'QM1' が 'Answer\_alias' という名前のリモート・キュー定義の有無を調べて、ブランクのキュー・マネージャー名に回答します。この名前のリモート・キュー定義がない場合、キュー・マネージャーは、メッセージ記述子の応答先キュー・マネージャー・フィールドにそれ自身の名前 'QM1' を入れます。
3. 'Answer\_alias' という名前のリモート・キュー定義が見つかった場合、キュー・マネージャーは、その定義からキュー名 ('Answer') およびキュー・マネージャー名 ('QM1\_relief') を取り出します。次に、それをメッセージ記述子の応答先フィールドに入れます。
4. キュー・マネージャー「QM1」は、リモート・キュー定義「Inquiry」を使用して、送信先キューがキュー・マネージャー「QM2」であるかどうかを判別し、メッセージが伝送キュー「QM2」に置かれていることを判別します。「QM2」は、キュー・マネージャー「QM2」のキューに向けられたメッセージのデフォルトの送信キュー名である。
5. キュー・マネージャー 'QM1' は、メッセージをこの伝送キューに入れるときに、メッセージに伝送ヘッダーを追加します。このヘッダーには、宛先キューの名前 'Inquiry' と宛先キュー・マネージャー 'QM2' が入っています。
6. メッセージはキュー・マネージャー 'QM2' に到達し、ローカル・キュー 'Inquiry' に入ります。
7. アプリケーションはこのキューからメッセージを受け取り、メッセージを処理します。そして、応答メッセージを用意して、元のメッセージのメッセージ記述子から得られた応答先キュー名にその応答メッセージを入れます。

<b>応答先キュー名</b>	<b>Answer</b>
応答先キュー・マネージャー名	QM1_relief

8. キュー・マネージャー 'QM2' は書き込みコマンドを実行します。さらに、キュー・マネージャー名 'QM1\_relief' がリモート・キュー・マネージャーであることを調べ、同じ名前の伝送キュー 'QM1\_relief' にメッセージを入れます。このメッセージには、宛先キューの名前 'Answer' と宛先キュー・マネージャー 'QM1\_relief' の入った伝送ヘッダーが付けられます。
9. メッセージはキュー・マネージャー 'QM1' に転送されます。キュー・マネージャーは、キュー・マネージャー名 'QM1\_relief' が別名であることを認識し、別名定義 'QM1\_relief' から物理キュー・マネージャー名 'QM1' を取り出します。
10. キュー・マネージャー 'QM1' は、伝送ヘッダーに収められているキュー名 'Answer' にメッセージを入れます。
11. アプリケーションがキュー 'Answer' から応答メッセージを取り出します。

## ネットワーキングの考慮事項

分散キューイング環境では、メッセージの宛先は、キュー名とキュー・マネージャー名だけでアドレッシングされるため、特定の規則が適用されます。

1. キュー・マネージャー名が指定され、その名前がローカル・キュー・マネージャーの名前と異なっている場合、次のいずれかが成立すること。
  - 伝送キューは同じ名前が使用可能でなければならない。この伝送キューが別のキュー・マネージャーにメッセージを移動するためのメッセージ・チャンネルの一部でなければならない。
  - キュー・マネージャー名を解決して同じキュー・マネージャー名または別のキュー・マネージャー名、およびオプションの伝送キューを得るためにキュー・マネージャー別名定義が存在していること。
  - 伝送キュー名が解決できないで、デフォルト伝送キューが定義されている場合は、デフォルト伝送キューが使用される。
2. キュー名だけが指定された場合は、同じ名前の任意のタイプのキューがローカル・キュー・マネージャーで使用可能でなければなりません。このキューは、隣接キュー・マネージャーへの伝送キュー、キュー・マネージャー名、およびオプションの伝送キューを解決する、リモート・キュー定義である場合があります。

これがクラスタリング環境でどのように機能するかを調べるには、[クラスター](#)を参照してください。

**Z/OS** キュー共有グループ内でキュー・マネージャーを実行している場合に、グループ内キューイング (IGQ) が使用可能であると、SYSTEM.QSG.TRANSMIT.QUEUE を使うことができます。詳しくは、[グループ内キューイング](#)を参照してください。

分散キューイング環境で、あるキュー・マネージャーから別のキュー・マネージャーへメッセージを移動するメッセージ・チャンネルのシナリオについて考えてみます。

移動されるメッセージは、ネットワーク内の他のいずれかのキュー・マネージャーから出されたものであり、それらのメッセージのうちには、未知のキュー・マネージャー名が宛先として指定されて到達するものもあります。例えば、キュー・マネージャー名が変更されたりシステムから削除されたりした場合、この問題が発生することがあります。

チャンネル・プログラムは、これらのメッセージの伝送キューを見つけられない場合にこの状況を認識し、メッセージを未配布メッセージ (送達不能) キューに入れます。このようなメッセージの有無を調べて正しい宛先に転送できるようにする必要があります。または、宛先が確認できる、発信元にメッセージを戻します。

元のメッセージでレポート・メッセージが要求されていた場合は、このような状況になると例外報告書が生成されます。

## 解決の規則

宛先キューの識別名を変更するネーム・レゾリューション (すなわち、論理名から物理名への変更) は、発信元のキュー・マネージャーでのみ、1 回だけなされます。

それ以降は、各種の可能な別名は、メッセージ・フローの分離および組み合わせを行うときにのみ使用しなければなりません。

## 戻り経路

メッセージには、キューおよびキュー・マネージャーの名前の形式で戻りアドレスが入ることがあります。この戻りアドレスの形式は、分散キューイング環境とクラスター化環境の両方で使用できます。

このアドレスは、通常はメッセージを作成するアプリケーションによって指定されます。それは、あとでそのメッセージを取り扱うアプリケーション (ユーザー出口アプリケーションを含む) によって変更できます。

このアドレスの発信元とは関係なく、メッセージを扱うアプリケーションは、応答、状況、またはレポート・メッセージをメッセージ発信元アプリケーションに戻すためにこのアドレスを選択できます。

これらの応答メッセージの経路指定方法は、元のメッセージの経路指定方法と変わりません。別のキュー・マネージャーに向けて作成するメッセージ・フローには、対応する戻りフローが必要です。

## 物理名の矛盾

宛先の応答先キュー名は、解決されて元のキュー・マネージャーにある物理キュー名になっています。応答元のキュー・マネージャーで再び解決してはなりません。

そのようにすると、名前の矛盾の問題が発生することがあります。この矛盾は、ネットワーク全体で物理キュー名と論理キュー名に関する合意を得なければ回避できません。

## キュー名変換の管理

キュー・マネージャー別名定義またはリモート・キュー定義を作成するときには、その名前を伴う各メッセージごとにネーム・レゾリューションが行われます。この状態は管理されていなければなりません。

以下の説明は、隣接システムへのメッセージ・チャンネルを備えた個々のシステムを扱うアプリケーション設計者およびチャンネル計画者向けの説明です。ここでは、ローカル側から見たチャンネルの計画と制御について説明します。

キュー・マネージャー別名定義またはリモート・キュー定義を作成するときには、メッセージのソースとは無関係に、その名前を伴う各メッセージごとに解決が行われます。キュー・マネージャー・ネットワーク内の多数のキューが関連している可能性があるため、この状況を把握するには、以下の内容を示す表を保持する必要があります。

- 発信元キューと発信元キュー・マネージャーの名前、それに関連する解決されたキュー名、解決されたキュー・マネージャー名、解決された伝送キュー名、および解決方法
- 発信元キューの名前と以下のものとの関係
  - 解決された宛先キュー名
  - 解決された宛先キュー・マネージャー名
  - 伝送キュー
  - メッセージ・チャンネル名
  - 隣接システム名
  - 応答先キュー名

**注:** このコンテキストで使用されるソース (発信元) という用語は、アプリケーションが提供するキュー名またはキュー・マネージャー名、あるいはメッセージを書き込むためにキューをオープンする場合のチャンネル・プログラムを指します。

これらのそれぞれの表の例が [158 ページの表 16](#)、[159 ページの表 17](#)、および [159 ページの表 18](#) に示されています。

これらの表で使用されている名前は、このセクションの例から取られたものであり、この表は1つのノードのキュー名解決の実例として示されたものではありません。

キューがオープンされるときに、入力元キューが指定される	キューのオープン時に、指定されたソース・キュー・マネージャー	解決されたキュー名	解決されたキュー・マネージャー名	解決された伝送キュー名	解決タイプ
QA_norm	-	QA_norm	QMB	QMB	リモート・キュー
(任意)	QMB	-	-	QMB	(なし)
QA_norm	-	QA_norm	QMB	TX1	リモート・キュー
QB	QMC	QB	QMD	QMB	キュー・マネージャーの別名

キューがオープンされるときに、入力元キューが指定される	キューのオープン時に、指定されたソース・キュー・マネージャー	解決されたキュー名	解決されたキュー・マネージャー名	解決された伝送キュー名	解決タイプ
QA_norm	-	QA_norm	QMB	-	(なし)
QA_norm	QMB	QA_norm	QMB	-	(なし)
QA_norm	QMB_PRIORITY	QA_norm	QMB	-	キュー・マネージャーの別名
(任意)	QMC	(任意)	QMC	QMC	(なし)
(任意)	QMD_norm	(任意)	QMD_norm	TX1	キュー・マネージャーの別名
(任意)	QMD_PRIORITY	(任意)	QMD_PRIORITY	QMD_fast	キュー・マネージャーの別名
(任意)	QMC_small	(任意)	QMC_small	TX_small	キュー・マネージャーの別名
(任意)	QMC_large	(任意)	QMC_large	TX_external	キュー・マネージャーの別名
QB_small	QMC	QB_small	QMC	TX_small	リモート・キュー
QB_large	QMC	QB_large	QMC	TX_large	リモート・キュー
(任意)	QME	(任意)	QME	TX1	キュー・マネージャーの別名
QA	QMC	QA	QMC	TX1	リモート・キュー
QB	QMD	QB	QMD	TX1	リモート・キュー

アプリケーション設計		応答先別名定義	
ローカル <b>QMGR</b>	メッセージ用のキュー名	応答先キュー別名	定義後
QMA	QRR	QR	QMA_class1 の QRR

## チャンネル・メッセージ・シーケンス番号付け

チャンネルはシーケンス番号を使用して、メッセージが伝送キューから取り出された順序と同じ順序で送達されることを検査します。

チャンネルの開始時にチャンネル・シーケンス番号が検査され、不一致が発生した場合は、チャンネルの両側で永続同期データが失われたことを意味します。例えば、災害復旧 (DR) 構成の場合や、チャンネルの未確定時にバッチ処理の終了が中断された場合などです。

RESET CHANNEL コマンドを発行しても、メッセージが失われたり重複したりすることはありません。RESET は、何かが悪く見えないようにという警告を IBM MQ から出します。永続状態を失った未確定チャンネルは、RESOLVE CHANNEL コマンドを発行するまで、RESET 後も始動に失敗し続けます。このアクションにより、バッチが失われたり重複したりする可能性があります。

この情報は、`DISPLAY CHSTATUS` を使用して表示できます。最後にバッチで転送されたメッセージのシーケンス番号と `LUWID` という名前の ID は、持続ストレージに保管されます。これらの値をチャンネルの開始時に使用して、どのメッセージの転送が正常に完了したかをリンクの両側で確認することができます。

## メッセージの順次検索

アプリケーションが一連のメッセージを同じ宛先キューに書き込んだ場合に、以下の条件が満たされると、**1つの** アプリケーションでの一連の `MQGET` 操作でそれらのメッセージを順次検索することができます。

- すべての書き込み要求が同じアプリケーションから出された。
- すべての書き込み要求が同じ作業単位から行われたか、あるいはすべての書き込み要求が作業単位の外部で行われた。
- どのメッセージも同じ優先順位をもっている。
- どのメッセージも同じ持続性をもっている。
- リモート・キューイングの場合に、書き込み要求を行うアプリケーションから、キュー・マネージャーまたは相互通信を経由して、宛先キュー・マネージャーとターゲット・キューにつながる経路が1つしかないように構成されている。
- メッセージが送達不能キューに入れられない (例えば、キューが一時的に満杯になっているため)。
- メッセージを読み取るアプリケーションが、例えば特定の `MsgId` または `CorrelId` を指定するなどの方法で、検索順序を意図的に変更していない。
- 宛先キューからのメッセージを検索するために、1つのアプリケーションだけが読み取り操作を行う。複数のアプリケーションがある場合、送信側アプリケーションによって書き込まれた各メッセージ順序内のすべてのメッセージを読み取れるようにすべてのアプリケーションを設計しなければなりません。

**注:** 一連のメッセージが単一の作業単位によって書き込まれた場合であっても、他のタスクおよび作業単位からのメッセージがこれらのメッセージ順序の間に散在していることがあります。

上記の条件が満たされない場合に、ターゲット・キューでのメッセージの順序は重要なので、その順序を守りたいときは、メッセージに独自のメッセージ・シーケンス番号を組み込むようにアプリケーションをコード化してもかまいません。

## 高速、非持続メッセージの順次検索

高速チャンネル上の非持続メッセージは、同じチャンネル上の持続メッセージに置き換わるため、順序どおりに到着しません。受信 MCA は、非持続メッセージを宛先キューに即時に書き込み、それらを可視にします。持続性メッセージは次の同期点まで可視になりません。

## ループバック・テスト

ループバック・テストは、z/OS 以外のプラットフォームで、接続は実際に別のマシンにリンクしないで通信リンクをテストできる手法です。

接続は、別々のマシン上にあるかのように2つのキュー・マネージャー間にセットアップされますが、同じマシンの別のプロセスにループバックして接続をテストします。この技法は、アクティブなネットワークを必要とせず通信コードをテストできることを意味します。

それを行う方法は、使用している製品およびプロトコルによって異なります。

Windows システムでは、「ループバック」アダプターを使用できます。

詳細については、使用している製品の資料を参照してください。

## 経路トレースおよびアクティビティ記録

2つの方法で、メッセージが一連のキュー・マネージャーを通る経路を確認することができます。

IBM MQ 経路表示アプリケーション (制御コマンド `dspmqrte` によって使用可能) を使用するか、アクティビティ記録を使用できます。これらの両方のトピックは、[モニター・リファレンス](#)で説明されています。

## 分散キュー管理の概要

分散キュー管理 (DQM) は、キュー・マネージャー間の通信を定義および制御するのに使用します。

分散キュー管理は以下のことを行います。

- キュー・マネージャー間の通信チャンネルを定義および制御できるようにします。
- あるタイプのローカル・キュー (伝送キュー) からローカル・システム上の通信リンクに、また通信リンクから宛先キュー・マネージャーのロケーションにあるローカル・キューにメッセージを移動するための、メッセージ・チャンネル・サービスを提供します。
- パネル、コマンド、およびプログラムを使用して、チャンネルの作動のモニターと問題の診断を行うための機能を提供します。

チャンネル定義は、伝送キュー、通信リンク ID、およびチャンネル属性にチャンネル名を関連付けます。チャンネル定義の実現方法は、プラットフォームごとに異なります。メッセージの送受信はメッセージ・チャンネル・エージェント (MCA) と呼ばれるプログラムによって制御されます。このプログラムは、チャンネル定義を使用して、通信の開始および制御を行います。

MCA は DQM によって直接制御されます。この制御構造はプラットフォームによって決まりますが、一般的にはオペレーター・コマンドとパネルの他、リスナーとトリガー・モニターが含まれます。

メッセージ・チャンネルとは、あるキュー・マネージャーから別のキュー・マネージャーにメッセージを移動させるための、1方向のパイプです。したがって、1つのメッセージ・チャンネルは、MCA のペアによって表される2つのエンドポイントをもつことになります。各エンドポイントで、メッセージ・チャンネルの側が定義されています。例えば、一方の側で送信側を定義し、他方の側で受信側を定義します。

チャンネルの定義方法については、以下の章を参照してください。

- [ULW](#) 191 ページの『UNIX, Linux, and Windows 上のチャンネルのモニターおよび制御』
- [z/OS](#) 732 ページの『z/OS におけるチャンネルのモニターと制御』
- [IBM i](#) 214 ページの『IBM i 上のチャンネルのモニターおよび制御』

メッセージ・チャンネルの計画の例については、以下を参照してください。

- [ULW](#) UNIX, Linux, and Windows におけるメッセージ・チャンネルの計画例
- [IBM i](#) IBM i におけるメッセージ・チャンネルの計画例
- [z/OS](#) z/OS におけるメッセージ・チャンネルの計画例
- [z/OS](#) キュー共有グループを使用する z/OS におけるメッセージ・チャンネルの計画例

チャンネル出口については、[メッセージ・チャンネル用のチャンネル出口プログラム](#)を参照してください。

### 関連概念

[162 ページの『メッセージの送受信』](#)

以下の図は、メッセージが送信されるときエンティティー間の関係を詳しく表す、分散キュー管理モデルを示しています。また、制御のフローも示しています。

[169 ページの『チャンネル制御機能』](#)

チャンネル制御機能は、チャンネルを定義、モニター、および制御するための機能を備えています。

[182 ページの『メッセージを送達できない場合の処理』](#)

メッセージを送達できない場合、MCA はメッセージをいくつかの方法で処理できます。再試行するか、送信側へ返信するか、または送達不能キューへ書き込むことができます。

[187 ページの『初期設定および構成ファイル』](#)

チャンネル初期設定データの処理は、運用している IBM MQ プラットフォームによって異なります。

[189 ページの『データ変換』](#)

IBM MQ メッセージは、異なるキュー・マネージャーでのキュー間で送信したときには、データ変換を必要とする場合があります。

[189 ページの『独自のメッセージ・チャンネル・エージェントの作成』](#)

IBM MQ を使用すれば、独自のメッセージ・チャンネル・エージェント (MCA) プログラムの記述、または独立ソフトウェア・ベンダーからのインストールを行うことができます。

### 190 ページの『分散キュー管理に関するその他の考慮事項』

分散キュー管理用に IBM MQ を準備する際に考慮する、その他のトピックです。このトピックでは、未配布メッセージ・キュー、使用中のキュー、システム拡張およびユーザー出口プログラム、ならびにトラスレッド・アプリケーションとしてチャンネルおよびリスナーを実行する方法について説明します。

## 関連情報

### 構成情報の例

## メッセージの送受信

以下の図は、メッセージが送信される際のエンティティー間の関係を詳しく表す、分散キュー管理モデルを示しています。また、制御のフローも示しています。

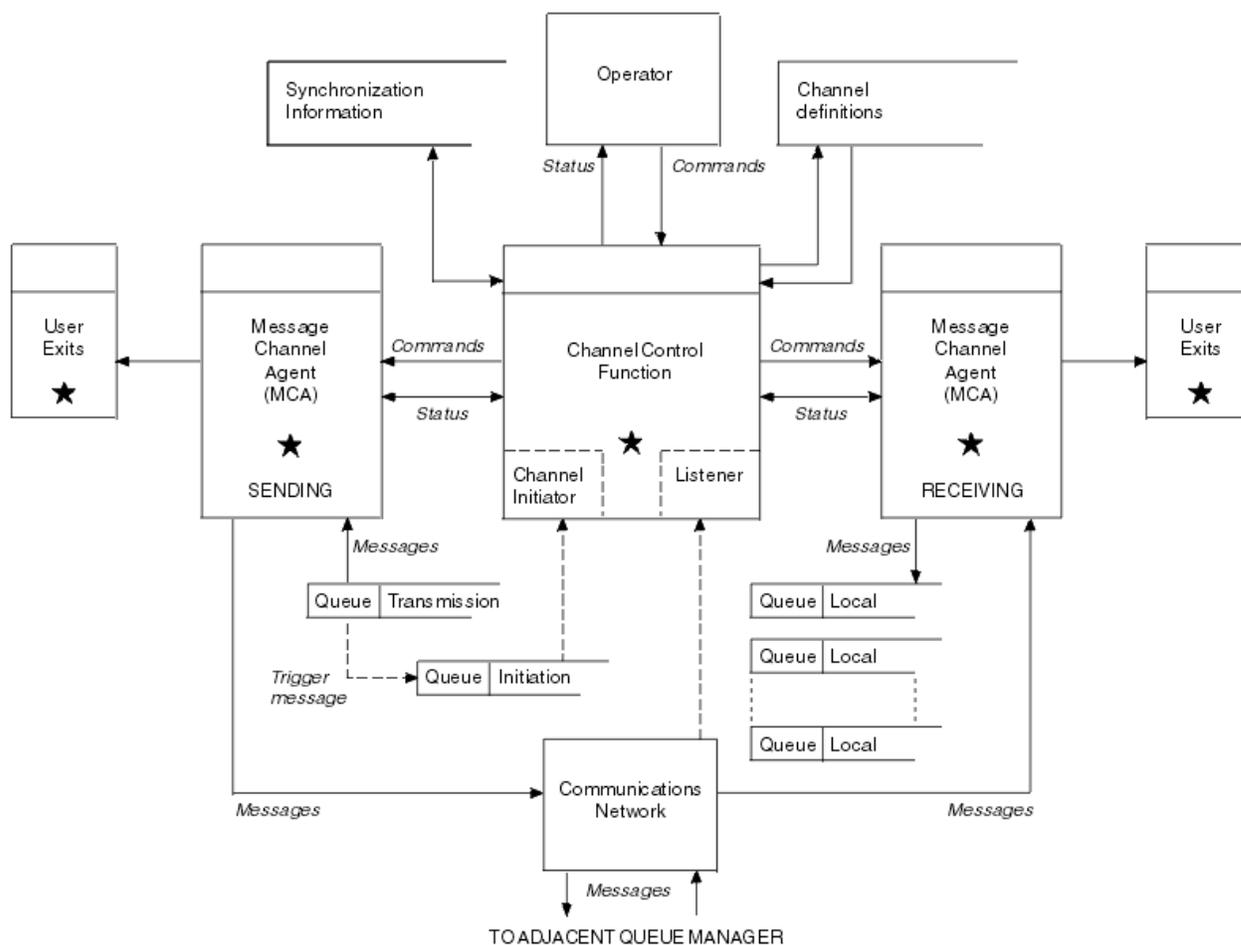


図 16. 分散キュー管理モデル

注:

1. プラットフォームに応じて、チャンネルごとに1つのMCAがあります。特定のキュー・マネージャーには、1つまたは複数のチャンネル制御機能が備わっていることがあります。
2. MCAの実装およびチャンネル制御機能は、プラットフォームに強く依存しています。これらは、プログラム、プロセス、またはスレッドのいずれかです。それぞれは、1つのエンティティーの場合もあり、いくつかの独立した部分またはリンクしている部分から構成される、複数のエンティティーの場合もあります。
3. 星印の付いたコンポーネントはすべてMQIを使用できます。

## チャンネル・パラメーター

MCA は、次のようないくつかの方法でパラメーターを受信します。

- コマンドで開始された場合は、データ域にチャンネル名が渡されます。すると、MCA はチャンネル定義を直接読み取り、その属性を取得します。
- 送信側の場合、および一部のサーバー・チャンネルの場合、MCA は、キュー・マネージャーのトリガー操作によって、自動的に開始されます。必要に応じてトリガー・プロセス定義からチャンネル名が取り出され、MCA に渡されます。それ以外の処理は、上記の場合と同じです。サーバー・チャンネルが起動するように設定されるのは、完全修飾されている、つまり CONNAME を指定して接続する場合のみです。
- 送信側、サーバー、要求側またはクライアント接続からリモートで開始された場合、チャンネル名はパートナーのメッセージ・チャンネル・エージェントからの初期データに入って渡されます。MCA はチャンネル定義を直接読み取り、その属性を取得します。

チャンネル定義内で定められていない以下の属性は折衝することもできます。

### 分割メッセージ

いずれかの側で分割メッセージがサポートされていない場合には、分割メッセージは送信されません。

### 変換機能

コード・ページ変換または数値エンコード変換が必要なときに、一方の側でこのような変換が行えない場合は、他方の側で行う必要があります。必要な場合にいずれの側でもこの変換がサポートされない場合には、チャンネルは開始できません。

### 配布リスト・サポート

一方の側が配布先リストをサポートしない場合、パートナーの MCA は伝送キューにフラグを設定して、複数の宛先に向けられたメッセージを代行受信することが分かるようにします。

## チャンネル状況およびシーケンス番号

メッセージ・チャンネル・エージェント・プログラムは、各チャンネルの現行シーケンス番号、論理作業単位番号、およびチャンネルの一般状況の記録を保持します。プラットフォームによっては、この状況情報を表示して、チャンネルの制御に役立てることができるものもあります。

## 別のキュー・マネージャーへのメッセージの送信方法

このセクションでは、キュー・マネージャー間でメッセージを送信する最も簡単な方法を、前提条件および必要な権限を含めて説明します。他の方法を使用してリモート・キュー・マネージャーにメッセージを送信することもできます。

あるキュー・マネージャーから別のキュー・マネージャーへメッセージを送信する前に、以下のステップを実行する必要があります。

1. 選択した通信プロトコルが使用可能であることをチェックします。
2. キュー・マネージャーを開始します。
3. チャンネル・イニシエーターを開始します。
4. リスナーを開始します。

また、必要なオブジェクトを作成するには、適切な IBM MQ セキュリティー権限をもっている必要があります。

あるキュー・マネージャーから別のキュー・マネージャーへメッセージを送信するには、次のようになります。

- 発信元キュー・マネージャーに次のオブジェクトを定義します。
  - 送信側チャンネル
  - リモート・キュー定義
  - 開始キュー (  z/OS の場合は必須、その他はオプション)
  - 伝送キュー

- 送達不能キュー
- ターゲット・キュー・マネージャーに次のオブジェクトを定義します。
  - 受信側チャンネル
  - ターゲット・キュー
  - 送達不能キュー

ご使用の IBM MQ プラットフォームに応じてそれぞれ異なる方法を使用してこれらのオブジェクトを定義することができます。

- すべてのプラットフォーム上で、MQSC コマンドに説明されている IBM MQ スクリプト・コマンド (MQSC)、[管理タスクの自動化](#)に説明されているプログラマブル・コマンド・フォーマット (PCF) コマンド、または IBM MQ エクスプローラーを使用できます。
-  z/OS では、[IBM MQ for z/OS の管理](#)で説明されている操作パネルおよび制御パネルも使用できます。
-  IBM i では、パネル・インターフェースも使用できます。

別のキュー・マネージャーにメッセージを送信するためのコンポーネントの作成についての詳細は、以下のサブトピックを参照してください。

### 関連概念

#### [140 ページの『IBM MQ 分散キューイング技法』](#)

このセクションのサブトピックでは、チャンネルを計画する際に役立つ技法について説明します。これらのサブトピックでは、キュー・マネージャー同士を接続して、アプリケーション間のメッセージのフローを管理する方法を計画するのに役立つ技法について説明します。

#### [161 ページの『分散キュー管理の概要』](#)

分散キュー管理 (DQM) は、キュー・マネージャー間の通信を定義および制御するのに使用します。

#### [184 ページの『チャンネルのトリガー操作』](#)

IBM MQ には、キューで特定の条件が満たされると自動的にアプリケーションを開始する機能が用意されています。この機能は、トリガー操作と呼ばれています。

#### [182 ページの『メッセージの安全性』](#)

IBM MQ の通常のリカバリー機能の他に、分散キュー管理は、メッセージ・チャンネルの両側で調整された同期点プロシージャーを使用して、メッセージが適切に送達されることを確認します。このプロシージャーはエラーを検出すると、問題を調査できるようにチャンネルをクローズし、チャンネルが再開されるまでメッセージを伝送キューに安全に保管します。

#### [728 ページの『他のキュー・マネージャーとの通信のセットアップ』](#)

このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

### 関連タスク

#### [5 ページの『マルチプラットフォームでのキュー・マネージャーの作成と管理』](#)

メッセージおよびキューを使用するには、事前に少なくとも 1 つのキュー・マネージャーとそれに関連するオブジェクトを作成し、開始しておく必要があります。キュー・マネージャーは、関連したリソース、特にキュー・マネージャー自体が所有しているキューを管理します。キュー・マネージャーは、メッセージ・キューイング・インターフェース (MQI) 呼び出しおよびコマンドで IBM MQ オブジェクトを作成、修正、表示、および削除できるように、アプリケーションに対してキューイング・サービスを提供します。

#### [191 ページの『UNIX, Linux, and Windows 上のチャンネルのモニターおよび制御』](#)

DQM では、リモート・キュー・マネージャーに対するチャンネルの作成、モニターおよび制御が必要です。コマンド、プログラム、IBM MQ Explorer、チャンネル定義ファイル、および同期情報のストレージ域を使用して、チャンネルを制御できます。

#### [214 ページの『IBM i 上のチャンネルのモニターおよび制御』](#)

リモート・キュー・マネージャーにつながるチャンネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM プログラムが備わっています。

#### [16 ページの『サーバーとクライアント間の接続の構成』](#)

IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャンネルを定義します。

### 236 ページの『キュー・マネージャー・クラスターの構成』

クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

## チャンネルの定義

あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信するには、2つのチャンネルを定義する必要があります。1つのチャンネルはソース・キュー・マネージャーに、もう1つのチャンネルはターゲット・キュー・マネージャーに定義する必要があります。

### 発信元キュー・マネージャーの場合

チャンネル・タイプがSENDERのチャンネルを定義します。以下を指定する必要があります。

- 使用される伝送キューの名前 (XMITQ 属性)。
- パートナー・システムの接続名 (CONNNAME 属性)。
- 使用している通信プロトコル名 (TRPTYPE 属性)。IBM MQ for z/OS では、プロトコルはTCPまたはLU6.2でなければなりません。他のプラットフォームでは、これを指定する必要はありません。そのままにしておき、デフォルトのチャンネル定義から値を選ぶことができます。

すべてのチャンネル属性の詳細は、[チャンネル属性](#)に記載されています。

### ターゲット・キュー・マネージャーの場合

チャンネル・タイプRECEIVERで、送信側チャンネルと同じ名前のチャンネルを定義します。

使用している通信プロトコル名を指定します (TRPTYPE 属性)。IBM MQ for z/OS では、プロトコルはTCPまたはLU6.2でなければなりません。他のプラットフォームでは、これを指定する必要はありません。そのままにしておき、デフォルトのチャンネル定義から値を選ぶことができます。

受信側チャンネルの定義は汎用にすることができます。これは、同じ受信側と通信する複数のキュー・マネージャーがある場合に、送信側チャンネルですべて受信側に同じ名前を指定でき、1つの受信側の定義がそれらすべてに適用されることを意味します。

チャンネルを定義するとき、PING CHANNEL コマンドを使用してテストできます。このコマンドは、送信側チャンネルから受信側チャンネルに特別なメッセージを送信し、それが返されることをチェックします。

注: TRPTYPE パラメーターの値は、応答メッセージ・チャンネル・エージェントで無視されます。例えば、送信側チャンネル定義のTRPTYPEがTCPであっても、パートナーの受信側チャンネル定義のTRPTYPEは正しくLU62で始まります。

## キューの定義

あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信するには、最大6個のキューを定義する必要があります。そのうち4個以下のキューはソース・キュー・マネージャーに、2個以下のキューはターゲット・キュー・マネージャーに定義する必要があります。

### 発信元キュー・マネージャーの場合

- リモート・キュー定義

この定義では、以下を指定します。

#### リモート・キュー・マネージャー名

ターゲット・キュー・マネージャーの名前です。

#### リモート・キュー名

ターゲット・キュー・マネージャーのターゲット・キューの名前です。

#### Transmission queue name

伝送キューの名前です。この伝送キューの名前を指定する必要はありません。指定しない場合、ターゲット・キュー・マネージャーと同じ名前の伝送キューが使用されます。それが存在しない場合には、デフォルトの伝送キューが使用されます。伝送キューをターゲット・キュー・マネージャーと同じ名前にして、デフォルトでキューが検出されるようにすることをお勧めします。

- 開始キュー定義

**z/OS** これは必須です。SYSTEM.CHANNEL.INITQ という開始キューを使用しなければなりません。

**Multi** これはオプションです。開始キュー SYSTEM.CHANNEL.INITQ の名前を指定することを考慮してください。

- 伝送キュー定義

XMITQ に設定された USAGE 属性をもつローカル・キューです。**IBM i** IBM MQ for IBM i 固有のインターフェースを使用する場合、USAGE 属性は \*TMQ となります。

- 送達不能キュー定義

未配布メッセージを書き込むことのできる送達不能キューの定義を選択してください。

#### ターゲット・キュー・マネージャーの場合

- ローカル・キュー定義

ターゲット・キュー。このキューの名前は、発信元キュー・マネージャーのリモート・キュー定義のリモート・キュー名フィールドに指定された名前と同じでなければなりません。

- 送達不能キュー定義

未配布メッセージを書き込むことのできる送達不能キューの定義を選択してください。

#### 関連概念

##### 166 ページの『伝送キューの作成』

チャンネル (要求側チャンネル以外) を開始する前に、このセクションの説明に従って伝送キューを定義しておく必要があります。チャンネル定義において、伝送キューの名前が指定されている必要があります。

##### 166 ページの『IBM i での伝送キューの作成』

「MQM キューの作成」パネルを使用して、IBM i プラットフォームで伝送キューを作成できます。

#### 伝送キューの作成

チャンネル (要求側チャンネル以外) を開始する前に、このセクションの説明に従って伝送キューを定義しておく必要があります。チャンネル定義において、伝送キューの名前が指定されている必要があります。

それぞれの送信側メッセージ・チャンネルについて、USAGE 属性を XMITQ に設定して、ローカル・キューを定義してください。リモート・キュー定義で、特定の伝送キューを使用したい場合は、以下で示すように、リモートのキューを作成してください。

伝送キューを作成するには、次の例に示すように、IBM MQ コマンド (MQSC) を使用してください。

#### 伝送キューの作成例

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') USAGE(XMITQ)
```

#### リモート・キューの作成例

```
DEFINE QREMOTE(PAYROLL) DESCR('Remote queue for QM2') +  
XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

伝送キューには、上記の例のように、リモート・システムのキュー・マネージャーの名前を付けることを考慮してください。

#### IBM i での伝送キューの作成

「MQM キューの作成」パネルを使用して、IBM i プラットフォームで伝送キューを作成できます。

それぞれの送信側メッセージ・チャンネルについて、「Usage」フィールド属性を \*TMQ に設定して、ローカル・キューを定義する必要があります。

リモート・キュー定義を利用する場合、同じコマンドを使用して、タイプが \*RMT で、使用法が \*NORMAL のキューを作成してください。

伝送キューを作成するには、コマンド行から CRTMQMQ コマンドを使用して、最初のキュー作成パネルを表示します。[167 ページの図 17](#) を参照してください。

```
Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
キュー名 . . . . .
キュー・タイプ . . . . . ____ *ALS、 *LCL、 *MDL、 *RMT
メッセージ・キュー・マネージャー名 . . . * DFT_____
-----

Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
+
```

図 17. キューの作成 (1)

キューの名前を入力し、作成したいキューのタイプとしてローカル (\*LCL)、リモート (\*RMT)、または別名 (\*ALS) のいずれかを指定してください。伝送キューの場合、このパネルにローカル (\*LCL) を指定して Enter キーを押してください。

「Create MQM Queue (MQM キューの作成)」パネルの 2 ページ目が表示されます。[167 ページの図 18](#) を参照してください。

```
Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
キュー名 . . . . . > HURS.2.HURS.PRIORIT
キュー・タイプ . . . . . > *LCL *ALS、 *LCL、 *MDL、 *RMT
メッセージ・キュー・マネージャー名 . . . * DFT
. . . . . *いいえ * いいえ、 *はい
テキスト 'description' . . . . .
書き込み可能 . . . . . *はい *SYSDFTQ、 *NO、 *YES
デフォルトのメッセージ優先順位 . . . . . 0 から 9、 *SYSDFTQ
デフォルトのメッセージ持続性。 *NO *SYSDFTQ、 *NO、 *YES
プロセス名 . . . . .
トリガーが有効化されました。 . . . *NO *SYSDFTQ、 *NO、 *YES
使用可能 . . . . . *はい *SYSDFTQ、 *NO、 *YES
共用が使用可能になっています。 . . . . *はい *SYSDFTQ、 *NO、 *YES

More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

図 18. キューの作成 (2)

表示されたデフォルト値を変更したい場合は、変更してください。Page Down (次ページ) キーを押して次の画面を表示させます。[168 ページの図 19](#) を参照してください。

```
Create MQM Queue (CRTMQMQ)
```

```
Type choices, press Enter.
```

```
デフォルトの共用オプション . . . . *はい *SYSDFTQ、 *NO、 *YES
Message delivery sequence . . . *PTY *SYSDFTQ、 *PTY、 *FIFO
バックアウト・カウンタをハードンします。 . . . *NO *SYSDFTQ、 *NO、 *YES
トリガー・タイプ . . . . *最初の *SYSDFTQ、 *最初、 *ALL...
Trigger depth . . . . 1 1-999999999、 *SYSDFTQ
トリガー・メッセージ優先順位 . . . . 0 から 9、 *SYSDFTQ
トリガー・データ . . . . ' '
保存間隔 . . . . 999999999 0-999999999、 *SYSDFTQ
最大キュー項目数 . . . . 5000 ~ 24000、 *SYSDFTQ
最大メッセージ長 . . . . 4194304 0-4194304、 *SYSDFTQ
バックアウトしきい値 . . . . 0 0 から 99999999999、 *SYSDFTQ
バックアウト・リキュー・キュー . . . . ' '
開始キュー . . . . ' '

```

```
More...
```

```
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

### 図 19. キューの作成 (3)

このパネルの「Usage (使用法)」フィールドに伝送キューを意味する \*TMQ を入力し、その他のフィールドに表示されているデフォルト値を必要に応じて変更してください。

```
Create MQM Queue (CRTMQMQ)
```

```
Type choices, press Enter.
```

```
使用法 . . . . *TMQ *SYSDFTQ、 *通常、 *TMQ
Queue depth high threshold . . . 80 0-100、 *SYSDFTQ
Queue depth low threshold . . . 20 0-100、 *SYSDFTQ
キュー・フル・イベントが使用可能。 *はい *SYSDFTQ、 *NO、 *YES
キューの高イベントが使用可能になりました。 *はい *SYSDFTQ、 *NO、 *YES
キュー・ロー・イベントが使用可能になりました。 *はい *SYSDFTQ、 *NO、 *YES
サービス間隔 . . . . 999999999 0-999999999、 *SYSDFTQ
Service interval events . . . . *NONE *SYSDFTQ、 *HIGH、 *OK、 *NONE
配布リストのサポート。 . *NO *SYSDFTQ、 *NO、 *YES
クラスター名 . . . . *SYSDFTQ
「クラスター名リスト」 . . . *SYSDFTQ
Default Binding . . . . *SYSDFTQ *SYSDFTQ、 *OPEN、 *NOTFIXED

```

```
Bottom
```

```
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

### 図 20. キューの作成 (4)

フィールドに適切なデータが入っていることを確認してから、Enter キーを押してキューを作成してください。

## チャネルの開始

発信元キュー・マネージャーで定義されるリモート・キューにメッセージを書き込むとき、これらのメッセージはチャネルが開始されるまで伝送キューに保管されます。チャネルが開始されると、メッセージはリモート・キュー・マネージャーのターゲット・キューに送達されます。

送信側キュー・マネージャーでチャネルを開始するには、START CHANNEL コマンドを使用します。送信側チャネルを開始すると、受信側チャネルは (リスナーによって) 自動的に開始され、メッセージがターゲ

ット・キューに送信されます。メッセージを転送するには、メッセージ・チャンネルの両側が共に実行されていなければなりません。

チャンネルの両側には別のキュー・マネージャーがあるため、チャンネルは別の属性で定義されている場合があります。この差を解消するには、チャンネルの開始時にチャンネルの両側で初期データ折衝が行われます。一般に、チャンネルの両側はより少ないリソースを必要とする属性で作動します。これにより、より大規模なシステムが、メッセージ・チャンネルの片側でより小規模なシステムのより少ないリソースに適応できます。

送信 MCA は、チャンネル間の送信前に大きなメッセージを分割します。分割されたメッセージは、リモート・キュー・マネージャーで再度組み立てられます。これは、ユーザーからは分かりません。

MCA は、複数のスレッドを使用してメッセージを転送できます。パイプライン化と呼ばれるこのプロセスにより、MCA は待ち状態になることが少なくなり、メッセージをより効率的に転送できます。パイプライン化により、チャンネルのパフォーマンスが向上します。

## チャンネル制御機能

チャンネル制御機能は、チャンネルを定義、モニター、および制御するための機能を備えています。

コマンドは、パネル、プログラム、またはコマンド行から出されて、チャンネル制御機能に渡されます。パネル・インターフェースは、チャンネル状況およびチャンネル定義データの表示も行います。プログラマブル・コマンド・フォーマットまたはそれらの IBM MQ コマンド (MQSC) を使用して、[191 ページの『UNIX, Linux, and Windows 上のチャンネルのモニターおよび制御』](#)に説明されているコマンドを制御することができます。

これらのコマンドは、以下のグループに分類されます。

- チャンネル管理
- チャンネル制御
- チャンネル状況モニター

チャンネル管理コマンドは、チャンネルの定義について扱います。これらを使用して以下のことができます。

- チャンネル定義の作成
- チャンネル定義のコピー
- チャンネル定義の変更
- チャンネル定義の削除

チャンネル制御コマンドは、チャンネルの動作を管理します。これらを使用して以下のことができます。

- チャンネルの開始
- チャンネルの停止
- パートナーとの再同期 (一部のインストール先で)
- メッセージ・シーケンス番号のリセット
- メッセージの未確定バッチの解決
- ping。チャンネルを介するテスト通信の送信

チャンネル・モニターでは、以下のようなチャンネルの状態が表示されます。

- 現行のチャンネル設定
- チャンネルがアクティブか非アクティブか
- チャンネルが同期化状態で終了したか

チャンネルの定義、制御、およびモニターについての詳細は、以下のサブトピックを参照してください。

## チャンネルの準備

メッセージ・チャンネルまたは MQI チャンネルを開始しようとする前に、そのチャンネルを準備しておく必要があります。ローカルおよびリモートのチャンネル定義の全属性が正しく、かつ互換性のある設定になっていなければなりません。

チャンネル属性は、チャンネルの定義と属性を説明しています。

明示的にチャンネル定義を設定しても、チャンネル開始時に行われるチャンネル折衝によって定義の値が部分的に指定変更される場合があります。この動作は正常であり、ユーザーには意識されない仕方で行われます。これは、そのままでは矛盾する定義同士を一緒に使用できるようにするために調整されている動作です。

## 受信側チャンネルおよびサーバー接続チャンネルの自動定義

z/OS を除くすべてのプラットフォーム上の IBM MQ では、適切なチャンネル定義がない場合、自動定義が使用可能になっている受信側チャンネルまたはサーバー接続チャンネルの定義は自動的に作成されます。定義は、以下のものを使用して作成されます。

1. 適切なモデルのチャンネル定義、SYSTEM.AUTO.RECEIVER、または SYSTEM.AUTO.SVRCONN。自動定義用のモデル・チャンネル定義は、システム・デフォルトの SYSTEM.DEF.RECEIVER、および SYSTEM.DEF.SVRCONN と同じです。ただし、49 文字の空白が続く「Auto-defined by (自動定義に使用するモデル・チャンネル定義)」の説明フィールドを除きます。システム管理者は、システムに備えられたモデル・チャンネル定義の任意の部分を選択して変更できます。
2. パートナー・システムからの情報。チャンネル名およびシーケンス番号の折り返し値には、パートナーの値が使用されます。
3. 自動定義で作成される値を変更するために使用できる、チャンネル出口プログラム。[チャンネル自動定義出口プログラム](#)を参照してください。

その後、記述がチェックされ、自動定義出口プログラムによって定義が変更されたのかまたはモデル定義が変更されたためなのかを判別します。最初の 44 文字がまだ 29 文字の空白が続く「Auto-defined by (自動定義に使用するモデル・チャンネル定義)」の場合、キュー・マネージャー名が追加されます。最終の 20 文字がまだすべて空白の場合は、ローカルの時刻と日付が追加されます。

定義が作成され、保管されると、その定義が以前から引き続き存在していた場合と同じ条件でチャンネルの開始が行われます。バッチ・サイズ、伝送サイズ、メッセージ・サイズがパートナーと折衝されます。

## その他のオブジェクトの定義

メッセージ・チャンネルを開始するには、チャンネルの両端をそれぞれのキュー・マネージャーで定義しておく(または自動定義を使用可能にしておく)必要があります。チャンネルがサービスを行う伝送キューを、送信側でキュー・マネージャーに対して定義しておく必要があります。通信リンクを定義し、使用可能にしておかなければなりません。場合によっては、その他の IBM MQ オブジェクト(リモート・キュー定義、キュー・マネージャー別名定義、応答先キュー別名定義など)を準備して、140 ページの『分散キューイングの構成』に記載しているシナリオを実現できるようにする必要があります。

MQI チャンネルの定義については、30 ページの『MQI チャンネルの定義』を参照してください。

## 伝送キュー当たりの複数メッセージ・チャンネル

1 つの伝送キューに複数のチャンネルを定義することが可能ですが、これらのチャンネルのうち、一度にアクティブにしておけるのは 1 つだけです。このオプションは、通信量の平衡化およびリンク障害の訂正アクション用に、キュー・マネージャー間に代替経路を用意する方法とを考えてください。伝送キューを使用した直前のチャンネルが送信側でメッセージのバッチを未確定なままで終了した場合、別のチャンネルが伝送キューを使用することはできません。詳細については、180 ページの『未確定チャンネル』を参照してください。

## チャンネルの開始

チャンネルは、4 つの方法のいずれかでメッセージの送信することができます。次の方法にすることができます:

- オペレーターによって (受信側チャンネル、クラスター受信側チャンネル、サーバー接続チャンネル以外)。
- 伝送キューから起動。この方式は、送信側チャンネルおよび完全修飾サーバー・チャンネル (CONNNAME を指定するチャンネル) の場合にのみ使用できます。チャンネルをトリガーするために、必要なオブジェクトを準備する必要があります。
- アプリケーション・プログラムからの開始 (受信側チャンネル、クラスター受信側チャンネル、またはサーバー接続チャンネル以外)。
- 送信側、クラスター送信側、要求側、サーバー、またはクライアント接続チャンネルによるネットワークからのリモート開始。受信側チャンネル、クラスター受信側チャンネル、および場合によってはサーバー・チャンネルと要求側チャンネルによる伝送は、この方法で開始できます。サーバー接続チャンネルもこの方法で開始できます。チャンネル自体はすでに開始済み (つまり使用可能) になっていなければなりません。

注: チャンネルが「開始済み」であっても、メッセージを送送しているとは限りません。前述の 4 つのイベントのいずれかが発生して、伝送の開始が「使用可能になっている」ことも考えられます。チャンネルは、START および STOP オペレーター・コマンドを使用して使用可能または使用不可にします。

## チャンネルの状態

チャンネルは、いつでも多くの状態のうちの一つであることができます。いくつかの状態には、副状態もあります。チャンネルは、指定された状態から他の状態に移動することができます。

171 ページの図 21 は、考えられるすべてのチャンネル状態と各チャンネル状態に対応する副状態の階層を示しています。

172 ページの図 22 は、チャンネル状態間のリンクを示しています。これらのリンクは、すべてのタイプのメッセージ・チャンネルおよびサーバー接続チャンネルに適用されます。

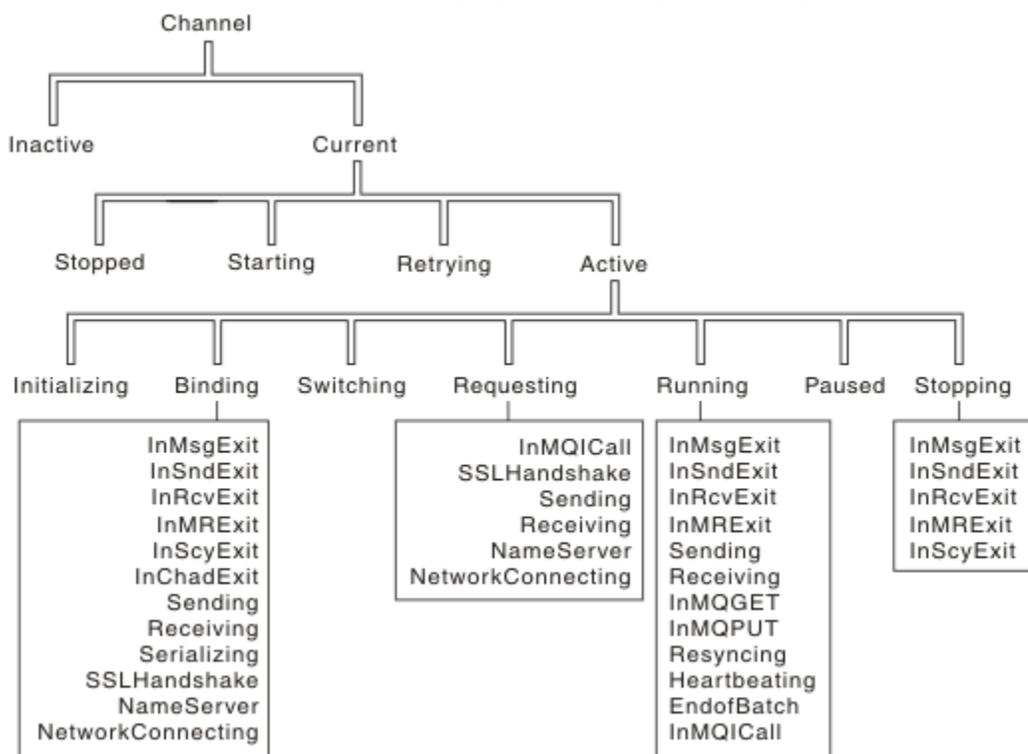


図 21. チャンネル状態および副状態

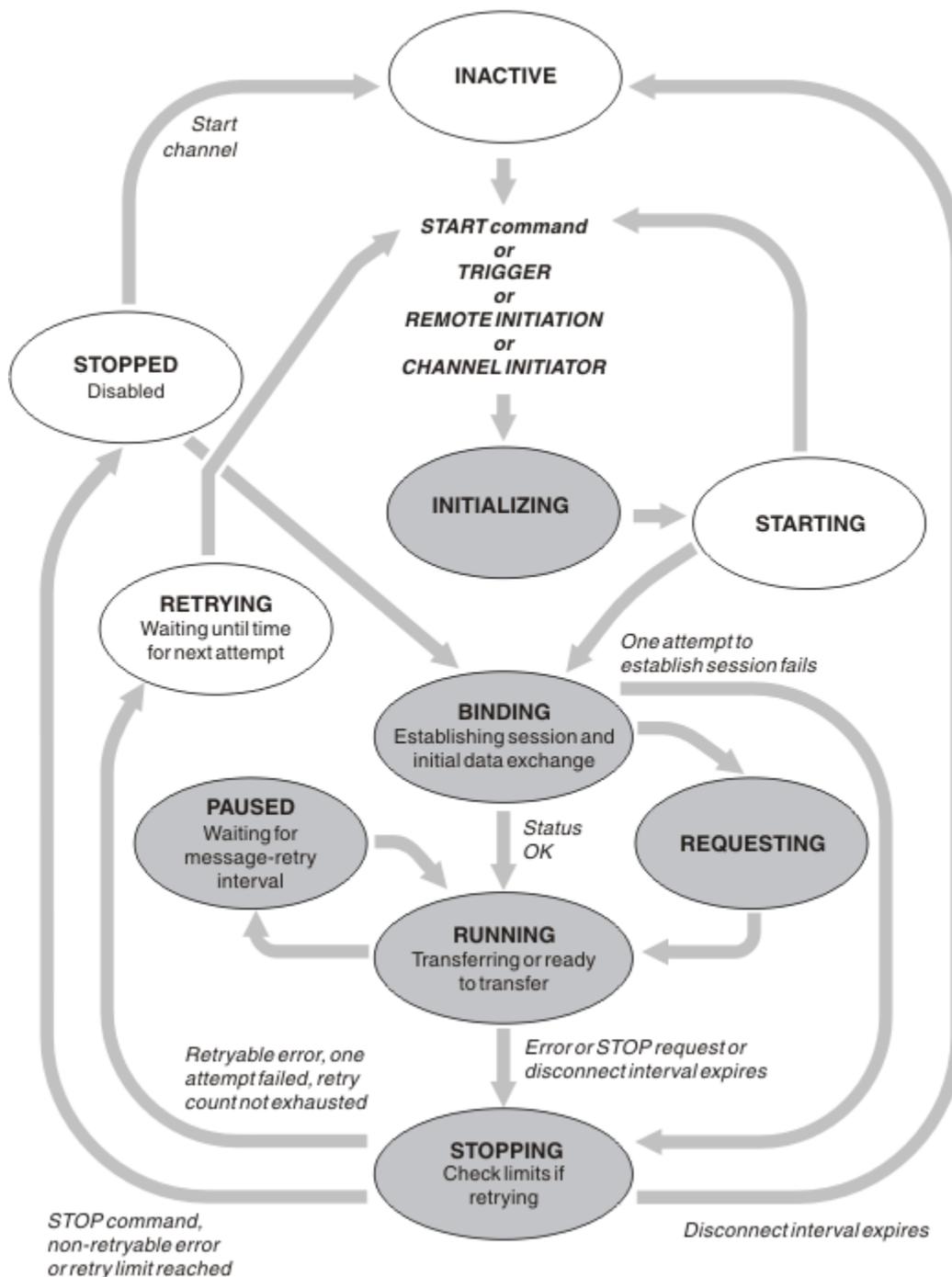


図 22. チャネル状態間のフロー

## 現行およびアクティブ

チャネルは、非アクティブ以外の状態であれば、現行です。現行チャネルは、RETRYING、STOPPED、または STARTING の状態でない限り、アクティブです。チャネルはアクティブであるとき、リソースを消費しており、プロセスまたはスレッドが実行されています。172 ページの図 22 では、アクティブ・チャネルの 7 つの可能な状態 (INITIALIZING、BINDING、SWITCHING、REQUESTING、RUNNING、PAUSED、または STOPPING) が強調表示されています。

アクティブ・チャネルは、そのチャネルが今どの処理を実行しているかを詳細に表す副状態も示すことができます。各状態の副状態は、171 ページの図 21 に示されています。

現行およびアクティブ

チャンネルは、非アクティブ以外の状態であれば、"現行"です。現行チャンネルは、RETRYING、STOPPED、または STARTING 状態でない限り、"アクティブ"です。

チャンネルが「アクティブ」の状態であれば、そのチャンネルが今どの処理を実行しているかを詳細に表す副状態も示されます。

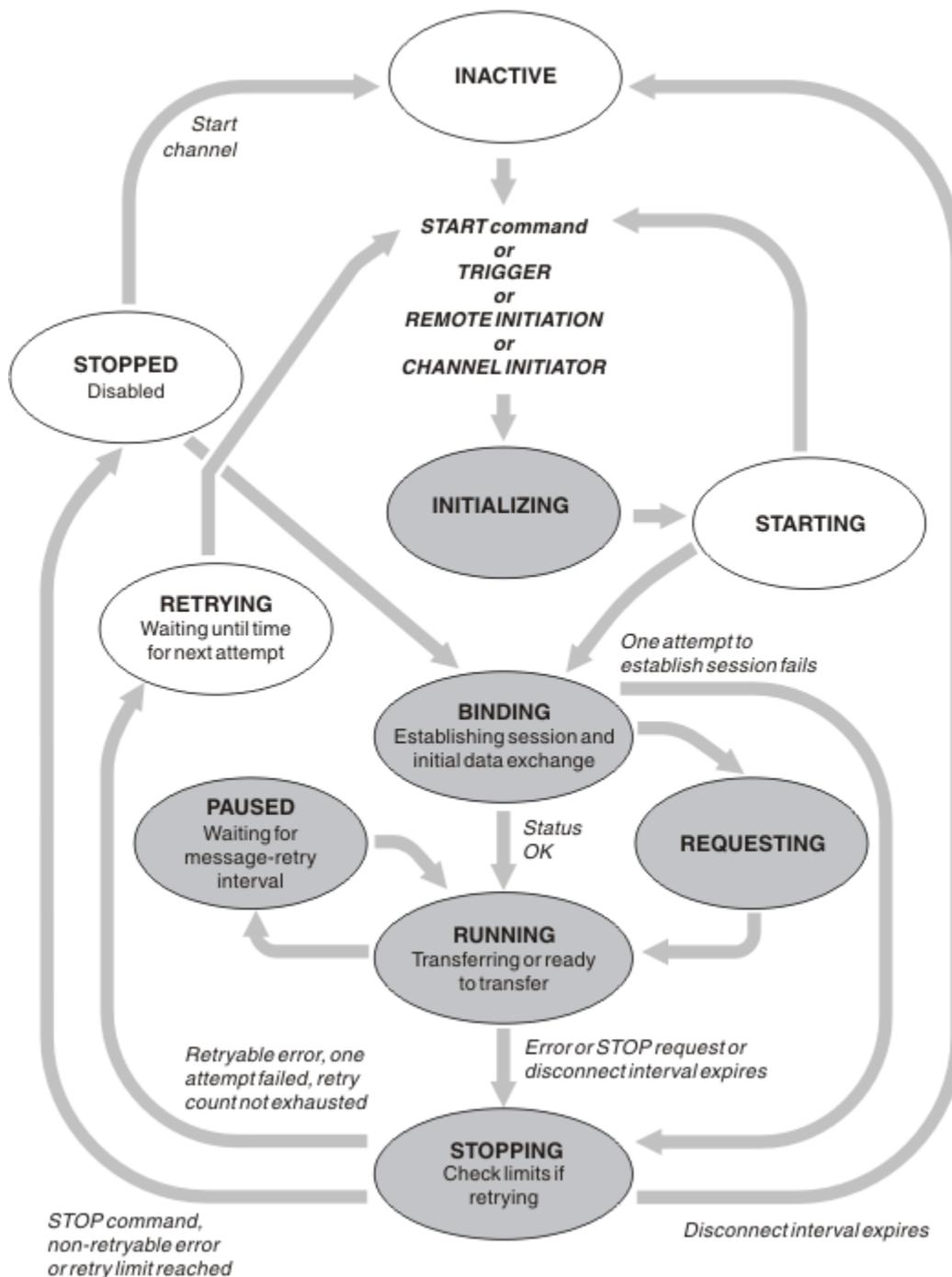


図 23. チャンネル状態間のフロー

注:

1. チャンネルは、173 ページの図 23 で強調表示されている 6 種類の状態 (初期化、バインディング、要求、実行、休止、または停止) のいずれかのときは、リソースを消費しており、プロセスまたはスレッドが実行されています。すなわち、チャンネルはアクティブです。

2. チャンネル停止状態のとき、次の状態が何か分からないため、セッションがアクティブである場合があります。

## 現行チャンネルの最大数の指定

一度に現行チャンネルにしてもよいチャンネルの最大数を指定することができます。この数は、チャンネル状況テーブルに項目があるチャンネルの数で、これには再試行中のチャンネルと停止中のチャンネルが含まれます。ご使用のプラットフォームに合わせて指定してください。

- ▶ **z/OS** ALTER QMGR MAXCHL コマンドを使用します。
- ▶ **IBM i** キュー・マネージャー初期設定ファイルを編集します。
- ▶ **Linux** ▶ **UNIX** キュー・マネージャー構成ファイルを編集します。
- IBM MQ Explorer を使用します。

初期設定ファイルまたは構成ファイルを使用して設定する値について詳しくは、[分散キューイング用の構成ファイル・スタンプ](#)を参照してください。チャンネルの最大数を指定する方法の詳細については、以下のトピックを参照してください。

- ▶ **ULW** [IBM MQ の管理](#)。
- ▶ **IBM i** [IBM MQ for IBM i の管理](#)。
- ▶ **z/OS** [IBM MQ for z/OS の管理](#)。

注：

1. この数にサーバー接続チャンネルが含まれています。
2. アクティブのチャンネルは現行チャンネルであることが必要です。開始したチャンネルが、現行チャンネルにならない場合、開始は失敗します。

## アクティブのチャンネルの最大数の指定

アクティブ・チャンネルの最大数を指定して、多くの開始チャンネルによってシステムの負荷が過大になるのを防ぐことができます。この方法を使用する場合は、切断時間間隔属性を低い値に設定して、他のチャンネルが終了したらすぐに待機中のチャンネルを開始できるようにします。

再試行するチャンネルがパートナーとの接続を確立しようとするたびに、アクティブのチャンネルになる必要があります。試行が失敗した場合、次の試行が再開されるまで、非アクティブのチャンネルは現行のままになります。チャンネルを再試行する回数および頻度は、再試行カウント属性および再試行間隔チャンネル属性によって決まります。これらの属性には両方とも長い値と短い値があります。詳細については、[チャンネル属性](#)を参照してください。

チャンネルがアクティブのチャンネルになる必要があるが (START コマンドが出されたか、または起動されたか、または別の再試行が試みられたため)、アクティブのチャンネル数がすでに最大数に達しているためにアクティブにできない場合、アクティブのスロットの1つが、アクティブになるのをやめた別のチャンネル・インスタンスによって解放されるまで、そのチャンネルは待機します。ただし、リモートで開始されたためにチャンネルが始動し、その時点で使用できるアクティブのスロットがない場合は、リモート開始は拒否されます。

要求側チャンネル以外のチャンネルがアクティブになる直前には必ず、開始状態になります。アクティブなスロットをすぐに利用できる場合でもこの状態が発生しますが、開始状態になるのは短時間のみです。ただし、チャンネルがアクティブのスロットを待っている必要がある場合は、その待機中に開始状態になります。

要求側チャンネルは開始状態にはなりません。アクティブにあるチャンネルの数がすでに上限に達しているために要求側チャンネルを開始できない場合、そのチャンネルは異常終了します。

要求側チャンネル以外のチャンネルがアクティブのスロットを確保できないため、待機している場合はいつでも、メッセージがログ ▶ **z/OS** または z/OS コンソールに書き込まれ、イベントが生成されます。その後、スロットが解放され、チャンネルがスロットを獲得できる場合、別のメッセージおよびイベントが生

成されます。チャンネルがスロットをじかに獲得できる場合は、これらのイベントおよびメッセージはどちらも生成されません。

チャンネルがアクティブになるのを待っている間に STOP CHANNEL コマンドが出された場合、そのチャンネルは停止状態になります。チャンネル停止イベントが発生します。

アクティブ・チャンネルの最大数にサーバー接続チャンネルが含まれています。

アクティブ・チャンネルの最大数を指定する方法の詳細については、以下のトピックを参照してください。

-  [ULW](#) IBM MQ の管理。
-  [IBM i](#) IBM MQ for IBM i の管理。
-  [z/OS](#) IBM MQ for z/OS の管理。

#### チャンネル・エラー

チャンネルでエラーが発生すると、それ以降の伝送が停止することになります。チャンネルが送信側またはサーバーの場合、問題が自然に解決する場合もあるので、チャンネルは再試行状態に入ります。再試行状態にできない場合、チャンネルは停止状態になります。

送信側チャンネルの場合は、関連する伝送キューは GET(DISABLED) に設定され、トリガー操作はオフにされます。(STATUS(STOPPED) を指定した STOP コマンドによってもチャンネルは停止状態になります。切断時間間隔の満了時に、または STATUS(INACTIVE) を指定した STOP コマンドによってのみ、チャンネルは正常に終了し、非アクティブになります。) STOPPED 状態のチャンネルは、再始動する前にオペレーターの介入が必要になります (180 ページの『[停止したチャンネルの再始動](#)』を参照してください)。

**注:**  [IBM i](#) IBM i、UNIX、Linux、and Windows システムの場合、再試行を行うにはチャンネル・イニシエーターを実行しなければなりません。チャンネル・イニシエーターが利用不能の場合は、チャンネルは非アクティブになるので手動で再始動する必要があります。スクリプトを使用してチャンネルを開始する場合は、スクリプトを実行する前にチャンネル・イニシエーターを実行するようにしてください。

[長期再試行カウント \(LONGRTY\)](#) は、再試行がどのように機能するかを説明しています。エラーが解消されると、自動的にチャンネルが再始動されて、伝送キューが再び使用可能になります。エラーが解消されずに再試行限界に達すると、チャンネルは停止状態になります。停止状態のチャンネルは、オペレーターが手動で再始動する必要があります。エラーが持続している場合は、もう一度再始動を行うことはありません。正常に開始されると、再び伝送キューが使用可能になります。

 [z/OS](#) チャンネルが RETRYING または STOPPED の状況の間にチャンネル・イニシエーターが停止した場合、チャンネル・イニシエーターの再開時にチャンネル状況が記憶されています。ただし、SVRCONN チャンネル・タイプのチャンネル状況は、チャンネルが STOPPED の状況の間にチャンネル・イニシエーターが停止すると再設定されます。

 [Multi](#) チャンネルが RETRYING または STOPPED の状況の間にキュー・マネージャーが停止した場合、キュー・マネージャーの再開時にチャンネル状況が記憶されています。IBM MQ 8.0 以降、これは SVRCONN チャンネルにも当てはまります。以前は、SVRCONN チャンネル・タイプのチャンネル状況は、チャンネルが STOPPED の状況の間にチャンネル・イニシエーターが停止するとリセットされていました。

キューが満杯または使用禁止のためにチャンネルがターゲット・キューにメッセージを書き込めない場合、チャンネルはある時間間隔 (メッセージ再試行間隔属性に指定された時間) で何度も (メッセージ再試行カウント属性に指定された回数) 操作を再試行します。また、メッセージ再試行出口を作成して、どのような状態のときに何回再試行を行うかを定めることもできます。チャンネルは、メッセージ再試行間隔が終わるのを待つ間、休止状態になります。

チャンネル属性についてはチャンネル属性を、メッセージ再試行出口については [メッセージ・チャンネル用のチャンネル出口プログラム](#) を参照してください。

#### サーバー接続チャンネルの制限

サーバー接続チャンネルの制限を設定して、クライアント・アプリケーションがキュー・マネージャーのチャンネル・リソース **MAXINST** を使い果たしたり、単一のクライアント・アプリケーションがサーバーの接続チャンネル容量 **MAXINSTC** を使い果たしたりすることを避けることができます。

**DEFINE CHANNEL** コマンドを使用して、**MAXINST** および **MAXINSTC** を設定します。

最大合計数のチャンネルは、個々のキュー・マネージャーでいつでもアクティブにできます。サーバー接続チャンネル・インスタンスの合計数は、アクティブ・チャンネルの最大数に含まれます。

サーバー接続チャンネルの同時インスタンスを最大でいくつ開始できるかを指定しないと、単一のサーバー接続チャンネルに接続する単一のクライアント・アプリケーションが、使用可能なアクティブ・チャンネルの最大数を使い果たす可能性があります。アクティブ・チャンネルの最大数に達すると、キュー・マネージャー上で他のチャンネルは開始できなくなります。この状態を避けるために、どのクライアントが開始したかに関係なく、個々のサーバー接続チャンネルの開始可能な同時インスタンスの数を制限する必要があります。

制限の値が、現在実行中のサーバー接続チャンネルのインスタンス数より少ない数まで引き下げられる場合でも (たとえ 0 でも)、実行中のチャンネルは影響を受けません。十分な数の既存のインスタンスが実行を終了して、現在実行中のインスタンスの数が制限の値を下回らないと、新規インスタンスは開始できません。

多くの異なるクライアント接続チャンネルが、個々のサーバー接続チャンネルに接続することもできます。どのクライアントが開始したかに関係なく、個々のサーバー接続チャンネルの開始可能な同時インスタンス数の制限により、いずれのクライアントもキュー・マネージャーの最大アクティブ・チャンネル容量を使い果たせなくなります。個々のクライアントから開始可能な個々のサーバー接続チャンネルの同時インスタンス数も制限しておかないと、欠陥のある単一のクライアント・アプリケーションが多くの接続をオープンして、個々のサーバー接続チャンネルに割り振られているチャンネル容量を使い果たしてしまうことがあります。したがって、チャンネルを使用する必要がある他のクライアントはそれに接続できなくなります。この状態を避けるために、個々のクライアントから開始可能な、個々のサーバー接続チャンネルの同時インスタンスの数を制限する必要があります。

個々のクライアントの制限値が、個々のクライアントから現在実行中のサーバー接続チャンネルのインスタンス数より少ない数まで引き下げられる場合でも (たとえ 0 でも)、実行中のチャンネルは影響を受けません。しかし、クライアントが開始した十分な数の既存のインスタンスが実行を終了して、現在実行中のインスタンスの数がこのパラメーターの値を下回るまで、新しい制限を超過する個々のクライアントからサーバー接続チャンネルの新規インスタンスを開始することはできません。

## 関連情報

[チャンネル属性とチャンネル・タイプ](#)

[DEFINE CHANNEL](#)

## チャンネルの相手側がまだ使用可能であるかどうかの検査

チャンネルの相手側が使用可能であることを検査するために、ハートビート間隔、キープアライブ間隔、およびタイムアウトの受信を使用できます。

## ハートビート

ハートビート間隔チャンネル属性を使用して、[ハートビート間隔 \(HBINT\)](#) の説明のように、伝送キューにメッセージがない場合にそのフローが送信 MCA から渡されるように指定できます。

## キープアライブ

IBM MQ for z/OS では、TCP/IP をトランスポート・プロトコルとして使用している場合、**Keepalive** 間隔チャンネル属性 (KAINT) の値を指定することもできます。**Keepalive** 間隔には、ハートビート間隔よりも大きい値で、かつ、切断値よりも小さい値を指定するようお勧めします。この属性を使用して、[キープアライブ間隔 \(KAINT\)](#) の説明のように、各チャンネルのタイムアウト値を指定できます。

IBM MQ for IBM i, UNIX, Linux, and Windows システムでは、トランスポート・プロトコルとして TCP を使用する場合、`keepalive=yes` を設定できます。このオプションを指定すると、TCP は、接続の相手側がまだ使用可能であるかどうかを定期的に検査します。使用可能でない場合、チャンネルは終了されます。このオプションについては、[キープアライブ間隔 \(KAINT\)](#) を参照してください。

TCP エラーを報告する信頼性の低いチャンネルがある場合は、**Keepalive** オプションを使用するとチャンネルがリカバリーしやすくなります。

時間間隔を指定して、**Keepalive** オプションの動作を制御することができます。時間間隔を変更するとき、変更後に開始される TCP/IP チャンネルだけが影響を受けます。時間間隔として選択する値は、チャンネルの切断間隔の値よりも小さい値にしなければなりません。

**Keepalive** オプションの使用に関する詳細については、[DEFINE CHANNEL](#) コマンドの [KAIN](#)T パラメータを参照してください。

## 受信タイムアウト

トランスポート・プロトコルとして TCP を使用している場合、一定の期間データを受信しないと、使用されていない非 MQI チャンネル接続の受信側もクローズされます。その期間を示す *receive time-out* 値は、HBINT (ハートビート間隔) 値に従って決定されます。

IBM MQ for IBM i、UNIX、Linux、and Windows システムでは、*receive time-out* 値は以下のように設定されます。

1. 折衝が行われる前の初期のいくつかのフローでは、この *receive time-out* 値はチャンネル定義の HBINT 値の 2 倍です。
2. チャンネルが HBINT 値を折衝したら、HBINT が 60 秒未満に設定されていると、*receive time-out* 値はその HBINT 値の 2 倍に設定されます。HBINT が 60 秒以上に設定されている場合は、*receive time-out* 値は HBINT の値より 60 秒大きい値に設定されます。

IBM MQ for z/OS では、*receive time-out* 値は次のように設定されます。

1. 折衝が行われる前の初期のいくつかのフローでは、この *receive time-out* 値はチャンネル定義の HBINT 値の 2 倍です。
2. RCVTIME を設定すると、タイムアウト次いずれかに設定されます。
  - 折衝された HBINT に定数を掛けた値
  - 折衝された HBINT に定数 (秒数) を加えた値
  - 定数 (秒数)

どれに設定されるかは RCVTTYPE パラメータによって決まり、RCVTMIN が適用される場合にはその制限に従います。RCVTTYPE(EQUAL) が構成されている場合には、RCVTMIN は適用されません。RCVTIME の定数値を使用し、かつハートビート間隔を使用する場合は、ハートビート間隔よりも小さい RCVTIME を指定しないでください。RCVTIME、RCVTMIN、および RCVTTYPE 属性の詳細については、[ALTER QMGR](#) コマンドを参照してください。

### 注:

1. これらの値のいずれかがゼロの場合、タイムアウトは存在しません。
2. ハートビートをサポートしていない接続では、HBINT 値はステップ 2 の折衝でゼロとなり、したがって、タイムアウトは存在しません。このため、TCP/IP KEEPALIVE を使用しなければなりません。
3. 会話の共有を使用するクライアント接続では、MQGET が未解決のときだけでなく、常に (両端から) ハートビートがチャンネルを流れることができます。
4. 会話の共有が使用されていないクライアント接続では、クライアントが待機条件付きの MQGET 呼び出しを発行した場合のみ、ハートビートがサーバーから流れます。この理由から、クライアント・チャンネルではハートビート間隔を極端に短く設定しないことをお勧めします。例えば、ハートビートが 10 秒に設定されている場合に、コミット処理にかかった時間が 20 秒を超えていると、この処理中には何もデータが流れていないので MQCMIT 呼び出しは失敗します (MQRC\_CONNECTION\_BROKEN)。このエラーは、使用している作業単位が大きいほど、よく発生します。ただし、ハートビート間隔に対して適切な値が選択されている場合、極端に長い時間がかかるのは、待機条件を指定された MQGET しかないため、このエラーが発生することはありません。

SHARECNV がゼロでない場合、クライアントは全二重接続を使用します。つまり、クライアントはすべての MQI 呼び出しの間もハートビートを実行できます (そして実際に実行します)。

5. IBM WebSphere MQ 7 クライアント・チャンネルでは、ハートビートはサーバーおよびクライアント・サイドの両方から流れることがあります。片側のタイムアウトは、 $2 \times \text{HBINT}$  (HBINT が 60 秒未満の場合) および  $\text{HBINT} + 60$  (HBINT が 60 秒より長い場合) に基づいています。
6. ハートビート間隔の 2 倍の時間が経過した後で接続を打ち切ることも有効です。これは、データまたはハートビート・フローが、少なくとも各ハートビート間隔おきに発生するためです。ハートビート間隔を小さく設定しすぎると、特にチャンネル出口を使用している場合に問題が生じる可能性があります。例えば、HBINT 値が 1 秒であり、送信出口または受信出口が使用されていると、受信側は 2 秒だけ待つ

てからチャンネルを打ち切ります。MCA がメッセージの暗号化などの作業を実行している場合は、この値は小さすぎる可能性があります。

## 推奨設定

### IBM MQ for z/OS

初期の開始点として、次を使用できます。

```
/cpcf ALTER QMGR TCPKEEP(YES) RCVTTYPE(ADD) RCVTIME(60) ADOPTMCA(ALL) ADOPTCHK(ALL)
```

ここで、cpcf はキュー・マネージャー・サブシステムのコマンド接頭部です。

さまざまなパラメーターについての詳細は、[QMGR の変更](#)および [IBM MQ ネットワーク可用性](#)を参照してください。

送信側の IP アドレスが複数のアドレスに変換される可能性がある場合は、ADOPTCHK を ALL ではなく QMNAME に設定する必要があります。

### IBM MQ for Multiplatforms

qm.ini に、次の情報を追加します。

```
TCP:
KeepAlive=Yes
CHANNELS:
AdoptNewMCA=ALL
AdoptNewMCACheck=ALL
```

詳しくは、[ALTER QMGR](#)、[分散キューイング用の構成ファイル・スタンザ](#)、および [121 ページの『channels スタンザの属性』](#)を参照してください。

送信側の IP アドレスが複数のアドレスに変換される可能性がある場合は、**AdoptNewMCACheck** を ALL ではなく QMNAME に設定する必要があります。

## MCA の採用

MCA の採用機能を使用して、IBM MQ で、受信側チャンネルを取り消してから、代替りの別のチャンネルを起動することができます。

チャンネルで通信が失われると、受信側チャンネルは「通信受信」状態のままになる可能性があります。通信が再確立されたら、送信側チャンネルは再接続を試みます。受信側チャンネルがすでに実行されていることがリモート・キュー・マネージャーで検出されると、同じ受信側チャンネルの別のバージョンを開始することはできません。この問題が生じたら、ユーザーが介入してそれを解決するか、またはシステム・キープアライブを使用する必要があります。

Adopt MCA 機能は、この問題を自動的に解決します。IBM MQ でこの機能を使用して、受信側チャンネルを取り消してから、代替りの別のチャンネルを起動することができます。

## 関連情報

[管理 IBM MQ](#)

[管理 IBM MQ for z/OS](#)

[管理 IBM MQ for IBM i](#)

## チャンネルの停止および静止

切断時間間隔が満了する前に、チャンネルを停止および静止することができます。

メッセージ・チャンネルは、キュー・マネージャー間の接続を長時間保持するように設計されており、正規の終了が切断時間間隔チャンネル属性だけによって制御されるようになっています。このメカニズムは、切断時間間隔が満了する前にオペレーターがチャンネルを終了する必要がない限り、効率的に働きます。このような必要が生じるのは、以下の場合です。

- システムの静止

- リソースの保存
- チャンネルの片側からの一方的な処置

このような場合、チャンネルを停止することができます。以下の方法を使用します。

- STOP CHANNEL MQSC コマンド
- STOP CHANNEL PCF コマンド
- IBM MQ エクスプローラー

- ▶ **IBM i** ▶ **z/OS** プラットフォーム固有のその他のメカニズムは以下のとおりです。

▶ **z/OS** **z/OS の場合:**

「Stop a Channel (チャンネルの停止)」パネル

▶ **IBM i** **IBM i の場合:**

ENDMQMCHL CL コマンドまたは WRKMQMCHL パネルの END オプション

これらのコマンドを使用してチャンネルを停止する場合は、以下の 3 つのオプションを指定できます。

### QUIESCE

QUIESCE オプションを指定すると、チャンネルを停止する前にメッセージの現行バッチの終了が試みられます。

### FORCE

FORCE オプションを指定すると、即時にチャンネルの停止が試みられますが、再始動時には、チャンネルが未確定のままである可能性があるため、再同期化が必要になることがあります。

▶ **z/OS** IBM MQ for z/OS では、FORCE は、進行中のメッセージ再割り振りをすべて中断します。これにより、BIND\_NOT\_FIXED メッセージの再割り振りが不完全になったり、正しくない順序になったりする場合があります。

### TERMINATE

TERMINATE オプションを指定すると、即時にチャンネルの停止が試みられ、チャンネルのスレッドまたは処理が終了します。

▶ **z/OS** IBM MQ for z/OS では、TERMINATE は、進行中のメッセージ再割り振りをすべて中断します。これにより、BIND\_NOT\_FIXED メッセージの再割り振りが不完全になったり、正しくない順序になったりする場合があります。

これらのすべてのオプションは、チャンネルを停止状態のままにしておくので、オペレーターの介入によって再始動する必要があります。

送信側でチャンネルを停止するのは効果的ですが、再始動には、オペレーターの介入が必要です。チャンネルの受信側では、MCA は送信側からのデータを待機しており、受信側からチャンネルの正規の終了を開始する方法はないので、受信側でチャンネルを終了させるのは困難です。停止コマンドは、MCA がデータ待ちから解放されるまで保留されます。

必要とされる操作の特性によって、次の 3 つのチャンネル使用法を推奨します。

- チャンネルを長期間実行させたい場合は、正規の方法でチャンネルを終了できるのは、送信側からだけであることに注意してください。チャンネルに割り込みが行われる (停止される) と、そのチャンネルの再始動にはオペレーターの介入 (START CHANNEL コマンド) が必要です。
- 送信するメッセージがあるときだけチャンネルをアクティブにする場合は、切断間隔を非常に小さな値に設定してください。デフォルト設定は高い値であるため、このようなレベルの制御が必要なチャンネルにはお勧めできません。受信側チャンネルに割り込みを行うのは困難なため、最も効率的なオプションはワークロードの需要に応じて自動的にチャンネルの切断および再接続を行うことです。ほとんどのチャンネルでは、切断時間間隔に適切な値を見つけて設定することができます。
- ハートビート間隔属性を使用して、送信するメッセージがない期間に送信側 MCA から受信側 MCA へハートビート・フローを送信させることができます。このアクションによって、受信側 MCA は待機状態から解放され、切断時間間隔の満了を待たずにチャンネルを静止させる機会が与えられます。ハートビート間隔は、切断時間間隔値よりも小さい値にします。

注:

1. サーバー・チャンネルには特に小さい切断時間間隔値を設定するかまたはハートビートを使用することをお勧めします。小さい値を使用するのは、サーバー・チャンネルから送信すべきメッセージがないときに、(例えば、チャンネルが取り消されて) 要求側のチャンネルが異常終了するような場合に備えるためです。切断間隔の値が高く設定され、ハートビートが使用されていない場合、サーバーでは、要求側の終了は検出されません(検出されるのは、次に要求側にメッセージを送信しようとした場合のみです)。サーバーがまだ実行中の際、さらに伝送キューに入ってくるメッセージを入手するために、サーバーは、入力専用としてそのキューをオープンしたままにしておきます。要求側からチャンネルを再始動しようとする、サーバーが入力専用としてまだ伝送キューをオープンにしているため、開始要求はエラーを受け取ります。サーバー・チャンネルを停止して、要求側からサーバー・チャンネルを再始動する必要があります。

## 停止したチャンネルの再始動

チャンネルが STOPPED 状態になるときは、手動でチャンネルを再始動しなければなりません。

## このタスクについて

送信側またはサーバー・チャンネルでは、チャンネルが停止状態になった場合、関連する伝送キューは GET(DISABLED) に設定され、トリガー操作はオフに設定されます。開始要求が受信されたとき、これらの属性は自動的にリセットされます。

**z/OS** チャンネルが RETRYING または STOPPED の状況の間にチャンネル・イニシエーターが停止した場合、チャンネル・イニシエーターの再開時にチャンネル状況が記憶されています。ただし、SVRCONN チャンネル・タイプのチャンネル状況は、チャンネルが STOPPED の状況の間にチャンネル・イニシエーターが停止すると再設定されます。

**Multi** チャンネルが RETRYING または STOPPED の状況の間にキュー・マネージャーが停止した場合、キュー・マネージャーの再開時にチャンネル状況が記憶されています。IBM MQ 8.0 以降、これは SVRCONN チャンネルにも当てはまります。以前は、SVRCONN チャンネル・タイプのチャンネル状況は、チャンネルが STOPPED の状況の間にチャンネル・イニシエーターが停止するとリセットされていました。

## 手順

- 以下のいずれかの方法で、チャンネルを再始動します。
  - [START CHANNEL MQSC コマンド](#)を使用します。
  - [Start Channel PCF コマンド](#)を使用します。
  - [IBM MQ Explorer](#)を使用します。
  - **z/OS** z/OS で、[「チャンネルの開始 \(Start a channel\)」](#)パネルを使用します。
  - **IBM i** IBM i で、[STRMQMCHL CL コマンド](#)または [WRKMQMCHL パネル](#)の START オプションを使用します。

## 未確定チャンネル

未確定チャンネルとは、メッセージの送受信の対象となっているリモート・チャンネルが未確定のチャンネルのことです。

これと、キューに対するメッセージ・コミットの対象になるキュー・マネージャーが未確定であることとの相違点に注意してください。

バッチ・ハートビート・チャンネル・パラメーター (BATCHHB) を使用することによって、チャンネルが未確定になる機会を減らすことができます。このパラメーターの値を指定すると、送信側チャンネルは、リモート・チャンネルがアクティブになっていることを確認してからでなければ、一切処置を行いません。応答が受信されない場合は、受信側チャンネルはアクティブではないと見なされます。メッセージのロールバックおよび再経路指定が可能なので、送信側チャンネルは未確定になりません。このため、チャンネルが未確定になるのは、受信側チャンネルがアクティブになっていることを送信側チャンネルが確認する期間と、送信されたメッセージを受信側チャンネルが受信したことを送信側チャンネルが確認する期間だけとなります。バッチ・ハートビート・パラメーターの詳細については、[チャンネル属性](#)を参照してください。

未確定チャンネルの問題は、通常は自動的に解決されます。通信が中断され、受信状態が不明のメッセージ・バッチが送信側に存在する状況でチャンネルが未確定になる場合でも、その通信が再度確立されたときに、この状態は解決されます。そのために、シーケンス番号と LUWID レコードが保持されています。LUWID 情報が交換されるまでチャンネルは未確定になります。また、未確定になるメッセージ・バッチはチャンネルごとに 1 つだけです。

必要な場合は、チャンネルを手動で再同期化できます。手動でという表現には、オペレーターまたは、IBM MQ システム管理コマンドを含むプログラムの使用が含まれます。手動による再同期プロセスは次のように行われます。ここでは MQSC コマンドを使用しますが、PCF の同等コマンドを使用することもできます。

1. DISPLAY CHSTATUS コマンドを使用して、チャンネルの **それぞれの** 側に対して最後に行われた作業論理単位 ID (LUWID) を検出することができます。次のコマンドを使用してこれを行います。

- 未確定側のチャンネルの場合

```
DISPLAY CHSTATUS( name ) SAVED CURLWID
```

さらにチャンネルを識別するには、CONNAME および XMITQ パラメーターを使用できます。

- 受信側チャンネルの場合

```
DISPLAY CHSTATUS( name ) SAVED LSTLUWID
```

さらにチャンネルを識別するには、CONNAME パラメーターを使用できます。

未確定の可能性があるのはチャンネルの送信側だけであるため、コマンドは異なります。受信側が未確定になることはありません。

IBM MQ for IBM i では、DISPLAY CHSTATUS コマンドは、STRMQMMQSC コマンドまたは「MQM チャンネル状況の操作」CL コマンド (WRKMQMCHST) を使用してファイルから実行できます。

2. 2 つの LUWID が同じである場合、受信側は送信側が未確定であると見なす作業単位をすでにコミットしています。したがって、送信側は伝送キューから未確定のメッセージを除去して、それを再度使用可能にすることができます。これは、次のチャンネル RESOLVE コマンドで実行されます。

```
RESOLVE CHANNEL( name ) ACTION(COMMIT)
```

3. 2 つの LUWID が異なっている場合、受信側は送信側が未確定であると見なす作業単位をコミットしていません。送信側では、伝送キューに未確定のメッセージを保存しておき、それらを再送信する必要があります。これは、次のチャンネル RESOLVE コマンドで実行されます。

```
RESOLVE CHANNEL( name ) ACTION(BACKOUT)
```

**IBM i** IBM MQ for IBM i では、「MQM チャンネルの解決」コマンド (RSVMQMCHL) を使用できます。

以上の処理で、チャンネルは未確定ではなくなります。必要であれば伝送キューを別のチャンネルで使用できます。

## 問題判別

問題判別には、コマンドが実行依頼されるときに検出された問題と、チャンネルの操作中に検出された問題の 2 つの別個の側面があります。

## コマンドの妥当性検査

コマンドおよびデータが処理のために受け入れられるには、エラーが除去されていなければなりません。妥当性検査で検出されたエラーは、ただちにエラー・メッセージによってユーザーに通知されます。

問題診断では、まず最初にこれらのエラー・メッセージの解釈が行われ、訂正処置が取られます。

## 処理の問題

通常のチャンネル操作中に検出されたエラーは、システム・コンソールに表示されるか、またはシステム・ログに記録されます。問題診断では、まず最初にログからすべての関連情報が収集され、さらに問題識別のための分析が続けられます。

可能な場合には、コマンドを開始した端末に確認メッセージとエラー・メッセージが戻されます。

IBM MQ は、会計および統計データを作成します。これを使用して使用率やパフォーマンスの傾向を確認できます。 **Multi** Multiplatforms では、この情報が PCF レコードとして作成されます。構造体データ・タイプを参照してください。 **z/OS** z/OS では、この情報は SMF レコードとして作成されます。パフォーマンスと資源の使用量のモニターを参照してください。

## メッセージおよびコード

問題の一次診断に役立つメッセージやコードについては、「[メッセージと理由コード](#)」をご覧ください。

## メッセージの安全性

IBM MQ の通常のリカバリー機能の他に、分散キュー管理は、メッセージ・チャンネルの両側で調整された同期点プロシージャーを使用して、メッセージが適切に送達されることを確認します。このプロシージャーはエラーを検出すると、問題を調査できるようにチャンネルをクローズし、チャンネルが再開されるまでメッセージを伝送キューに安全に保管します。

同期点プロシージャーには、チャンネルの開始時に未確定状態をリカバリーしようとするという利点が追加されています。(未確定は、同期点が要求されたが、要求の結果がまだ認識されていない、リカバリー単位の状況です。) また、この機能には、次の 2 つの機能が関連付けられています。

1. コミットまたはバックアウトによる解決
2. シーケンス番号のリセット

ほとんどの場合、チャンネルは自動的にリカバリーするので、これらの機能は例外状況でのみ使用されます。

## 高速、非持続メッセージ

非持続メッセージ速度 (NPMSPEED) チャンネル属性を使用して、チャンネルのすべての非持続メッセージの送達速度を上げるように指定できます。この属性の詳細については、[非持続メッセージ速度 \(NPMSPEED\)](#) を参照してください。

高速で、非持続メッセージの転送中にチャンネルが終了した場合、メッセージは失われることがあります。必要な場合にリカバリーの措置をとることはアプリケーション側の処理範囲です。

受信側チャンネルが宛先キュー内にメッセージを入れられない場合、送達不能キューが定義されていれば、メッセージはそのキューに入れられます。定義されていないと、メッセージは廃棄されます。

注: チャンネルの片側でこのオプションをサポートしていない場合、そのチャンネルは通常で速度で実行されます。

## 未配布メッセージ

メッセージを送達できない場合の処理については、[182 ページの『メッセージを送達できない場合の処理』](#)を参照してください。

## メッセージを送達できない場合の処理

メッセージを送達できない場合、MCA はメッセージをいくつかの方法で処理できます。再試行するか、送信側へ返信するか、または送達不能キューへ書き込むことができます。

[183 ページの図 24](#) は、MCA が宛先キューにメッセージを入れることができないときに行われる処理を示したものです。(図に示したオプションはすべてのプラットフォームには当てはまりません。)

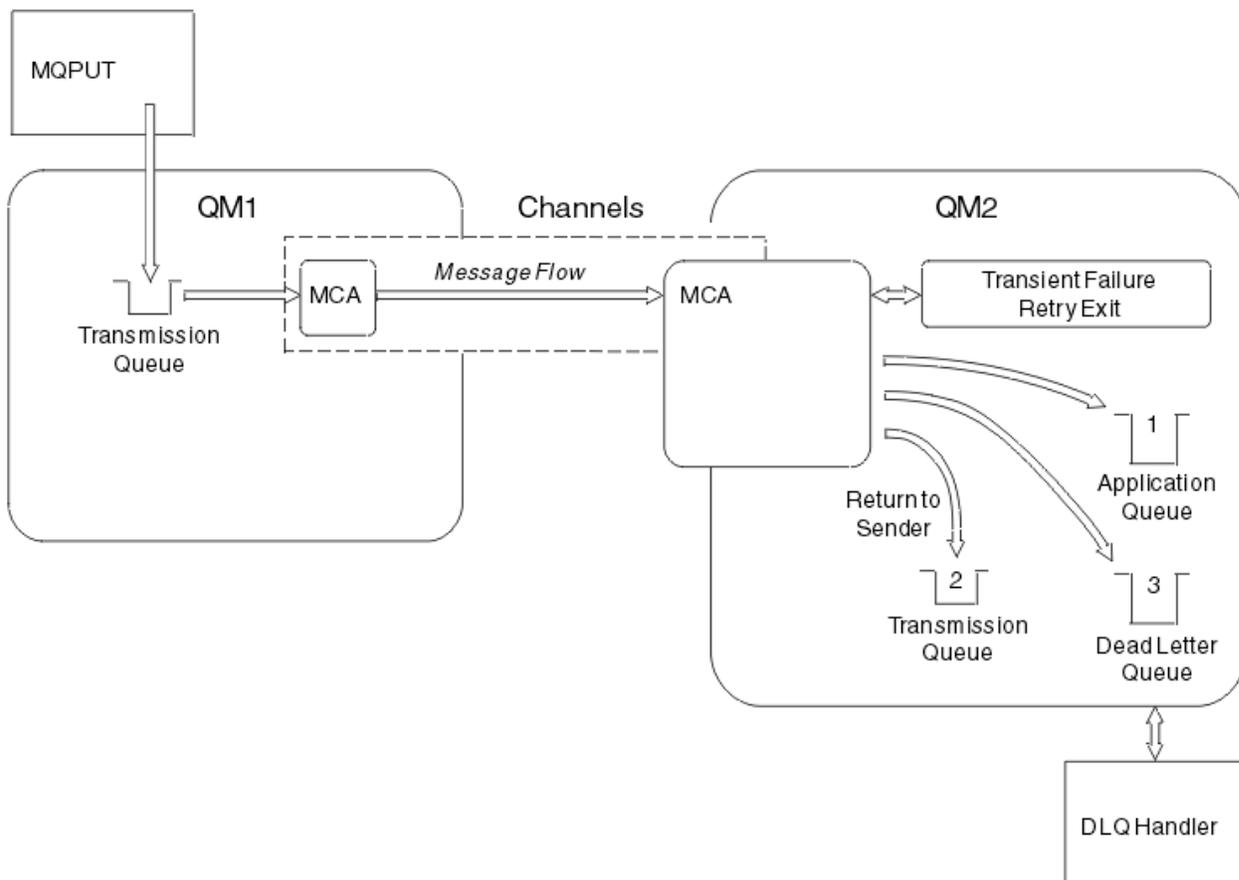


図 24. メッセージを送達できない場合の処理

図に示すように、MCA は、送達できないメッセージについていくつかの処置を行うことができます。行われる処置は、チャンネルの定義時に指定したオプションおよびメッセージの MQPUT レポート・オプションで指定したオプションによって決まります。

### 1. メッセージ再試行

一時的な理由 (キューが満杯であるなど) で MCA がターゲット・キューにメッセージを書き込むことができない場合、MCA は待機して、あとで操作を再試行できます。MCA が待機するかどうか、待機時間、および試行の回数を決めることができます。

- チャンネルを定義するときに、MQPUT エラーの場合のメッセージ再試行時間および間隔を指定することができます。キューが満杯または書き込み禁止のためメッセージを宛先キューに書き込むことができない場合、MCA は、指定された時間間隔で指定された回数だけ操作を試行します。
- 独自のメッセージ再試行出口を作成することができます。この出口を使用して、MCA に MQPUT 操作または MQOPEN 操作を再試行させる条件を指定することができます。チャンネルの定義時に出口の名前を指定します。

### 2. 送信側への返信

メッセージ再試行が失敗した場合、または別のタイプのエラーが検出された場合、MCA は発信元にメッセージを送り返すことができます。送信側への返信を可能にするには、メッセージを元のキューに書き込むとき、メッセージ記述子に以下のオプションを指定する必要があります。

- MQRO\_EXCEPTION\_WITH\_FULL\_DATA レポート・オプション
- MQRO\_DISCARD\_MSG レポート・オプション
- 応答先キューおよび応答先キュー・マネージャーの名前

宛先キューにメッセージを書き込むことができない場合、MCA は、元のメッセージが入った例外レポートを生成し、伝送キューに書き込んで、元のメッセージで指定されている応答先キューに送信されるよ

うにします。(応答先キューが MCA と同じキュー・マネージャー上にある場合は、メッセージは伝送キューには送られず、その応答先キューに直接書き込まれます。)

### 3. 送達不能キュー

送達することも戻すこともできないメッセージは、送達不能キュー (DLQ) に入れられます。DLQ ハンドラーを使用してメッセージを処理することができます。この処理については、[送達不能キューのメッセージの処理](#) (IBM MQ for UNIX、Linux、および Windows システムの場合)、および [送達不能キュー・ハンドラー・ユーティリティー \(CSQUDLQH\)](#) (z/OS システムの場合) で説明されています。送達不能キューが使用できない場合、送信側 MCA は、伝送キューにメッセージを残したまま、チャンネルは停止します。高速チャンネルでは、送達不能キューに書き込めない非持続メッセージは失われます。

IBM WebSphere MQ 7.0 では、ローカル送達不能キューが定義されていない場合、リモート・キューは使用できないか、定義されません。また、リモート送達不能キューがない場合、送信側チャンネルは RETRY 状態となり、メッセージは自動的に伝送キューにロールバックされます。

#### 関連情報

[送達不能キューの使用 \(USEDLQ\)](#)

## チャンネルのトリガー操作

IBM MQ には、キューで特定の条件が満たされると自動的にアプリケーションを開始する機能が用意されています。この機能は、トリガー操作と呼ばれています。

ここでの説明は、トリガー操作のおおよその概念を示すことを目的としています。完全な説明については、[トリガーによる IBM MQ アプリケーションの開始](#)を参照してください。

プラットフォーム特有の情報については、以下のセクションを参照してください。

- Windows、UNIX and Linux システムについては、[185 ページの『UNIX, Linux, and Windows でのチャンネルのトリガー操作』](#)を参照してください。
- **IBM i** IBM i については、[186 ページの『IBM MQ for IBM i におけるチャンネルのトリガー操作』](#)を参照してください。
- **z/OS** z/OS については、[731 ページの『伝送キューおよびチャンネルのトリガー操作』](#)を参照してください。

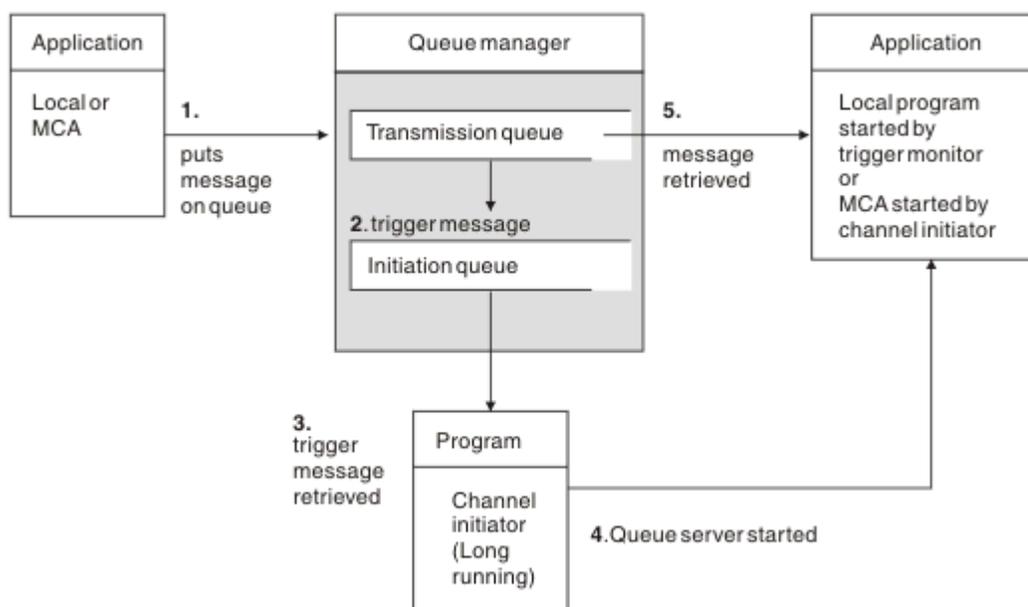


図 25. トリガー操作の概念

トリガー操作に必要なオブジェクトが、[184 ページの図 25](#) に示されています。この図には、以下のイベント順序が示されています。

1. アプリケーションまたはメッセージ・チャンネル・エージェント (MCA) からのメッセージがローカル・キュー・マネージャーによって伝送キューに書き込まれます。
2. トリガー操作の条件が満たされると、開始キューにトリガー・メッセージがローカル・キュー・マネージャーによって書き込まれます。
3. 長期実行チャンネル・イニシエーター・プログラムは開始キューをモニターし、到達したメッセージを取り出します。
4. チャンネル・イニシエーターは、トリガー・メッセージに入れられた情報に従ってそれら进行处理します。この情報には、チャンネル名も入っていることがあります。その場合、対応する MCA が開始されます。
5. ローカル・アプリケーションまたは MCA は、起動されると、伝送キューからのメッセージを検索します。

このシナリオを設定するには、以下のことを行う必要があります。

- 対応する属性の開始キューの名前 (つまり SYSTEM.CHANNEL.INITQ) でローカル・キューを作成する。
- 開始キュー (つまり、SYSTEM.CHANNEL.INITQ) が存在することを確認する。
- チャンネル・イニシエーターが使用可能であって、しかも実行されていることを確認する。チャンネル・イニシエーターには、開始コマンドに入れて開始キューの名前を提供しなければなりません。  
 z/OS では、開始キューの名前は固定であるため、開始コマンドでは使用されません。
- オプションで、トリガー操作用のプロセス定義が存在していない場合はそれを作成し、*UserData* フィールドにサービス対象となるチャンネルの名前が入っていることを確認する。プロセス定義を作成する代わりに、伝送キューの **TriggerData** 属性にチャンネル名を指定することもできます。IBM MQ  
 IBM i、UNIX, Linux, and Windows システムの場合) では、チャンネル名をブランクとして指定できます。この場合、この伝送キューを持つ使用可能な最初のチャンネル定義が使用されます。
- サービス対象となるプロセス定義名 (適用できる場合)、開始キュー名、および最適と思われるトリガー操作特性が伝送キュー定義に入っていることを確認する。トリガー制御属性は、必要に応じてトリガー操作を有効/無効にします。

注:

1. チャンネル・イニシエーターは、チャンネルの開始に使われる開始キューをモニターするための「トリガー・モニター」として働きます。
2. 開始キューおよびトリガー・プロセスを使用して、任意の数のチャンネルを起動できます。
3. 開始キューとトリガー・プロセスは、いくつでも定義できます。
4. FIRST トリガー・タイプを使用して、システムで過剰にチャンネルが開始されないようにすることをお勧めします。

## UNIX, Linux, and Windows でのチャンネルのトリガー操作



トリガー操作の対象となるプロセスを定義するプロセス定義を IBM MQ 内に作成します。MQSC コマンド DEFINE PROCESS を使用して、伝送キューにメッセージが到達したときに起動されるプロセスを指定して、プロセス定義を作成します。プロセス定義の USERDATA 属性には、その伝送キューでサービスを受けるチャンネルの名前が含まれます。

ローカル・キュー (QM4) を定義します。そのためには、トリガー・メッセージを開始キュー (IQ) に書き込んで、アプリケーションにトリガー操作を行うように指定します。それによって、チャンネル (QM3.TO.QM4) が開始されます。

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(P1) USAGE(XMITQ)
```

開始するアプリケーション (プロセス P1) を定義します。

```
DEFINE PROCESS(P1) USERDATA(QM3.TO.QM4)
```

また、IBM MQ for UNIX、Linux および Windows システムの場合、伝送キューの TRIGDATA 属性にチャンネル名を指定すると、プロセスを定義する必要がなくなります。

ローカル・キュー (QM4) を定義します。次のように、トリガー・メッセージをデフォルト開始キュー SYSTEM.CHANNEL.INITQ に書き込んで、チャンネル (QM3.TO.QM4) を開始させるアプリケーション (プロセス P1) を起動するように指定します。

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ)  
USAGE(XMITQ) TRIGDATA(QM3.TO.QM4)
```

チャンネル名を指定しないと、チャンネル・イニシエーターが、指定された伝送キューと関連付けられたチャンネルを発見するまで、チャンネル定義ファイルを探します。

## IBM MQ for IBM i におけるチャンネルのトリガー操作

### IBM i

IBM MQ for IBM i におけるチャンネルのトリガー操作は、チャンネル・イニシエーター・プロセスで実現されます。開始キュー SYSTEM.CHANNEL.INITQ のチャンネル・イニシエーター・プロセスは、キュー・マネージャーで自動的に開始されます (キュー・マネージャーの SCHINIT 属性を変更してそれを使用不可にした場合を除く)。

SYSTEM.CHANNEL.INITQ を開始キューとして指定し、そのキューでのトリガーを使用可能にすることで、チャンネルの伝送キューをセットアップします。チャンネル・イニシエーターは、この伝送キューを指定する、使用可能な最初のチャンネルを開始します。

```
CRTMQMQ QNAME(MYXMITQ1) QTYPE(*LCL) MQMNAME(MYQMGR)  
TRGENBL(*YES) INITQNAME(SYSTEM.CHANNEL.INITQ)  
USAGE(*TMQ)
```

STRMQMCHLI コマンドで最大 3 つのチャンネル・イニシエーター・プロセスを手動で開始でき、異なる開始キューを指定できます。また、伝送キューを処理できるチャンネルを複数指定し、開始するチャンネルを選択することもできます。この機能は、以前のリリースと互換性を持たせるために今でも提供されています。この使用は推奨されません。

**注:** 一度に 1 つのチャンネルのみが伝送キューを処理できます。

```
STRMQMCHLI QNAME(MYINITQ)
```

TRGENBL(\*YES) を指定して、チャンネルの伝送キューをセットアップします。開始しようとするチャンネルを選択するには、TRIGDATA フィールドにチャンネル名を指定します。以下に例を示します。

```
CRTMQMQ QNAME(MYXMITQ2) QTYPE(*LCL) MQMNAME(MYQMGR)  
TRGENBL(*YES) INITQNAME(MYINITQ)  
USAGE(*TMQ) TRIGDATA(MYCHANNEL)
```

### 関連概念

[187 ページの『チャンネル・イニシエーターの開始と停止』](#)

トリガー操作は、チャンネル・イニシエーターのプロセスによって実行されます。

### 関連タスク

[140 ページの『分散キューイングの構成』](#)

このセクションでは、キュー定義、チャンネル定義、トリガー操作および同期点プロシージャを含む、IBM MQ インストール間の相互通信の詳細について説明します。

### 関連情報

[UNIX, Linux, and Windows でのチャンネル・プログラム](#)

## チャンネル・イニシエーターの開始と停止

トリガー操作は、チャンネル・イニシエーターのプロセスによって実行されます。

このチャンネル・イニシエーターの処理は、MQSC コマンド START CHINIT を使用して開始します。デフォルトの始動キューを使用していない場合、コマンドの始動キューの名前を指定します。例えば、START CHINIT コマンドを使用してデフォルト・キュー・マネージャーのキュー IQ を開始するには、以下のように入力します。

```
START CHINIT INITQ(IQ)
```

デフォルトでは、チャンネル・イニシエーターはデフォルトの開始キュー SYSTEM.CHANNEL.INITQ を使用して自動的に開始されます。すべてのチャンネル・イニシエーターを手動で開始したい場合は、以下のステップに従ってください:

1. キュー・マネージャーを作成して開始します。
2. キュー・マネージャーの SCHINIT プロパティを MANUAL に変更します。
3. キュー・マネージャーを終了して再始動します。

IBM MQ for Multiplatforms システムでは、チャンネル・イニシエーターが自動的に開始されます。開始できるチャンネル・イニシエーターの数は制限されています。デフォルト値と最大値は 3 です。UNIX and Linux システムの qm.ini ファイル内の MAXINITIATORS、および Windows システムのレジストリー内で変更できます。

チャンネル・イニシエーター実行コマンド `runmqchi` と、その他の制御コマンドの詳細については、[IBM MQ 制御コマンド](#)を参照してください。

## チャンネル・イニシエーターの停止

デフォルトのチャンネル・イニシエーターは、キュー・マネージャーを開始すると自動的に開始します。すべてのチャンネル・イニシエーターは、キュー・マネージャーが停止すると自動的に停止します。

## 初期設定および構成ファイル

チャンネル初期設定データの処理は、運用している IBM MQ プラットフォームによって異なります。

### z/OS システム



IBM MQ for z/OS では、初期化および構成情報は ALTER QMGR MQSC コマンドを使用して指定します。CSQINP2 初期設定入力データ・セットに ALTER QMGR コマンドを置いた場合、これらはキュー・マネージャーが始動するたびに処理されます。

START LISTENER などの MQSC コマンドをチャンネル・イニシエーターの始動のたびに実行するには、これらのコマンドを CSQINPX 初期設定入力データ・セットに置いて、チャンネル・イニシエーター始動タスク・プロシージャにオプションの DD ステートメント CSQINPX を指定します。

CSQINP2 と CSQINPX について詳しくは、[初期設定入力データ・セットをカスタマイズするおよび ALTER QMGR](#)を参照してください。

### Windows、 IBM i、UNIX and Linux システム

IBM MQ for Windows、 IBM i、UNIX and Linux システムには、IBM MQ インストールに関する基本構成情報を保持するための構成ファイルがあります。

構成ファイルは 2 種類あり、1 つはマシンに適用されるもので、もう 1 つは個々のキュー・マネージャーに適用されるものです。

## IBM MQ 構成ファイル

このファイルには、IBM MQ システムのすべてのキュー・マネージャーに関連する情報が保持されています。ファイル名は `mqs.ini` です。これについては、[管理 \(IBM MQ for Windows の場合\)](#)、および [IBM i の管理](#)、および UNIX and Linux システムで詳しく説明されています。

## キュー・マネージャー構成ファイル

このファイルには、1つの特別なキュー・マネージャーに関する構成情報が保持されています。ファイル名は `qm.ini` です。

このファイルは、キュー・マネージャーの作成時に作成され、キュー・マネージャーのどの面に関する構成情報も保持することができます。ファイルに保管される情報には、ログ構成が IBM MQ 構成ファイルのデフォルト設定とどのように異なるかの詳細情報が入っています。

キュー・マネージャー構成ファイルは、キュー・マネージャーが所有するディレクトリー・ツリーのルート内に入っています。例えば、`DefaultPath` 属性については、`QMNAME` というキュー・マネージャーのキュー・マネージャー構成ファイルは、次のとおりです。

UNIX and Linux システムの場合:

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

`qm.ini` ファイルの一部分を以下に示します。これは、TCP/IP リスナーがポート 2500 にあり、現行チャンネルの最大数が 200、アクティブのチャンネルの最大数が 100であることを指定します。

```
TCP:
Port=2500
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

アウトバウンド・チャンネルが使用する TCP/IP ポートの範囲を指定できます。`qm.ini` を使用してポート値の範囲の始まりと終わりを指定する方法があります。以下の例は、チャンネルの範囲を指定する `qm.ini` ファイルを示しています。

```
TCP:
StrPort=2500
EndPort=3000
CHANNELS:
MaxChannels=200
MaxActiveChannels=100
```

`StrPort` または `EndPort` の値を指定する場合は、両方の値を指定する必要があります。`EndPort` の値は、常に `StrPort` の値より大きくなければなりません。

チャンネルは、指定した範囲の両方のポート値の使用を試みます。接続が成功すると、ポート値はチャンネルが使用するポートになります。

 IBM i の場合:

```
/QIBM/UserData/mqm/qmgrs/QMNAME/qm.ini
```

Windows システムの場合:

```
C: \ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```

`qm.ini` ファイルについて詳しくは、[分散キューイング用の構成ファイル・スタンザ](#)を参照してください。

## データ変換

IBM MQ メッセージは、異なるキュー・マネージャーでのキュー間で送信したときには、データ変換を必要とする場合があります。

IBM MQ メッセージは次の 2 つの部分から構成されています。

- メッセージ記述子内の制御情報
- アプリケーション・データ

2 つのいずれの部分についても、異なるキュー・マネージャーでのキュー間で送信したときにはデータ変換を必要とする場合があります。アプリケーション・データ変換について詳しくは、[アプリケーション・データの変換](#)を参照してください。

## 独自のメッセージ・チャンネル・エージェントの作成

IBM MQ を使用すれば、独自のメッセージ・チャンネル・エージェント (MCA) プログラムの記述、または独立ソフトウェア・ベンダーからのインストールを行うことができます。

独自の MCA プログラムの作成は、IBM MQ にユーザー独自の専用通信プロトコルとの相互運用性をもたせたり、IBM MQ がサポートしていないプロトコルを介してメッセージを送信したい場合に、必要になることがあります。(相手側にある IBM MQ 提供の MCA と相互運用性をもつユーザー独自の MCA を作成することはできません。)

IBM MQ で提供されたもの以外の MCA を使用することに決めた場合は、以下の点を考慮する必要があります。

### メッセージの送受信

アプリケーションがメッセージを置く場所、例えば伝送キューなどからメッセージを取得し、それらのメッセージを、通信に使用するプロトコルで送信する送信アプリケーションを作成する必要があります。また、このプロトコルからメッセージを受け取って宛先キューに入れる受信アプリケーションも作成する必要があります。送信アプリケーションと受信アプリケーションは、特別のインターフェースではなくメッセージ・キュー・インターフェース (MQI) 呼び出しを使用します。

メッセージが一度しか送達されないようにする必要があります。この送達には、同期点調整が役立ちます。

### チャンネル制御機能

チャンネルを制御するための独自の管理機能を用意する必要があります。IBM MQ チャンネル管理機能は、チャンネルの構成用 (例えば、DEFINE CHANNEL コマンド) にもモニター用 (例えば、DISPLAY CHSTATUS) にも使用できません。

### 初期設定ファイル

初期設定ファイルが必要な場合は、独自に用意する必要があります。

### アプリケーション・データの変換

別のシステムに送信するメッセージのためのデータ変換を覚えておく必要もあります。その場合は、アプリケーションが書き込む先の伝送キューなどの場所からメッセージを検索するときに、MQGET 呼び出しで MQGMO\_CONVERT オプションを使用してください。

### ユーザー出口

ユーザー出口が必要かどうかを検討してください。ユーザー出口が必要な場合は、IBM MQ が使用するのと同じインターフェース定義を使用できます。

### トリガー発行

アプリケーションが伝送キューにメッセージを書き込む場合、伝送キュー属性を設定して、メッセージがキューに到着したときに送信 MCA が起動されるように設定できます。

### チャンネル・イニシエーター

独自のチャンネル・イニシエーターを用意する必要があります。

## 分散キュー管理に関するその他の考慮事項

分散キュー管理用に IBM MQ を準備する際に考慮する、その他のトピックです。このトピックでは、未配布メッセージ・キュー、使用中のキュー、システム拡張およびユーザー出口プログラム、ならびにトラスレッド・アプリケーションとしてチャンネルおよびリスナーを実行する方法について説明します。

### 未配布メッセージ・キュー

未配布メッセージ・キュー (送達不能キュー、略して DLQ と呼ぶ) に到達したメッセージが確実に処理されるように、一定の間隔でトリガーされ、実行される、これらのメッセージを処理するプログラムを作成します。

**Linux** **UNIX** DLQ ハンドラーは、UNIX and Linux システム上の IBM MQ で提供されます。詳しくは、[サンプル DLQ ハンドラー amqsdlq](#) を参照してください。

**IBM i** IBM MQ for IBM i について詳しくは、[IBM MQ for IBM i 送達不能キュー・ハンドラー](#) を参照してください。

### 使用中のキュー

受信側チャンネルの MCA は、メッセージが伝送されていないときでも宛先キューをオープンしておくことができます。これにより、キューが"使用中"になることが示されます。

### チャンネルの最大数

**IBM i** IBM MQ for IBM i の場合は、システムで許可されるチャンネルの最大数と、一度にアクティブ化できるチャンネルの最大数を指定できます。これらの番号は、ディレクトリー QIBM/UserData/mqm/qmgrs/queue\_manager\_name 内の qm.ini ファイルに指定します。[分散キューイング用の構成ファイル・スタンザ](#)を参照してください。

### システム拡張部分およびユーザー出口プログラム

チャンネル定義には、メッセージ処理中の特定の時期に追加プログラムを実行できるようにする機能が用意されています。これらのプログラムは IBM MQ 提供ではありませんが、ローカル要件に応じてそれぞれのインストール先で用意することができます。

これらのユーザー出口プログラムを実行するには、定義済みの名前を割り当て、チャンネル・プログラムに対する呼び出しで利用できるようにしなければなりません。ユーザー出口プログラムの名前はメッセージ・チャンネル定義に入っています。

これらのプログラムに制御を渡したり、これらのプログラムから制御が戻ってきたりするときに必要な、定義済み制御ブロック・インターフェースがあります。

これらのプログラムを呼び出す正確な位置、および制御ブロックと名前の詳細については、[メッセージ・チャンネル用のチャンネル出口プログラム](#)を参照してください。

### 承認されたアプリケーションとしてのチャンネルおよびリスナーの実行

ご使用の環境でパフォーマンスが重要な考慮事項であり、その環境が安定している場合は、FASTPATH バインディングを使用して、チャンネルおよびリスナーを承認されたものとして実行することができます。チャンネルおよびリスナーが承認されたものとして実行されるかどうかを左右する要因は 2 つあります。

- 環境変数 MQ\_CONNECT\_TYPE=FASTPATH または MQ\_CONNECT\_TYPE=STANDARD。これは、大文字小文字が区別されます。無効な値を指定すると、無視されます。
- qm.ini ファイルまたはレジストリー・ファイルのチャンネル・スタンザの MQIBindType。これを FASTPATH または STANDARD に設定できます。これは、大文字小文字を区別しません。デフォルトは STANDARD です。

必要な効果を達成するために、次のように環境変数と関連付けて MQIBindType を使用できます。

MQIBindType	環境変数	結果
STANDARD	UNDEFINED	STANDARD
FASTPATH	UNDEFINED	FASTPATH
STANDARD	STANDARD	STANDARD
FASTPATH	STANDARD	STANDARD
STANDARD	FASTPATH	STANDARD
FASTPATH	FASTPATH	FASTPATH
STANDARD	CLIENT	CLIENT
FASTPATH	CLIENT	STANDARD
STANDARD	ローカル	STANDARD
FASTPATH	ローカル	STANDARD

つまり、実際にチャネルおよびリスナーを承認されたものとして実行させるには、次の2つの方法しかありません。

1. qm.ini またはレジストリーに MQIBindType=FASTPATH を指定して、環境変数を指定しない。
2. qm.ini またはレジストリーに MQIBindType=FASTPATH を指定し、環境変数を FASTPATH に設定する。

リスナーは安定したプロセスなので、承認されたものとして実行することを検討してください。不安定なチャネル出口または STOP CHANNEL MODE(TERMINATE) コマンドを使用していないのであれば、チャネルを承認されたものとして実行することを検討してください。

## ULW UNIX, Linux, and Windows 上のチャネルのモニターおよび制御

DQM では、リモート・キュー・マネージャーに対するチャネルの作成、モニターおよび制御が必要です。コマンド、プログラム、IBM MQ Explorer、チャネル定義ファイル、および同期情報のストレージ域を使用して、チャネルを制御できます。

### このタスクについて

以下のタイプのコマンドを使用して、チャネルを制御できます。

#### IBM MQ コマンド (MQSC)

UNIX, Linux, and Windows システムの MQSC セッションでは、MQSC を単一コマンドとして使用することができます。より複雑な、あるいは複数のコマンドを出すためには、MQSC をファイルとして組み立て、それをコマンド行から実行することができます。詳細については、[MQSC コマンド](#)を参照してください。このセクションでは、分散キューイングに対して MQSC を使用する簡単な例をいくつか記載しています。

チャネル・コマンドは、IBM MQ コマンド・セット (MQSC) のサブセットです。MQSC と制御コマンドを使用して、以下のことを行うことができます。

- チャネル定義の作成、コピー、表示、変更、および削除
- チャネルの開始と停止、ping、チャネル・シーケンス番号のリセット、およびリンクを再確立できないときの未確定メッセージの解決
- チャネルに関する状況情報の表示

#### 制御コマンド

ある種の機能については、コマンド行から制御コマンドを出すこともできます。詳しくは、[IBM MQ 制御コマンド・リファレンス](#)を参照してください。

#### プログラム式コマンド形式コマンド

詳細については、[PCF コマンド](#)を参照してください。

Linux、および Windows システムでは、IBM MQ Explorer を使用できます。これは、制御コマンドまたは MQSC コマンドを使用しないで管理タスクを実行するためのグラフィカル管理インターフェースを提供します。チャンネル定義はキュー・マネージャー・オブジェクトとして保持されます。

各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM コンポーネントが備わっています。保管域には、シーケンス番号と、作業論理単位 (LUW) ID があります。これらはチャンネルの同期をとる目的で使用されます。

異なるタイプのコマンドを使用して、メッセージ・チャンネルのセットアップおよび制御を行う場合に使用できる機能のリストについては、[193 ページの表 19](#) を参照してください。

## 手順

- [192 ページの『チャンネルのセットアップおよび制御に必要な機能』](#)
- [194 ページの『オブジェクトの準備作業』](#)
- [201 ページの『Windows における通信のセットアップ』](#)
- [208 ページの『UNIX and Linux における通信のセットアップ』](#)

## 関連タスク

[214 ページの『IBM i 上のチャンネルのモニターおよび制御』](#)

リモート・キュー・マネージャーにつながるチャンネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM プログラムが備わっています。

## 関連情報

[ULW](#) [UNIX, Linux, and Windows でのチャンネル・プログラム](#)

[ULW](#) [UNIX, Linux, and Windows におけるメッセージ・チャンネルの計画例](#)

構成情報の例

チャンネルの属性

## [ULW](#) [チャンネルのセットアップおよび制御に必要な機能](#)

いくつかの IBM MQ 機能が、チャンネルをセットアップおよび制御するのに必要となる場合があります。チャンネルの機能についてはこのトピックで説明します。

チャンネル定義を作成する際には、IBM MQ が提供するデフォルト値を使用して、チャンネル名、作成しようとするチャンネルのタイプ、使用する通信方式、伝送キュー名、および接続名を指定することができます。

チャンネル名はチャンネルの両端で同じ名前になっていなければならない、またネットワーク内で固有の名前でなければなりません。ただし、IBM MQ オブジェクト名として有効な文字のみを使用するようにしてください。

他のチャンネル関連機能についての詳細は、以下のトピックを参照してください。

- [194 ページの『オブジェクトの準備作業』](#)
- [195 ページの『関連オブジェクトの作成』](#)
- [195 ページの『デフォルト・オブジェクトの作成』](#)
- [195 ページの『チャンネルの作成』](#)
- [196 ページの『チャンネルの表示』](#)
- [196 ページの『チャンネル状況の表示』](#)
- [197 ページの『Ping を使用したリンクの検査』](#)
- [197 ページの『チャンネルの開始』](#)
- [199 ページの『チャンネルの停止』](#)
- [199 ページの『チャンネルの名前変更』](#)
- [200 ページの『チャンネルのリセット』](#)

• 200 ページの『チャンネルの未確定メッセージの解決』

193 ページの表 19 は、必要になる場合のある IBM MQ 機能の全リストです。

表 19. UNIX, Linux, and Windows システムで必要な機能			
関数	制御コマンド	MQSC	IBM MQ エクスプローラーで同等のコマンドがあるか
キュー・マネージャー機能			
キュー・マネージャーの変更		ALTER QMGR	Yes
キュー・マネージャーの作成	<a href="#">crtmqm</a>		Yes
キュー・マネージャーの削除	<a href="#">dlmqm</a>		Yes
キュー・マネージャーの表示		DISPLAY QMGR	Yes
キュー・マネージャーの終了	<a href="#">endmqm</a>		Yes
キュー・マネージャーの ping		PING QMGR	No
キュー・マネージャーの始動	<a href="#">strmqm</a>		Yes
コマンド・サーバー機能			
コマンド・サーバーの表示	<a href="#">dspmqcsv</a>		No
コマンド・サーバーの終了	<a href="#">endmqcsv</a>		No
コマンド・サーバーの開始	<a href="#">strmqcsv</a>		No
キュー機能			
キューの変更		ALTER QALIAS ALTER QLOCAL ALTER QMODEL ALTER QREMOTE  ALTER (キュー) を参照してください。	Yes
キューのクリア		CLEAR QLOCAL	Yes
キューの作成		DEFINE QALIAS DEFINE QLOCAL DEFINE QMODEL DEFINE QREMOTE  DEFINE (キュー) を参照してください。	Yes
キューの削除		DELETE QALIAS DELETE QLOCAL DELETE QMODEL DELETE QREMOTE  DELETE (キュー) を参照してください。	Yes
キューの表示		DISPLAY QUEUE	Yes
プロセス機能			
プロセスの変更		ALTER PROCESS	Yes

表 19. UNIX, Linux, and Windows システムで必要な機能 (続き)			
関数	制御コマンド	MQSC	IBM MQ エクスプローラーで同等のコマンドがあるか
プロセスの作成		<a href="#">DEFINE PROCESS</a>	Yes
プロセスの削除		<a href="#">DELETE PROCESS</a>	Yes
プロセスの表示		<a href="#">DISPLAY PROCESS</a>	Yes
チャンネル機能			
チャンネルの変更		<a href="#">ALTER CHANNEL</a>	Yes
チャンネルの作成		<a href="#">DEFINE CHANNEL</a>	Yes
チャンネルの削除		<a href="#">DELETE CHANNEL</a>	Yes
チャンネルの表示		<a href="#">DISPLAY CHANNEL</a>	Yes
チャンネル状況の表示		<a href="#">DISPLAY CHSTATUS</a>	Yes
チャンネルの終了		<a href="#">STOP CHANNEL</a>	Yes
チャンネルの ping		<a href="#">PING CHANNEL</a>	Yes
チャンネルのリセット		<a href="#">RESET CHANNEL</a>	Yes
チャンネルの解決		<a href="#">RESOLVE CHANNEL</a>	Yes
チャンネルの実行	<a href="#">runmqchl</a>	<a href="#">START CHANNEL</a>	Yes
チャンネル・イニシエーターの実行	<a href="#">runmqchi</a>	<a href="#">START CHINIT</a>	No
リスナー <sup>1</sup> の実行	<a href="#">runmqslr</a>	<a href="#">START LISTENER</a>	No
リスナーの終了	<a href="#">endmqslr</a> (Windows システム、AIX、HP-UX、および Solaris のみ)		No

注:

1. リスナーは、キュー・マネージャーが開始されると自動的に開始されます。

## **ULW** オブジェクトの準備作業

チャンネルを開始する前に、チャンネルが定義されていて、それに関連するオブジェクトが存在し、使用可能になっていなければなりません。このセクションでは、その方法を示します。

IBM MQ コマンド (MQSC) または IBM MQ Explorer を使用して、以下のことを行います。

1. メッセージ・チャンネルおよび関連オブジェクトの定義
2. メッセージ・チャンネルのモニターと制御

定義が必要な関連オブジェクトは次のとおりです。

- 伝送キュー
- リモート・キュー定義
- キュー・マネージャー別名定義
- 応答先キュー別名の定義
- 応答先ローカル・キュー
- トリガー操作のプロセス (MCA)

## • メッセージ・チャンネル定義

各チャンネルに関する特定の通信リンクが定義されて、使用可能になっていなければ、チャンネルを実行することはできません。LU 6.2、TCP/IP、NetBIOS、SPX、および DECnet リンクの定義方法については、ご使用のシステムに関する特定の通信の手引きを参照してください。構成情報の例も参照してください。

オブジェクトの作成および処理についての詳細は、以下のサブトピックを参照してください。

### ULW 関連オブジェクトの作成

MQSC は関連オブジェクトを作成するのに使用されます。

MQSC を使用して、次のようなキューおよび別名オブジェクトを作成します。伝送キュー、リモート・キュー定義、キュー・マネージャー別名定義、応答先キュー別名定義、および応答先ローカル・キュー。

トリガー操作のプロセス (MCA) の定義も同様に作成する必要があります。

必要なすべてのオブジェクトを作成する方法を示す例については、[UNIX, Linux, and Windows におけるメッセージ・チャンネルの計画例](#)を参照してください。

### ULW デフォルト・オブジェクトの作成

デフォルト・オブジェクトは、キュー・マネージャーが作成されると自動的に作成されます。これらのオブジェクトとは、キュー、チャンネル、プロセス定義、管理キューです。デフォルト・オブジェクトが作成された後、`-c` オプションで `strmqm` コマンドを実行することによっていつでもそれらを置換できます。

`crtmqm` コマンドを使用して、キュー・マネージャーを作成すると、コマンドは、プログラムを開始して、一連のデフォルト・オブジェクトも作成します。

1. 次に、各デフォルト・オブジェクトが作成されます。プログラムは、正しく定義されたオブジェクト、存在していて置換されたオブジェクト、および試みが失敗したオブジェクトのカウントを保持します。
2. プログラムは作成者に結果を表示し、エラーが発生すると、詳細については該当するエラー・ログを参照するように指示します。

プログラムが実行を終了したら、`strmqm` コマンドを使用して、キュー・マネージャーを開始します。

`crtmqm` および `strmqm` コマンドについて詳しくは、[IBM MQ 制御コマンド・リファレンス](#)を参照してください。

## デフォルト・オブジェクトの変更

`-c` オプションを指定すると、オブジェクトが作成されているとき、キュー・マネージャーが一時的に開始されて、その後、再度終了します。`strmqm` に `-c` オプションを付けて発行すると、既存のシステム・オブジェクトがデフォルト値でリフレッシュされます (例えば、チャンネル定義の `MCAUSER` 属性はブランクに設定されます)。キュー・マネージャーを開始したい場合は、`-c` オプションなしで、再度 `strmqm` コマンドを使用する必要があります。

デフォルト・オブジェクトを変更する場合は、旧 `amqscoma.tst` ファイルの自分用のバージョンを作成して、それを編集できます。

### ULW チャンネルの作成

2つのチャンネル定義、つまり接続の両端に1つずつの定義を作成します。第1のキュー・マネージャーで最初のチャンネルを作成します。次に、リンクのもう一方の第2のキュー・マネージャーで2番目のチャンネル定義を作成します。

両端とも同じチャンネル名を使用して定義する必要があります。チャンネルの両端は、送信側と受信側のように、互換性を備えたチャンネル・タイプでなければなりません。

リンクの一端でチャンネル定義を作成するためには、MQSC コマンド `DEFINE CHANNEL` を使用します。これには、チャンネル名、該当接続側のチャンネル・タイプ、接続名、記述 (必要な場合)、伝送キューの名前 (必要な場合)、および伝送プロトコルを指定します。また、すでに収集した情報を使用して、必要なチャンネル・タイプについて、システムのデフォルト値とは異なる値にしたい属性を指定します。

[チャンネル属性](#)で説明されているチャンネル属性の値を決めるためのヘルプ情報が用意されています。

注: 使用するネットワーク内のすべてのチャンネルに固有の名前を付けることをお勧めします。発信元および宛先のキュー・マネージャーの名前をチャンネル名に入れるのもこのための1つの方法です。

## チャンネル作成の例

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) +  
DESCR('Sender channel to QM2') +  
CONNNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

MQSC のすべての例で、このコマンドがコマンドのファイル内にあるように、また UNIX, Linux, and Windows で入力されているように表示されます。2つの方法は同じに思われますが、コマンドを対話形式で入力するにはまず、MQSC セッションを開始しておかなければならない点が異なります。runmqsc(デフォルト・キュー・マネージャーの場合)またはrunmqsc qmname (qmname は必要なキュー・マネージャーの名前)を入力します。その後で、例に示されているように任意の数のコマンドを入力します。

移植性を考慮して、コマンド行の長さは、最高 72 文字に限定されています。複数行にわたる場合は、上で示したとおり、連結文字+を使用します。

- **Windows** Windows では、Ctrl-z を使用してコマンド・ラインの入力を終了します。
- **Linux** **UNIX** UNIX and Linux では、Ctrl-d を使用します。
- また、UNIX, Linux, and Windows で、**end** コマンドを使用します。

### **ULW** チャンネルの表示

MQSC コマンド DISPLAY CHANNEL は、チャンネルの属性を表示するために使用されます。

特定の属性が要求されておらず、指定されているチャンネル名が汎用でない場合、DISPLAY CHANNEL コマンドのすべてのパラメーターがデフォルトで使用されます。

属性については [チャンネル属性](#) を参照してください。

## チャンネル表示の例

```
DISPLAY CHANNEL(QM1.TO.QM2) TRPTYPE,CONVERT  
DISPLAY CHANNEL(QM1.TO.*) TRPTYPE,CONVERT  
DISPLAY CHANNEL(*) TRPTYPE,CONVERT  
DISPLAY CHANNEL(QM1.TO.QMR34) ALL
```

### **ULW** チャンネル状況の表示

MQSC コマンド DISPLAY CHSTATUS は、チャンネル名、およびチャンネルの現在状況または保管済み情報の状況を入手したいかどうかを指定して使用します。

DISPLAY CHSTATUS はすべてのメッセージ・チャンネルに適用されます。サーバー接続チャンネル以外の MQI チャンネルには適用されません。

表示される情報は以下のとおりです。

- チャンネル名
- 通信接続名
- チャンネルの未確定状況 (該当する場合)
- 最終シーケンス番号
- 伝送キュー名 (該当する場合)
- 未確定 ID (該当する場合)
- 最後にコミットされたシーケンス番号
- 論理作業単位 ID

- プロセス ID
- **Windows** スレッド ID (Windows のみ)

## チャンネル状況の表示例

```
DISPLAY CHSTATUS(*) CURRENT
DISPLAY CHSTATUS(QM1.TO.*) SAVED
```

保存された状況は、メッセージの少なくとも 1 つのバッチがチャンネルに伝送されるまでは適用されません。状況は、チャンネルが (STOP CHL コマンドで) 停止した場合、およびキュー・マネージャーが終了した場合にも保管されます。

### **ULW** Ping を使用したリンクの検査

MQSC コマンド PING CHANNEL は、固定データ・メッセージをリモート側と交換するために使用します。

Ping により、システム監視プログラムは、リンクが使用可能であって機能していることをある程度確認することができます。

Ping を選択しても、伝送キューとターゲット・キューが使用されることはありません。チャンネル定義、関連する通信リンク、およびネットワーク設定が使用されます。それは、現在チャンネルがアクティブでない場合にのみ使用できます。

Ping は、送信側チャンネル、サーバー・チャンネル、クラスター送信側チャンネルからのみ使用できます。リンクの反対側で対応チャンネルが開始され、開始パラメーターの折衝が実施されます。エラーがある場合には、通常どおり通知されます。

メッセージ交換の結果は、Ping complete またはエラー・メッセージで示されます。

## LU 6.2 での ping

ping が呼び出される時、デフォルトでは、受信側には、ユーザー ID またはパスワードは送られません。ユーザー ID やパスワードが必要な場合は、開始側のチャンネル定義で作成することができます。チャンネル定義にパスワードを入れると、それは、保管される前に IBM MQ によって暗号化されます。その後、暗号解読されて、会話を介して流されます。

### **ULW** チャンネルの開始

MQSC コマンド START CHANNEL は、送信側、サーバー、および要求側のチャンネルに対して使用します。アプリケーションがメッセージ交換を行えるようにするには、インバウンド接続に対してリスナー・プログラムを開始する必要があります。

START CHANNEL は、チャンネルがキュー・マネージャーによるトリガー操作で設定されている場合には必要ありません。

送信側 MCA は、開始されると、チャンネル定義を読み取り、伝送キューをオープンします。チャンネル開始シーケンスが実行され、リモートで受信側またはサーバー側のチャンネルの対応する MCA が開始されます。これらのチャンネルが開始すると、送信側プロセスおよびサーバー・プロセスはメッセージが伝送キューに到達するのを待ち、到達したメッセージを伝送します。

トリガー操作を使用するか、チャンネルをスレッドとして実行する場合には、チャンネル・イニシエーターが開始キューをモニターできるようにしておいてください。チャンネル・イニシエーターは、キュー・マネージャーの一部としてデフォルトで始動します。

ただし、TCP および LU 6.2 は、その他に以下の機能を提供します。

- **Linux** **UNIX** UNIX and Linux 上の TCP では、inetd を構成してチャンネルを開始できます。inetd は、独立したプロセスとして開始されます。

- Linux
UNIX
 UNIX and Linux における LU 6.2 の場合、SNA 製品を構成して LU 6.2 応答側プロセスを開始します。
- Windows
 Windows での LU 6.2 では、SNA サーバーを使用すると、TpStart (SNA サーバーに付属のユーティリティ) を使用してチャンネルを開始できます。TpStart は、独立したプロセスとして開始されません。

「Start (開始)」オプションを使用すると、必要な場合には必ずチャンネルが再同期化されます。

開始手続きを正常に行うためには、次の条件が必要です。

- ローカルおよびリモートのチャンネル定義が存在している。受信側またはサーバー接続のチャンネルに該当するチャンネル定義がない場合は、チャンネルが自動定義されていると、自動的にデフォルトのチャンネル定義が作成されます。チャンネル自動定義出口プログラムを参照してください。
- 伝送キューが存在していて、しかも他のチャンネルで使用されていない。
- ローカルおよびリモートの MCA が存在している。
- 通信リンクが利用可能である。
- ローカルおよびリモートのキュー・マネージャーが実行されている。
- メッセージ・チャンネルがまだ実行されていない状態である。

チャンネル開始要求が受け入れられたことを確認させるメッセージが画面に戻されます。開始コマンドが正常に行われたかどうかを確認するためには、エラー・ログを調べるか、DISPLAY CHSTATUS を使用してください。エラー・ログは次のとおりです。

#### Windows Windows

`MQ_DATA_PATH\qmgrs\qmname\errors\AMQERR01.LOG` (qmname というキュー・マネージャーごとに)

`MQ_DATA_PATH\qmgrs\@SYSTEM\errors\AMQERR01.LOG` (一般エラーの場合)

`MQ_DATA_PATH` は、IBM MQ がインストールされている上位ディレクトリーを表します。

注: Windows では、Windows システム・アプリケーション・イベント・ログでメッセージを取得することもできます。

#### Linux UNIX UNIX and Linux

`/var/mqm/qmgrs/qmname/errors/AMQERR01.LOG` (qmname というキュー・マネージャーごとに)

`/var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG` (一般エラーの場合)

UNIX, Linux, and Windows では、**runmqlsr** コマンドを使用して IBM MQ リスナー・プロセスを開始します。デフォルトでは、チャンネル接続インバウンド要求によってリスナー・プロセスが MCA を amqrmppa プロセスのスレッドとして開始します。

```
runmqlsr -t tcp -m QM2
```

アウトバウンド接続については、次の 3 つの方法のどれか 1 つでチャンネルを開始する必要があります。

- MCATYPE パラメーターに基づいて、プロセスまたはスレッドとしてチャンネルを開始させるには、チャンネル名を指定して、MQSC コマンド START CHANNEL を使用します。(チャンネルがスレッドとして開始されるときには、チャンネルがチャンネル・イニシエーターのスレッドです。)

```
START CHANNEL(QM1.TO.QM2)
```

- 制御コマンド runmqchl を使用して、プロセスとしてチャンネルを開始させます。

```
runmqchl -c QM1.TO.QM2 -m QM1
```

- チャンネル・イニシエーターを使用して、チャンネルを起動します。

## ULW チャンネルの停止

MQSC コマンド STOP CHANNEL は、チャンネルに活動の停止を要求するために使用します。オペレーターがチャンネルを再始動させるまで、チャンネルは新しいメッセージのバッチを開始しません。

停止したチャンネルの再始動に関する情報については、[180 ページの『停止したチャンネルの再始動』](#)を参照してください。

このコマンドは、MQCHT\_CLNTCONN 以外のどのタイプのチャンネルに対しても発行できます。

必要な停止のタイプを選択できます。

### 静止の停止例

```
STOP CHANNEL(QM1.TO.QM2) MODE(QUIESCE)
```

このコマンドは、通常の手順でチャンネルがクローズされることを要求します。現行のメッセージ・バッチは完了され、チャンネルの他方の側で同期点手順が行われます。チャンネルがアイドル状態であると、このコマンドは受信側チャンネルを終了させません。

### 強制停止の例

```
STOP CHANNEL(QM1.TO.QM2) MODE(FORCE)
```

このオプションは、チャンネルを即時に停止しますが、チャンネルのスレッドやプロセスを終了しません。チャンネルは、メッセージの現行バッチの処理を完了しないため、そのチャンネルが未確定のままになります。一般的には、静止の停止オプションの使用を考慮します。

### 終了停止の例

```
STOP CHANNEL(QM1.TO.QM2) MODE(TERMINATE)
```

このオプションは、チャンネルを即時に停止し、チャンネルのスレッドやプロセスを終了します。

### 停止 (静止) 停止の例

```
STOP CHANNEL(QM1.TO.QM2) STATUS(STOPPED)
```

このコマンドでは MODE を指定しないため、デフォルトで MODE(QUIESCE) になります。このコマンドはチャンネルの停止を要求するので、チャンネルの自動的な再始動は行えません。したがって、手動で開始する必要があります。

### 停止 (静止) の非アクティブ状態の例

```
STOP CHANNEL(QM1.TO.QM2) STATUS(INACTIVE)
```

このコマンドでは MODE を指定しないため、デフォルトで MODE(QUIESCE) になります。このコマンドはチャンネルの非アクティブ化を要求するので、必要に応じて、チャンネルは自動的に再始動されます。

## ULW チャンネルの名前変更

MQSC を使用して、メッセージ・チャンネルを名前変更します。

MQSC を使用して、以下のステップを行ってください。

1. STOP CHANNEL を使用してチャンネルを停止させる。
2. DEFINE CHANNEL を使用して、新しい名前で重複したチャンネル定義を作成する。

3. DISPLAY CHANNEL を使用して正常に作成されたかを確認する。

4. DELETE CHANNEL を使用して元のチャンネル定義を削除する。

メッセージ・チャンネルの名前を変更することを決めた場合は、チャンネルは両側に1つずつ、計2個のチャンネル定義をもっていることに注意してください。必ず両側のチャンネルを同時に名前変更するようにします。

## **ULW** チャンネルのリセット

MQSC コマンド RESET CHANNEL は、メッセージ・シーケンス番号を変更するために使用します。

RESET CHANNEL コマンドはどのメッセージ・チャンネルに対しても使用できますが、MQI チャンネル (クライアント接続またはサーバー接続) には使用できません。次にそのチャンネルが開始されると、最初のメッセージは新しいシーケンス番号になります。

送信側またはサーバーのチャンネルでコマンドが出されると、他方のチャンネルが再始動されたときに、変更が通知されます。

### 関連概念

[194 ページの『オブジェクトの準備作業』](#)

チャンネルを開始する前に、チャンネルが定義されていて、それに関連するオブジェクトが存在し、使用可能になっていなければなりません。このセクションでは、その方法を示します。

[169 ページの『チャンネル制御機能』](#)

チャンネル制御機能は、チャンネルを定義、モニター、および制御するための機能を備えています。

### 関連タスク

[140 ページの『分散キューイングの構成』](#)

このセクションでは、キュー定義、チャンネル定義、トリガー操作および同期点プロシージャを含む、IBM MQ インストール間の相互通信の詳細について説明します。

### 関連情報

RESET CHANNEL

## **ULW** チャンネルの未確定メッセージの解決

MQSC コマンド RESOLVE CHANNEL は、送信側またはサーバーによってメッセージが未確定のままにされているときに使用します。例えば、リンクの反対側が終了していて、メッセージが戻される見込みがないなどの理由によります。

RESOLVE CHANNEL コマンドは、2つのパラメーター、BACKOUT または COMMIT のいずれかを受信します。バックアウトではメッセージは伝送キューに復元され、コミットではメッセージは廃棄されます。

チャンネル・プログラムは、パートナーとのセッションの確立を試みません。その代わりに、未確定メッセージを表す作業論理単位 ID (LUWID) を判別します。そのあとで、要求に応じて次のいずれかを発行します。

- メッセージを伝送キューに復元するための BACKOUT
- メッセージを伝送キューから削除するための COMMIT

解決を正常に行うためには、次の条件が必要です。

- チャンネルが非アクティブになっている。
- チャンネルが未確定になっている。
- チャンネル・タイプが送信側、サーバー、またはクラスター送信側である。
- ローカルのチャンネル定義が存在している。
- ローカル・キュー・マネージャーが実行している。

### 関連概念

[194 ページの『オブジェクトの準備作業』](#)

チャンネルを開始する前に、チャンネルが定義されていて、それに関連するオブジェクトが存在し、使用可能になっていなければなりません。このセクションでは、その方法を示します。

[169 ページの『チャンネル制御機能』](#)

チャンネル制御機能は、チャンネルを定義、モニター、および制御するための機能を備えています。

## 関連タスク

[140 ページの『分散キューイングの構成』](#)

このセクションでは、キュー定義、チャンネル定義、トリガー操作および同期点プロシージャを含む、IBM MQ インストール間の相互通信の詳細について説明します。

## 関連情報

[RESOLVE CHANNEL](#)

## Windows Windows における通信のセットアップ

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。これを正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。このセクションでは、IBM MQ for Windows システムで使用可能な通信の形式を使用して、これを行う方法について説明しています。

## 始める前に

[構成例 - IBM MQ for Windows](#) を参照すると役に立つ場合があります。

## このタスクについて

Windows で IBM MQ の通信をセットアップする場合は、以下の通信タイプから選択できます。

- TCP/IP
- LU 6.2
- NetBIOS

## 手順

- Windows システムの通信のセットアップについては、選択した通信タイプのサブトピックを参照してください。
  - [202 ページの『Windows での TCP 接続の定義』](#)
  - [203 ページの『Windows での LU 6.2 接続の定義』](#)
  - [205 ページの『Windows での NetBIOS 接続の定義』](#)

## 関連タスク

[191 ページの『UNIX, Linux, and Windows 上のチャンネルのモニターおよび制御』](#)

DQM では、リモート・キュー・マネージャーに対するチャンネルの作成、モニターおよび制御が必要です。コマンド、プログラム、IBM MQ Explorer、チャンネル定義ファイル、および同期情報のストレージ域を使用して、チャンネルを制御できます。

[16 ページの『サーバーとクライアント間の接続の構成』](#)

IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャンネルを定義します。

[208 ページの『UNIX and Linux における通信のセットアップ』](#)

DQM は IBM MQ 用のリモート・キューイング機能です。これはキュー・マネージャーにチャンネル制御プログラムを提供し、このプログラムによって、通信リンクに対する、システム・オペレーターからの制御が可能なインターフェースが形成されます。分散キューイング管理によって維持されるチャンネル定義は、これらの接続を使用します。

## 関連資料

[17 ページの『使用する通信タイプ』](#)

サポートされる通信プロトコルはプラットフォームによって異なります。どの伝送プロトコルを選択するかは、IBM MQ MQI client とサーバー双方のプラットフォームの組み合わせ方によります。

## Windows Windows での TCP 接続の定義

送信側でチャンネルを構成してターゲットのアドレスを指定し、受信側でリスナー・プログラムを実行することにより、TCP 接続を定義します。

### 送信側

チャンネル定義の「Connection name (接続名)」フィールドにホスト名またはターゲット・マシンの TCP アドレスを指定します。

接続されるポートのデフォルト値は 1414 です。Internet Assigned Numbers Authority (インターネットによる番号割り当て権限) によってポート番号 1414 が IBM MQ に割り当てられます。

デフォルト以外のポート番号を使用する場合は、チャンネル・オブジェクト定義の接続名フィールドに次のようにポート番号を指定します。

```
DEFINE CHANNEL('channel name') CHLTYPE(SDR) +  
  TRPTYPE(TCP) +  
  CONNAME('OS2ROG3(1822)') +  
  XMITQ('XMITQ name') +  
  REPLACE
```

ここで、OS2ROG3 はリモート・キュー・マネージャーの DNS ホスト名、1822 は必要なポート番号です。(これは受信側のリスナーが待機するポートでなければなりません。)

チャンネル・オブジェクト定義の変更を反映させるには、実行中のチャンネルを停止して再始動する必要があります。

デフォルトのポート番号は、IBM MQ for Windows の .ini ファイルにポート番号を指定することで変更できます。

```
TCP:  
Port=1822
```

注: IBM MQ は、使いたい TCP/IP ポート番号を選択するときに、次のような順序で最初に見つけたポート番号を使います。

1. チャンネル定義またはコマンド行で明示的に指定されたポート番号。この番号を指定すると、チャンネルのデフォルト・ポート番号を変更することができます。
2. .ini ファイルの TCP スタンザに指定されたポート属性。この番号を指定すると、キュー・マネージャーのデフォルト・ポート番号を変更することができます。
3. デフォルト値 1414。これは、インバウンドとアウトバウンドの両方の接続用に Internet Assigned Numbers Authority が IBM MQ に割り当てた番号です。

qm.ini を使用して設定する値について詳しくは、[分散キューイング用の構成ファイル・スタンザを参照してください](#)。

### TCP での受信

受信側のチャンネル・プログラムを開始するには、着信ネットワーク要求を検出し、関連チャンネルを開始するリスナー・プログラムが起動されている必要があります。IBM MQ リスナーを使用することができます。

受信側のチャンネル・プログラムは、送信側チャンネルからの開始要求に応じて開始されます。

受信側のチャンネル・プログラムを開始するには、着信ネットワーク要求を検出し、関連チャンネルを開始するリスナー・プログラムが起動されている必要があります。IBM MQ リスナーを使用することができます。

IBM MQ が提供するリスナーは、新しいチャンネルをスレッドとして開始させますが、これを実行させるためには、`runmqslsr` コマンドを使用します。

**runmqtsr** コマンドを使用する基本的な例を次に示します。

```
runmqtsr -t tcp [-m QMNAME] [-p 1822]
```

大括弧は、オプション・パラメーターを示します。デフォルト・キュー・マネージャーの場合は QMNAME は必要ありません。また、デフォルト・ポート番号 (1414) を使用する場合はポート番号は必要ありません。ポート番号は 65535 以下でなければなりません。

**注:** IBM MQ は、使いたい TCP/IP ポート番号を選択するときに、次のような順序で最初に見つけたポート番号を使います。

1. チャンネル定義またはコマンド行で明示的に指定されたポート番号。この番号を指定すると、チャンネルのデフォルト・ポート番号を変更することができます。
2. .ini ファイルの TCP スタンザに指定されたポート属性。この番号を指定すると、キュー・マネージャーのデフォルト・ポート番号を変更することができます。
3. デフォルト値 1414。これは、インバウンドとアウトバウンドの両方の接続用に Internet Assigned Numbers Authority が IBM MQ に割り当てた番号です。

パフォーマンスを最高にするには、[190 ページの『承認されたアプリケーションとしてのチャンネルおよびリスナーの実行』](#)で説明されているように、承認されたアプリケーションとして IBM MQ リスナーを実行します。承認されたアプリケーションについては、[承認されたアプリケーションに関する制限](#)を参照してください。

## TCP/IP SO\_KEEPALIVE オプションの使用

Windows SO\_KEEPALIVE オプションを使用する場合、レジストリーに以下の項目を追加する必要があります。

```
TCP:
KeepAlive=yes
```

SO\_KEEPALIVE オプションの詳細については、[176 ページの『チャンネルの相手側がまだ使用可能であるかどうかの検査』](#)を参照してください。

Windows では、WindowsKeepAliveTime オプションの HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters ジストリー値によって、接続が検査される前に経過する間隔が制御されます。デフォルト値は 2 時間です。

## TCP リスナー・バックログ・オプションの使用

TCP では、サーバーとクライアントとの間で 3 ウェイ・ハンドシェイクが実行されない限り、接続は不完全として扱われます。これらの接続は、未解決接続要求と呼ばれます。最大値をこれらの未解決接続要求に設定し、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

詳細については、[211 ページの『UNIX and Linux での TCP リスナー・バックログ・オプションの使用』](#)を参照してください。また、Windows の特定の値を参照してください。

### **Windows** Windows での LU 6.2 接続の定義

2 つのマシン間で LU 6.2 の会話が確立できるように SNA を構成する必要があります。

SNA が構成された後、次のように進んでください。

詳細は以下の表を参照してください。

表 20. ローカル Windows システムにおけるリモート・キュー・マネージャー・プラットフォーム用の設定値

リモート・プラットフォーム	TPNAME	TPPATH
z/OS または MVS™/ESA (CICS を使用しない)	リモート・キュー・マネージャーの対応するサイド情報内と同じ。	-
z/OS または MVS/ESA (CICS を使用する)	CKRC (送信側) CKSV (要求側) CKRC (サーバー)	-
IBM i	IBM i システムにおいて経路指定項目の比較値と同じ。	-
UNIX and Linux システム	リモート・キュー・マネージャーの対応するサイド情報内と同じ。	MQ_INSTALLATION_PATH/bin/amqcrs6a
Windows	Windows Run Listener コマンドで指定されたとおり、または Windows において TpSetup を使用して定義された呼び出し可能なトランザクション・プログラム。	MQ_INSTALLATION_PATH\bin\amqcrs6a

MQ\_INSTALLATION\_PATH は、IBM MQ がインストールされている上位ディレクトリーを表します。

同一のマシン上に複数のキュー・マネージャーがある場合は、チャンネル定義内の TPname をそれぞれ固有にする必要があります。

AnyNet® SNA over TCP/IP の構成に関する最新情報については、オンラインの IBM 資料である [AnyNet SNA over TCP/IP](#) および [SNA ノード操作](#) を参照してください。

## 関連概念

### 204 ページの『Windows における LU 6.2 での送信側』

使用している LU 6.2 製品の管理アプリケーションから CPI-C 側オブジェクト (シンボリック宛先) を作成します。その名前をチャンネル定義の「接続名」フィールドに入れます。また、パートナーに対する LU 6.2 リンクも作成します。

### 204 ページの『Windows における LU 6.2 での受信』

受信側のチャンネル・プログラムは、送信側チャンネルからの開始要求に応じて開始されます。

#### Windows Windows における LU 6.2 での送信側

使用している LU 6.2 製品の管理アプリケーションから CPI-C 側オブジェクト (シンボリック宛先) を作成します。その名前をチャンネル定義の「接続名」フィールドに入れます。また、パートナーに対する LU 6.2 リンクも作成します。

CPI-C サイド・オブジェクトには、受信側マシンのパートナー LU 名、TP 名、およびモード名を入れます。以下に例を示します。

```
Partner LU Name      OS2R0G2
Partner TP Name     recv
Mode Name           #INTER
```

#### Windows Windows における LU 6.2 での受信

受信側のチャンネル・プログラムは、送信側チャンネルからの開始要求に応じて開始されます。

受信側のチャンネル・プログラムを開始するには、着信ネットワーク要求を検出し、関連チャンネルを開始するリスナー・プログラムが起動されている必要があります。このリスナー・プログラムは、待機の対象となる TpName を指定して、RUNMQLSR コマンドを使用して開始させます。または、Windows では、SNA サーバー配下の TpStart を使用することもできます。

## RUNMQLSR コマンドの使用

次は、リスナーを開始するコマンドの一例です。

```
RUNMQLSR -t LU62 -n RECV [-m QMNAME]
```

ここで RECV は、相手側 (送信側) で「TpName to start on the remote side (リモート側で開始する際の TpName)」として指定されている TpName です。大括弧内の最後の部分はオプションで、デフォルト値のキュー・マネージャーに対しては必要ありません。

1つのマシンで、複数のキュー・マネージャーを実行することが可能です。どのキュー・マネージャーにも異なる TpName を割り当ててから、それぞれに対してリスナー・プログラムを開始する必要があります。以下に例を示します。

```
RUNMQLSR -t LU62 -m QM1 -n TpName1  
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

パフォーマンスを最高にするには、承認されたアプリケーションとしてのチャンネルおよびリスナーの実行で説明されているように、承認されたアプリケーションとして IBM MQ リスナーを実行します。承認されたアプリケーションについては、承認されたアプリケーションの制約事項を参照してください。

次のコマンドを使用して、非アクティブであるキュー・マネージャーで実行されているすべての IBM MQ リスナーを停止できます。

```
ENDMQLSR [-m QMNAME]
```

## Windows における Microsoft SNA Server の使用

TpSetup を (SNA サーバー SDK から) 使用して呼び出し可能 TP を定義して、amqcrs6a.exe を取り出すか、または手動で各種の登録値を設定することができます。amqcrs6a.exe に渡す必要のあるパラメーターは、次のとおりです。

```
-m QM -n TpName
```

ここで、QM はキュー・マネージャー名であり、TpName は TP 名です。詳細については、「Microsoft SNA Server APPC Programmers Guide」または「Microsoft SNA Server CPI-C Programmers Guide」を参照してください。

キュー・マネージャー名を指定しないと、デフォルトのキュー・マネージャーが想定されます。

### Windows Windows での NetBIOS 接続の定義

NetBIOS 接続は、Windows 稼働中のクライアントおよびサーバーにのみ適用されます。IBM MQ は、他の IBM MQ 製品との NetBIOS 接続を確立するとき、3つのタイプの NetBIOS リソース (セッション、コマンド、名前) を使用します。これらの各リソースには制限があり、それはデフォルトで、または NetBIOS のインストール中に選択によって設定されます。

タイプに関係なく、実行中のチャンネルは、1つの NetBIOS セッションおよび1つの NetBIOS コマンドを使用します。IBM NetBIOS 環境では、複数のプロセスで同一のローカル NetBIOS 名を使用できます。したがって、IBM MQ で使用する場合は、1つの NetBIOS 名のみが使用可能になっている必要があります。例えば、Novell の NetBIOS エミュレーションなど、他のベンダー環境では、プロセスごとに異なるローカル名が要求されます。使用している NetBIOS 製品の資料で、必要な条件を確認してください。

必ず各タイプの十分なリソースがすでに使用可能になっていることを確認してください。あるいは、構成情報として指定されている最大値を大きくします。値を変更した場合は常に、システムを再開する必要があります。

システム再開中、NetBIOS 装置ドライバーは、アプリケーションで使用するとき使用可能なセッション、コマンド、および名前の数を表示します。これらのリソースは、同一システムで実行しているすべての NetBIOS ベース・アプリケーションに使用可能です。したがって、IBM MQ がそれらのリソースの獲得を

必要とする前に、別のアプリケーションがそれらを消費してしまう可能性があります。この問題への対処法については、LAN ネットワーク管理者に相談してください。

## 関連概念

### 206 ページの『IBM MQ ローカル NetBIOS 名の定義』

IBM MQ チャンネル・プロセスで使用されるローカル NetBIOS 名は、3つの方法で指定できます。

### 207 ページの『キュー・マネージャーの NetBIOS セッション、コマンド、および名前制限の設定』

NetBIOS セッション、コマンド、および名前に関するキュー・マネージャー制限は、2つの方法で指定できます。

### 207 ページの『LAN アダプター番号の設定』

チャンネルが NetBIOS 間で正常に動作するためには、それぞれの側のアダプター・サポートに互換性がなければなりません。IBM MQ では、qm.ini ファイルの NETBIOS スタンザの中で AdapterNum 値を使用し、runmqclsr コマンドに **-a** パラメーターを指定することによって、LAN アダプター (LANA) 番号の選択を制御できます。

### 207 ページの『NetBIOS 接続の開始』

接続を開始するのに必要なステップを定義します。

### 208 ページの『NetBIOS 接続のターゲット・リスナーの定義』

NetBIOS 接続の受信側で行われるステップを定義します。

## Windows IBM MQ ローカル NetBIOS 名の定義

IBM MQ チャンネル・プロセスで使用されるローカル NetBIOS 名は、3つの方法で指定できます。

3つの方法を優先順位に従って示してあります。

1. 例えば、次のような RUNMQLSR コマンドの **-l** パラメーターに指定された値があります。

```
RUNMQLSR -t NETBIOS -l my_station
```

2. 次のコマンドで設定される値を持つ MQNAME 環境変数。

```
SET MQNAME= my_station
```

各プロセスごとに MQNAME 値を設定することができます。または、システム・レベルで、Windows レジストリー内に設定することもできます。

固有の名前が必要な NetBIOS 環境を使用している場合は、IBM MQ プロセスが開始される各ウィンドウで、SET MQNAME コマンドを入力する必要があります。MQNAME 値は任意ですが、各プロセスで固有でなければなりません。

3. キュー・マネージャー構成ファイル qm.ini 中の NETBIOS スタンザ。以下に例を示します。

```
NETBIOS:  
LocalName= my_station
```

## 注:

1. サポートされる NetBIOS 製品の環境のバリエーションに応じて、ネットワークでは、各 NetBIOS 名を固有にしておくことが重要です。名前を固有にしておかないと、予測不能な結果が生じることがあります。NetBIOS チャンネルの確立で問題が生じ、キュー・マネージャー・エラー・ログに NetBIOS 戻りコード X'15' を示すエラー・メッセージがある場合は、NetBIOS 名の使用を検討してください。
2. Windows では、マシン名を NetBIOS 名として使用することはできません。それは、Windows が既にその名前を使用しているためです。
3. 送信側のチャンネルを開始するには、NetBIOS 名が、qm.ini ファイルの中で、MQNAME 環境変数または LocalName を使用して指定されていなければなりません。

## Windows キュー・マネージャーの NetBIOS セッション、コマンド、および名前制限の設定

NetBIOS セッション、コマンド、および名前に関するキュー・マネージャー制限は、2つの方法で指定できます。

これらの方法を優先順位に従って示してあります。

1. RUNMQLSR コマンドで指定された値。

```
-s Sessions  
-e Names  
-o Commands
```

コマンドに `-m` オペランドが指定されていない場合、値はデフォルトのキュー・マネージャーにのみ適用されます。

2. キュー・マネージャー構成ファイル `qm.ini` 中の NETBIOS スタンザ。以下に例を示します。

```
NETBIOS:  
  
NumSess= Qmgr_max_sess  
NumCmds= Qmgr_max_cmds  
NumNames= Qmgr_max_names
```

## Windows LAN アダプター番号の設定

チャンネルが NetBIOS 間で正常に動作するためには、それぞれの側のアダプター・サポートに互換性がなければなりません。IBM MQ では、`qm.ini` ファイルの NETBIOS スタンザの中で `AdapterNum` 値を使用し、`runmqslr` コマンドに `-a` パラメーターを指定することによって、LAN アダプター (LANA) 番号の選択を制御できます。

IBM MQ によって、NetBIOS 接続に使用されるデフォルトの LAN アダプター番号は 0 です。次のように、システムで使用されている番号を確認します。

Windows 上では、オペレーティング・システムから直接 LAN アダプター番号を照会することはできません。代わりに、Microsoft から入手可能な `LANACFG.EXE` コマンド行ユーティリティを使用します。ツールの出力は、仮想 LAN アダプター番号と、その有効なバインディングを示します。LAN アダプター番号の詳細については、Microsoft サポート技術情報の記事 138037 『HOWTO: Use LANA Numbers in a 32-bit Environment』を参照してください。

次のように、キュー・マネージャー構成ファイル `qm.ini` の NETBIOS スタンザに正しい値を指定します。

```
NETBIOS:  
AdapterNum= n
```

ここで、`n` は、このシステムの正しい LAN アダプター番号です。

## Windows NetBIOS 接続の開始

接続を開始するのに必要なステップを定義します。

接続を開始するには、送信側で次のステップに従ってください。

1. `MQNAME` または `LocalName` 値を使用して、NetBIOS ステーション名を定義します。
2. システムで使用している LAN アダプター番号を確認し、`AdapterNum` を使用して正しいファイルを指定します。
3. チャンネル定義の「`ConnectionName`」フィールドで、ターゲット・リスナー・プログラムが使用する NetBIOS 名を指定します。Windows では、NetBIOS チャンネルをスレッドとして実行しなければなりません。チャンネル定義で `MCATYPE (THREAD)` を指定して、これを実行してください。

```
DEFINE CHANNEL (chname) CHLTYPE(SDR) +  
TRPTYPE(NETBIOS) +  
CONNAME(your_station) +  
XMITQ(xmitq) +
```

```
MCATYPE(THREAD) +  
REPLACE
```

## Windows NetBIOS 接続のターゲット・リスナーの定義

NetBIOS 接続の受信側で行われるステップを定義します。

受信側では、次のステップに従ってください。

1. MQNAME または LocalName 値を使用して、NetBIOS ステーション名を定義します。
2. システムで使用している LAN アダプター番号を確認し、AdapterNum を使用して正しいファイルを指定します。
3. 次のように受信側チャンネルを定義します。

```
DEFINE CHANNEL (chname) CHLTYPE(RCVR) +  
TRPTYPE(NETBIOS) +  
REPLACE
```

4. IBM MQ リスナー・プログラムを開始してステーションを設定し、連絡を受けられる状態にします。以下に例を示します。

```
RUNMQLSR -t NETBIOS -l your_station [-m qmgr]
```

このコマンドは、`your_station` を連絡待ちの NetBIOS ステーションとして設定します。NetBIOS ステーション名は、NetBIOS ネットワークで固有のものでなければなりません。

パフォーマンスを最高にするには、190 ページの『承認されたアプリケーションとしてのチャンネルおよびリスナーの実行』で説明されているように、承認されたアプリケーションとして IBM MQ リスナーを実行します。承認されたアプリケーションについては、承認されたアプリケーションの制約事項を参照してください。

次のコマンドを使用して、非アクティブであるキュー・マネージャーで実行されているすべての IBM MQ リスナーを停止できます。

```
ENDMQLSR [-m QMNAME]
```

キュー・マネージャー名を指定しないと、デフォルトのキュー・マネージャーが想定されます。

## Linux UNIX UNIX and Linux における通信のセットアップ

DQM は IBM MQ 用のリモート・キューイング機能です。これはキュー・マネージャーにチャンネル制御プログラムを提供し、このプログラムによって、通信リンクに対する、システム・オペレーターからの制御が可能なインターフェースが形成されます。分散キューイング管理によって維持されるチャンネル定義は、これらの接続を使用します。

### 始める前に

次のセクションを参照すると役立つ場合があります。

- ▶ **AIX** [構成例 - IBM MQ for AIX](#)
- ▶ **HP-UX** [構成例 - IBM MQ for HP-UX](#)
- ▶ **Solaris** [構成例 - IBM MQ for Solaris](#)
- ▶ **Linux** [構成例 - IBM MQ for Linux](#)

## このタスクについて

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。このセクションでは、そのための方法を説明します。

UNIX and Linux 上の IBM MQ の通信をセットアップする場合、以下のタイプの通信から選択できます。

- TCP/IP
- LU 6.2

どのチャンネル定義も伝送プロトコル (Transport Type) 属性を 1 つだけ指定する必要があります。1 つのキュー・マネージャーが 1 つまたは複数のプロトコルを使用することができます。

IBM MQ MQI clients の場合、別の伝送プロトコルを使用する代替チャンネルがあると役に立つことがあります。IBM MQ MQI clients について詳しくは、[IBM MQ MQI clients の概要](#)を参照してください。

## 手順

UNIX and Linux システムの通信のセットアップについては、選択した通信タイプのサブトピックを参照してください。

- [209 ページの『UNIX and Linux での TCP 接続の定義』](#)
- [213 ページの『UNIX and Linux での LU 6.2 接続の定義』](#)

## 関連タスク

[191 ページの『UNIX, Linux, and Windows 上のチャンネルのモニターおよび制御』](#)

DQM では、リモート・キュー・マネージャーに対するチャンネルの作成、モニターおよび制御が必要です。コマンド、プログラム、IBM MQ Explorer、チャンネル定義ファイル、および同期情報のストレージ域を使用して、チャンネルを制御できます。

[16 ページの『サーバーとクライアント間の接続の構成』](#)

IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャンネルを定義します。

[201 ページの『Windows における通信のセットアップ』](#)

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。これを正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。このセクションでは、IBM MQ for Windows システムで使用可能な通信の形式を使用して、これを行う方法について説明しています。

## 関連資料

[17 ページの『使用する通信タイプ』](#)

サポートされる通信プロトコルはプラットフォームによって異なります。どの伝送プロトコルを選択するかは、IBM MQ MQI client とサーバー双方のプラットフォームの組み合わせ方によります。

Linux

UNIX

## UNIX and Linux での TCP 接続の定義

送信側のチャンネル定義では、宛先のアドレスを指定します。受信側の接続については、リスナーまたは inet デーモンが構成されます。

## 送信側

チャンネル定義の「Connection Name (接続名)」フィールドにホスト名またはターゲット・マシンの TCP アドレスを指定します。接続されるポートのデフォルト値は 1414 です。Internet Assigned Numbers Authority (インターネットによる番号割り当て権限) によってポート番号 1414 が IBM MQ に割り当てられます。

デフォルト以外のポート番号を使用するには、接続名フィールドを次のように変更します。

```
Connection Name REMHOST(1822)
```

ここで、REMHOST はリモート・マシンのホスト名、1822 は必要なポート番号です。(これは受信側のリスナーが待機するポートでなければなりません。)

または、次のようにキュー・マネージャーの構成ファイル (qm.ini) にポート番号を指定して、変更することもできます。

```
TCP:  
Port=1822
```

qm.ini を使用して設定する値について詳しくは、[分散キューイング用の構成ファイル・スタンザを参照してください](#)。

## TCP での受信

inet デーモンである TCP/IP リスナー (inetd)、または IBM MQ リスナーのいずれかを使用することができます。

現在は、inet デーモンではなく、拡張 inet デーモン (xinetd) を使用する Linux 配布版もあります。Linux システムにおける拡張 inet デーモンの使用方法について詳しくは、[Linux での TCP 接続の確立](#)を参照してください。

### 関連概念

[210 ページの『UNIX and Linux での TCP/IP リスナーの使用』](#)

UNIX and Linux 上でチャンネルを開始するには、/etc/services ファイルと inetd.conf ファイルを編集しなければなりません。

[211 ページの『UNIX and Linux での TCP リスナー・バックログ・オプションの使用』](#)

TCP では、サーバーとクライアントとの間で 3 ウェイ・ハンドシェイクが実行されない限り、接続は不完全として扱われます。これらの接続は、未解決接続要求と呼ばれます。最大値をこれらの未解決接続要求に設定し、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

[212 ページの『IBM MQ リスナーの使用』](#)

IBM MQ に提供されているリスナーは、新しいチャンネルをスレッドとして起動します。このリスナーを実行するには、runmqslsr コマンドを使用します。

[213 ページの『TCP/IP SO\\_KEEPALIVE オプションの使用』](#)

一部の UNIX and Linux システムでは、接続がまだ利用できるかどうか検査する前の TCP の待機時間を指定できます。また、最初の検査が失敗した場合の接続の再試行回数も指定できます。それには、カーネル調整パラメーターを使うか、またはコマンド行で指定できます。

Linux

UNIX

UNIX and Linux での TCP/IP リスナーの使用

UNIX and Linux 上でチャンネルを開始するには、/etc/services ファイルと inetd.conf ファイルを編集しなければなりません。

次の指示に従ってください。

1. /etc/services ファイルを編集します。

**注:** /etc/services ファイルを編集するには、スーパーユーザーまたはルートとしてログインしなければなりません。このポート番号は変更できますが、送信側で指定したポート番号と一致しなければなりません。

次の行をファイルに追加します。

```
MQSeries 1414/tcp
```

ここで、1414 は IBM MQ に必要なポート番号です。ポート番号は 65535 以下でなければなりません。

2. `inetd.conf` ファイルに行を追加して、プログラム `amqcrsta` を呼び出します。ここで、`MQ_INSTALLATION_PATH` は、IBM MQ がインストールされている上位ディレクトリーを表します：

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta
[-m Queue_Man_Name]
```

これらの更新は、`inetd` が構成ファイルを再度読み取ったあとでアクティブになります。そのために、ルート・ユーザー ID で次のコマンドを出します。

- AIX の場合：

```
refresh -s inetd
```

- HP-UX の場合、`mqm` ユーザー ID から次のようにします。

```
inetd -c
```

- Solaris 10 以降の場合、次のようにします。

```
inetconv
```

- 他の UNIX and Linux システム (Solaris 9 を含む) の場合、次のようにします。

```
kill -1 process_number
```

`inetd` が開始したリスナー・プログラムが `inetd` からロケールを継承している場合は、MQMDE が使用 (マージ) されずに、メッセージ・データとしてキューに書き込まれる可能性があります。MQMDE を確実に使用するには、ロケールを正しく設定しなければなりません。`inetd` によって設定されるロケールは、IBM MQ プロセスが使用する他のロケールとして選んだロケールと一致していないことがあります。ロケールの設定は以下の方法で行ってください。

1. ロケール環境変数 `LANG`、`LC_COLLATE`、`LC_CTYPE`、`LC_MONETARY`、`LC_NUMERIC`、`LC_TIME`、および `LC_MESSAGES` を、他の IBM MQ プロセスが使用するロケールに設定するシェル・スクリプトを作成します。
2. 同じシェル・スクリプトで、リスナー・プログラムを呼び出します。
3. `inetd.conf` ファイルを編集して、リスナー・プログラムの代わりにこのシェル・スクリプトを呼び出すように変更します。

1つのサーバーに複数のキュー・マネージャーを存在させることができます。各キュー・マネージャーの2つのファイルのそれぞれに、1行を追加します。以下に例を示します。

```
MQSeries1 1414/tcp
MQSeries2 1822/tcp
```

```
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

`MQ_INSTALLATION_PATH` は、IBM MQ がインストールされている上位ディレクトリーを表します。

このようにすると、1つの TCP ポートのキューに入る未解決の接続要求の数に制限がある場合、エラー・メッセージの生成が回避されます。未解決の接続要求の数については、211 ページの『UNIX and Linux での TCP リスナー・バックログ・オプションの使用』を参照してください。

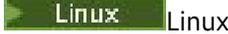
Linux

UNIX

UNIX and Linux での TCP リスナー・バックログ・オプションの使用

TCP では、サーバーとクライアントとの間で 3 ウェイ・ハンドシェイクが実行されない限り、接続は不完全として扱われます。これらの接続は、未解決接続要求と呼ばれます。最大値をこれらの未解決接続要求に設定し、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

デフォルトのリスナー・バックログ値は、[212 ページの表 21](#) に示すとおりです。

表 21. 1 つの TCP/IP ポートでキューに入れられる未解決の接続要求の最大数	
サーバーのプラットフォーム	接続要求の最大数
 AIX	100
 HP-UX	20
 Linux	100
 IBM i	255
 Solaris	100
 Windows サーバー	100
 Windows ワークステーション	100

バックログが [212 ページの表 21](#) に示された値に到達すると、TCP/IP 接続は拒否され、チャンネルを開始できません。

MCA チャンネルでは、結果としてチャンネルが RETRY 状態になり、後から接続を再試行します。

しかし、このエラーを回避するため、qm.ini ファイルに次の項目を追加できます。

```
TCP:
ListenerBacklog = n
```

こうすると、TCP/IP リスナーのデフォルトの最大未解決要求数 ([212 ページの表 21](#) を参照) が上書きされます。

**注:** 一部のオペレーティング・システムでは、デフォルトより大きい値がサポートされています。必要であれば、この値を使用して接続限度に到達するのを回避できます。

backlog オプション有効にしたままリスナーを実行するには、以下のいずれかを使用します。

- runmqclsr -b コマンドを使用する。
- MQSC コマンドの **DEFINE LISTENER** を、必要な値に設定した BACKLOG 属性を指定して使用する。

**runmqclsr** コマンドについては、runmqclsr を参照してください。DEFINE LISTENER コマンドの詳細については、[DEFINE LISTENER](#) を参照してください。

#### IBM MQ リスナーの使用

IBM MQ に提供されているリスナーは、新しいチャンネルをスレッドとして起動します。このリスナーを実行するには、runmqclsr コマンドを使用します。

以下に例を示します。

```
runmqclsr -t tcp [-m QMNAME] [-p 1822]
```

大括弧は、オプション・パラメーターを示します。デフォルト・キュー・マネージャーの場合は QMNAME は必要ありません。また、デフォルト・ポート番号 (1414) を使用する場合はポート番号は必要ありません。ポート番号は 65535 以下でなければなりません。

パフォーマンスを最高にするには、[190 ページの『承認されたアプリケーションとしてのチャンネルおよびリスナーの実行』](#) で説明されているように、承認されたアプリケーションとして IBM MQ リスナーを実行します。承認されたアプリケーションについては、[承認されたアプリケーションの制約事項](#) を参照してください。

次のコマンドを使用して、非アクティブであるキュー・マネージャーで実行されているすべての IBM MQ リスナーを停止できます。

```
endmqlsr [-m QMNAME]
```

キュー・マネージャー名を指定しないと、デフォルトのキュー・マネージャーが想定されます。

### Linux → UNIX TCP/IP SO\_KEEPALIVE オプションの使用

一部の UNIX and Linux システムでは、接続がまだ利用できるかどうか検査する前の TCP の待機時間を指定できます。また、最初の検査が失敗した場合の接続の再試行回数も指定できます。それには、カーネル調整パラメーターを使うか、またはコマンド行で指定できます。

SO\_KEEPALIVE オプション (詳細は、176 ページの『[チャンネルの相手側がまだ使用可能であるかどうかの検査](#)』を参照) を使用する場合、キュー・マネージャー構成ファイル (qm.ini) に次の項目を追加する必要があります。

```
TCP:
KeepAlive=yes
```

詳細については、運用 UNIX and Linux システムの資料を参照してください。

### Linux → UNIX UNIX and Linux での LU 6.2 接続の定義

2つのマシン間で LU 6.2 の会話が確立できるように SNA を構成する必要があります。

SNA over TCP/IP の構成に関する最新情報については、次に示すオンラインの IBM 資料を参照してください: [Communications Server](#)。

2つのシステム間で LU 6.2 の会話が確立できるように SNA を構成する必要があります。

詳細については、「[Multiplatform APPC Configuration Guide](#)」と次の表を参照してください。

リモート・プラットフォーム	TPNAME	TPPATH
CICS を使用しない z/OS	リモート・キュー・マネージャーのサイド情報の中の対応する TPName と同じ。	-
CICS を使用する z/OS	CKRC (送信側) CKSV (要求側) CKRC (サーバー)	-
IBM i	IBM i システムにおいて経路指定項目の比較値と同じ。	-
UNIX and Linux システム	リモート・キュー・マネージャーのサイド情報の中の対応する TPName と同じ。	MQ_INSTALLATION_PATH/bin/amqcrs6a
Windows	Windows Run Listener コマンドで指定されたとおり、または Windows において TpSetup を使用して定義された呼び出し可能なトランザクション・プログラム。	MQ_INSTALLATION_PATH\bin\amqcrs6a

MQ\_INSTALLATION\_PATH は、IBM MQ がインストールされている上位ディレクトリーを表します。

同一のマシン上に複数のキュー・マネージャーがある場合は、チャンネル定義内の TPname をそれぞれ固有にする必要があります。

#### 関連概念

214 ページの『[UNIX and Linux における LU 6.2 での送信側](#)』

UNIX and Linux システムの場合、CPI-C サイド・オブジェクト (シンボリック宛先) を作成して、その名前をチャンネル定義の「Connection name (接続名)」フィールドに入れます。また、パートナーに対する LU 6.2 リンクも作成します。

#### 214 ページの『UNIX and Linux における LU 6.2 での受信』

UNIX and Linux システムでは、受信側で listen 接続機構、LU 6.2 論理接続プロファイル、TPN プロファイルを作成します。

#### Linux → UNIX UNIX and Linux における LU 6.2 での送信側

UNIX and Linux システムの場合、CPI-C サイド・オブジェクト (シンボリック宛先) を作成して、その名前をチャンネル定義の「Connection name (接続名)」フィールドに入れます。また、パートナーに対する LU 6.2 リンクも作成します。

CPI-C サイド・オブジェクトには、受信側マシンのパートナー LU 名、トランザクション・プログラム名、およびモード名を入れます。以下に例を示します。

```
Partner LU Name          REMHOST
Remote TP Name          recv
Service Transaction Program no
Mode Name                #INTER
```

HP-UX では、APPCLLU 環境変数を使用して、送信側が使用するローカル LU の名前を指定します。Solaris では、ローカル LU 名になるように APPC\_LOCAL\_LU 環境変数を設定します。

SECURITY PROGRAM が CPI-C にサポートされていれば、IBM MQ は SNA セッションの確立を試みるときに、それを使用します。

#### Linux → UNIX UNIX and Linux における LU 6.2 での受信

UNIX and Linux システムでは、受信側で listen 接続機構、LU 6.2 論理接続プロファイル、TPN プロファイルを作成します。

TPN プロファイルには、実行可能ファイルへの絶対パス、およびトランザクション・プログラム名を入力します。

```
Full path to TPN executable  MQ_INSTALLATION_PATH/bin/amqcrs6a
Transaction Program name     recv
User ID                       0
```

MQ\_INSTALLATION\_PATH は、IBM MQ がインストールされている上位ディレクトリーを表します。

ユーザー ID を設定できるシステムでは、mqm グループのメンバーであるユーザーを指定します。AIX、Solaris、および HP-UX では、APPCTPN (トランザクション名) および APPCLLU (ローカル LU 名) 環境変数を設定してください (呼び出すトランザクション・プログラム用の構成パネルを使用します)。

デフォルトのキュー・マネージャー以外のキュー・マネージャーを使用しなければならないこともあります。その場合には、次のものを呼び出すコマンド・ファイルを定義します。

```
amqcrs6a -m Queue_Man_Name
```

その後で、コマンド・ファイルを呼び出します。

## IBM i IBM i 上のチャンネルのモニターおよび制御

リモート・キュー・マネージャーにつながるチャンネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM プログラムが備わっています。

### このタスクについて

次のリストに、チャンネル制御機能のコンポーネントについて簡単に説明します。

- チャンネル定義はキュー・マネージャー・オブジェクトとして保持されます。
- チャンネル・コマンドは、IBM MQ for IBM i のコマンド・セットのサブセットです。  
IBM MQ for IBM i コマンド・セット全体を表示するには、GO CMDMQM コマンドを使用します。
- チャンネル定義パネルまたはコマンドは、以下の用途に使用します。
  - チャンネル定義の作成、コピー、表示、変更、および削除
  - チャンネルの開始と停止、ping、チャンネル・シーケンス番号のリセット、およびリンクを再確立できないときの未確定メッセージの解決
  - チャンネルに関する状況情報の表示
- MQSC を使用してチャンネルを管理することもできます。
- IBM MQ エクスプローラーを使用してチャンネルを管理することもできます。
- シーケンス番号と作業論理単位 (LUW) ID は同期ファイルに格納され、チャンネル同期化のために使用されます。

コマンドおよびパネルを使用して、メッセージ・チャンネルと関連するオブジェクトを定義し、メッセージ・チャンネルをモニターおよび制御できます。F4=プロンプト・キーを使用することにより、関係のあるキュー・マネージャーを指定できます。プロンプトを使用しないと、デフォルトのキュー・マネージャーが想定されます。F4=プロンプトを使用すると、追加のパネルが表示されます。そのパネルで、関係のあるキュー・マネージャーの名前と、場合によっては他のデータを入力することができます。

パネルを使用して定義するオブジェクトは、次のとおりです。

- 伝送キュー
- リモート・キュー定義
- キュー・マネージャー別名定義
- 応答先キュー別名の定義
- 応答先ローカル・キュー
- メッセージ・チャンネル定義

これらのオブジェクトの使用に関連する概念の詳細については、[140 ページの『分散キューイングの構成』](#)を参照してください。

チャンネルは、完全に定義されていて、それに関連するオブジェクトが存在し、使用可能になっていなければ、開始することはできません。

また、各チャンネルに関する特定の通信リンクが定義されて、使用可能になっていなければ、チャンネルを実行することはできません。LU 6.2、および TCP/IP リンクの定義方法については、ご使用のシステムの特定の通信の手引きを参照してください。

## 手順

- オブジェクトの作成および処理についての詳細は、以下を参照してください。
  - [216 ページの『オブジェクトの作成 \(IBM i\)』](#)
  - [216 ページの『チャンネルの作成 \(IBM i\)』](#)
  - [218 ページの『チャンネルの開始 \(IBM i\)』](#)
  - [219 ページの『チャンネルの選択 \(IBM i\)』](#)
  - [219 ページの『チャンネルの表示 \(IBM i\)』](#)
  - [221 ページの『チャンネルの名前変更 \(IBM i\)』](#)
  - [221 ページの『チャンネル状況の操作 \(IBM i\)』](#)
  - [222 ページの『チャンネル操作の選択項目 \(IBM i\)』](#)

## 関連概念

[228 ページの『IBM i における通信のセットアップ』](#)

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。これを正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

## 関連タスク

16 ページの『サーバーとクライアント間の接続の構成』

IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャンネルを定義します。

## 関連情報

[構成例 - IBM MQ for IBM i](#)

[IBM MQ for IBM i におけるメッセージ・チャンネルの計画例](#)

[IBM MQ for IBM i CL コマンド](#)

## IBM i オブジェクトの作成 (IBM i)

CRTMQMQ コマンドを使用して、キュー・オブジェクトおよび別名オブジェクトを作成できます。

伝送キュー、リモート・キュー定義、キュー・マネージャー別名の定義、応答先キュー別名の定義、応答先ローカル・キューなどのキュー・オブジェクトおよび別名オブジェクトを作成できます。

デフォルト・オブジェクトのリストについては、[IBM MQ for IBM i でのシステムおよびデフォルト・オブジェクト](#)を参照してください。

## IBM i チャンネルの作成 (IBM i)

「チャンネルの作成」パネルから、またはコマンド行で CRTMQMCHL コマンドを使用して、チャンネルを作成できます。

チャンネルを作成するには、次のようにします。

1. 「Work with MQM Channels (MQM チャンネルの処理)」パネル (WRKMQMCHL) で F6 キーを使用します。  
あるいは、コマンド行から CRTMQMCHL コマンドを使用します。  
どちらの場合にも、「チャンネルの作成」パネルが表示されます。タイプ:
  - 表示されたフィールドにチャンネルの名前を入力します。
  - リンク側のチャンネル・タイプを入力します。
2. Enter キーを押します。

**注:** 使用するネットワーク内のすべてのチャンネルに固有の名前を付ける必要があります。すべてのチャンネルを示すネットワーク・ダイアグラムに示すように、発信元および宛先のキュー・マネージャー名をチャンネル名に含める方法を推奨します。

入力した項目は検査され、エラーがある場合にはただちに報告されます。エラーがある場合には訂正し、続行してください。

選択したチャンネルのタイプに対応するチャンネル設定パネルが表示されます。あらかじめまとめておいた情報を各フィールドに入力します。Enter キーを押してチャンネルを作成します。

チャンネル定義パネルの記述を行うさまざまなフィールドの内容を決める際には、ヘルプ・パネルにヘルプ情報が提供されます。また、[チャンネル属性](#)を参照してください。



```

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

送信出口 . . . . . *NONE_____ 名前, *SYSDFTCHL, *NONE
ライブラリー . . . . . _____ 名前
+ for more values _____
送信出口ユーザー・データ . . . . . _____
+ for more values _____
受信出口 . . . . . *NONE_____ 名前, *SYSDFTCHL, *NONE
ライブラリー . . . . . _____ 名前
+ for more values _____

受信出口ユーザー・データ . . . . . _____
+ for more values _____
メッセージ出口 . . . . . *NONE_____ 名前, *SYSDFTCHL, *NONE
ライブラリー . . . . . _____ 名前
+ for more values _____

More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys

```

図 28. チャンネルの作成 (3)

```

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

メッセージ出口ユーザー・データ . . . . . _____
+ for more values _____
メッセージを変換します。 . . . . . *SYSDFTCHL_ *YES、 *NO、 *NO、 *SYSDFTCHL
シーケンス番号の折り返し . . . . . 999999999999__ 100-999999999999、 *SYSDFTCHL
最大メッセージ長 . . . . . 4194304__0 - __ 0-4194304, *SYSDFTCHL
ハートビート間隔 . . . . . 300_____ 0 ~ 9999999999999 , *SYSDFTCHL
Non Persistent Message Speed . . . *FAST_____ *FAST, *NORMAL, *SYSDFTCHL
パスワード . . . . . *SYSDFTCHL_ 文字値 , *BLANK...
タスク・ユーザー・プロファイル . . . . *SYSDFTCHL_ 文字値 , *BLANK...
トランザクション・プログラム名 . . . . *SYSDFTCHL

Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys

```

図 29. チャンネルの作成 (4)

## IBM i チャンネルの開始 (IBM i)

「チャンネルの処理」パネルから、またはコマンド行で STRMQMCHL コマンドを使用して、チャンネルを開始できます。

リスナーは TCP にのみ有効です。SNA リスナーの場合は、通信サブシステムを構成しなければなりません。

アプリケーションがメッセージ交換を行えるようにするには、STRMQMLSR コマンドを使用して、インバウンド接続に対してリスナー・プログラムを開始する必要があります。

アウトバウンド接続については、次のいずれかの方法でチャンネルを開始させる必要があります。

1. MCATYPE パラメーターに基づいて、プロセスまたはスレッドとしてチャンネルを開始させるには、チャンネル名を指定して、CL コマンド STRMQMCHL を使用します (チャンネルがスレッドとして開始される場合には、チャンネルがチャンネル・イニシエーターのスレッドです)。

```
STRMQMCHL CHLNAME(QM1.TO.QM2) MQNAME(MYQMGR)
```

2. チャンネル・イニシエーターを使用して、チャンネルを起動します。キュー・マネージャーが開始されると、1つのチャンネル・イニシエーターが自動的に開始されます。この自動開始は、そのキュー・マネージャーの qm.ini ファイルの中の chinit スタンザを変更すると、除去できます。
3. WRKMQMCHL コマンドを使用して「Work with Channels (チャンネルの処理)」パネルを開始し、オプション 14 を選択してチャンネルを開始します。

## IBM i チャンネルの選択 (IBM i)

「チャンネルの処理」パネルから、チャンネルを選択できます。

チャンネルを選択するには、WRKMQMCHL コマンドを使用して「Work with Channels (チャンネルの処理)」パネルで次の手順を実行してください。

1. 必要なチャンネル名に関連付けられたオプション・フィールドにカーソルを移動させます。
2. オプション番号を入力します。
3. Enter キーを押して選択項目をアクティブ化させます。

複数のチャンネルを選択すると、オプションは順番にアクティブ化されます。

```
Work with MQM Channels
```

```
キュー・マネージャー名 . : CNX -
```

```
Type options, press Enter.
```

```
2=Change 3=Copy 4=Delete 5=Display 8=Work with Status 13=Ping  
14=Start 15=End 16=Reset 17=Resolve
```

Opt	Name	Type	Transport	Status
	CHLNIC	*RCVR	*TCP	INACTIVE
	CORSAIR.TO.MUSTANG	*SDR	*LU62	INACTIVE
	FV.CHANNEL.MC.DJE1	*RCVR	*TCP	INACTIVE
	FV.CHANNEL.MC.DJE2	*SDR	*TCP	INACTIVE
	FV.CHANNEL.MC.DJE3	*RQSTR	*TCP	INACTIVE
	FV.CHANNEL.MC.DJE4	*SVR	*TCP	INACTIVE
	FV.CHANNEL.PETER	*RCVR	*TCP	INACTIVE
	FV.CHANNEL.PETER.LU	*RCVR	*LU62	INACTIVE
	FV.CHANNEL.PETER.LU1	*RCVR	*LU62	INACTIVE

```
More...
```

```
Parameters or command
```

```
====>
```

```
F3=Exit F4=Prompt F5=Refresh F6=Create F9=Retrieve F12=Cancel  
F21=Print
```

図 30. チャンネルの操作

## IBM i チャンネルの表示 (IBM i)

「チャンネルの表示」パネルから、またはコマンド行で DSPMQMCHL コマンドを使用して、チャンネルをブラウズできます。

チャンネルの設定をブラウズするには、WRKMQMCHL コマンドを使用して「Display Channel (チャンネルの表示)」パネルで次の手順を実行してください。

1. 必要なチャンネル名に対してオプション 5 (表示) を入力します。
2. Enter キーを押して選択項目をアクティブ化させます。

複数のチャンネルを選択すると、チャンネルは順番に表示されます。

あるいは、コマンド行から DSPMQMCHL コマンドを使用します。

すると、そのチャンネルの現行設定値の詳細を示した適切な「チャンネルの表示」パネルが表示されます。フィールドについては、[チャンネル属性](#)を参照してください。

```

Display MQM Channel

チャンネル名 . . . . . : ST.JST.2T01
キュー・マネージャー名 . . . . . : キュレル
チャンネル・タイプ . . . . . : *SDR
トランスポート・タイプ . . . . . : *TCP
テキストの「説明」 . . . . . : John の送信者の WINSDOA1

接続名 . . . . . : マスタング

伝送キュー . . . . . : WINSDOA1

メッセージ・チャンネル・エージェント . . . :
ライブラリー . . . . . :
Message channel agent user ID : *NONE
Batch interval . . . . . : 0
バッチ・サイズ . . . . . : 50
切断間隔 . . . . . : 6000

F3=Exit F12=Cancel F21=Print

```

図 31. TCP/IP チャンネルの表示 (1)

```

Display MQM Channel

短い再試行間隔 . . . . . : 60
短い再試行カウント . . . . . : 10
長い再試行間隔。 . . . . : 6000
長い再試行カウント . . . . . : 10
セキュリティー出口 . . . . . :
ライブラリー . . . . . :
セキュリティー出口ユーザー・データ . . . :
送信出口 . . . . . :
ライブラリー . . . . . :
Send exit user data . . . . . :
受信出口 . . . . . :
ライブラリー . . . . . :
受信出口ユーザー・データ . . . . . :
メッセージ出口 . . . . . :
ライブラリー . . . . . :
メッセージ出口ユーザー・データ . . . . . :
More...

F3=Exit F12=Cancel F21=Print

```

図 32. TCP/IP チャンネルの表示 (2)

Display MQM Channel

シーケンス番号の折り返し . . . . . : 999999999  
最大メッセージ長 . . . . . : 10000  
メッセージを変換します。 . . . : \*いいえ  
ハートビート間隔 . . . . . : 300  
非持続メッセージ速度。 \* 高速

Bottom

F3=Exit F12=Cancel F21=Print

図 33. TCP/IP チャンネルの表示 (3)

### IBM i チャンネルの名前変更 (IBM i)

「チャンネルの処理」パネルから、チャンネルの名前を変更できます。

メッセージ・チャンネルの名前を変更するためには、「Work with Channels (チャンネルの処理)」パネルで次の手順を実行してください。

1. チャンネルを終了させます。
2. オプション「3 (Copy (コピー))」を選び、新しい名前チャンネルの複製を作ります。
3. オプション「5 (Display (表示))」を選び、その複製チャンネルが正しく作成されたか確認します。
4. オプション「4 (Delete (削除))」を選び、元のチャンネルを削除します。

メッセージ・チャンネルの名前を変更する際には、チャンネルの両側が同時に名前変更されるようにしてください。

### IBM i チャンネル状況の操作 (IBM i)

「チャンネル状況の処理」パネルからチャンネル状況を処理できます。

WRKMQMCHST コマンドを使用すると、使用チャンネルの状況を示す一連のパネルのうちの最初のパネルが表示されます。「パネルの変更」(F11)を選択すると、状況パネルを順番に見ることができます。

あるいは、「MQM チャンネルの処理」パネルからオプション「8 (状況の処理)」を選択しても、最初の状況パネルを表示することができます。

## MQSeries Work with Channel Status

Type options, press Enter.

5=Display 13=Ping 14=Start 15=End 16=Reset 17=Resolve

Opt Name	Connection	Indoubt	Last Seq
CARTS_CORSAIR_CHAN	GBIBMIYA.WINSDO01	NO	1
CHLNIC	9.20.2.213	NO	3
FV.CHANNEL.PETER2	9.20.2.213	NO	6225
JST.1.2	9.20.2.201	NO	28
MP_MUST_TO_CORS	9.20.2.213	NO	100
MUSTANG_TO_CORSAIR	GBIBMIYA.WINSDO01	NO	10
MP_CORS_TO_MUST	9.20.2.213	NO	101
JST.2.3	9.5.7.126	NO	32
PF_WINSDO01_LU62	GBIBMIYA.IYA80020	NO	54
PF_WINSDO01_LU62	GBIBMIYA.WINSDO01	NO	500
ST.JCW.EXIT.2T01.CHL	9.20.2.213	NO	216

Bottom

Parameters or command

==>

F3=Exit F4=Prompt F5=Refresh F6=Create F9=Retrieve F11=Change view

F12=Cancel F21=Print

図 34. 一連のチャンネル状況パネルの最初のパネル

「Work with Channel Status (チャンネル状況の処理)」パネルで利用できるオプションは、次のとおりです。

メニュー・オプション	説明
5=Display	チャンネル設定値を表示します。
13=Ping	該当する場合には、ping アクションを開始します。
14=Start	チャンネルを開始します。
15=End	チャンネルを停止します。
16=Reset	チャンネル・シーケンス番号をリセットします。
17=Resolve	未確定チャンネル状況を手動で解決します。

## IBM i チャンネル操作の選択項目 (IBM i)

「Work with Channels (チャンネルの処理)」パネルは、WRKMQMCHL コマンドによって表示でき、これを使用して、リストされたすべてのチャンネルの状況をモニターしたり、選択したチャンネルに対してコマンドを実行したりできます。

「Work with Channels (チャンネルの処理)」パネルで利用できるオプションは、次のとおりです。

メニュー・オプション	説明
<u>223 ページの</u> 『2=Change』	チャンネルの属性を変更します。
<u>223 ページの</u> 『3=Copy』	あるチャンネルの属性を新しいチャンネルにコピーします。
<u>223 ページの</u> 『4=Delete』	チャンネルを削除します。
<u>223 ページの</u> 『5=Display』	チャンネルの現行設定値を表示します。
<u>223 ページの</u> 『6=Create』	「チャンネルの作成」パネルの表示
<u>224 ページの</u> 『8=Work with Status』	チャンネル状況パネルを表示します。
<u>225 ページの</u> 『13=Ping』	ping 機能を実行し、リモート側と固定データ・メッセージを交換して、隣接システムとの接続をテストします。

## メニュー・オプション 説明

225 ページの『14=Start』	選択されたチャンネルを開始するか、あるいは使用不可の受信側チャンネルをリセットします。
226 ページの『15=End』	チャンネルの終了を要求します。
227 ページの『16=Reset』	リンク先でシーケンス番号をリセットするようにチャンネルに要求します。シーケンス番号は、チャンネルの両側で同じになっていなければ開始できません。
227 ページの『17=Resolve』	反対側の接続を確立しないで未確定メッセージを解決するようにチャンネルに要求します。
227 ページの『18=Display authority』	IBM MQ オブジェクト権限を表示します。
227 ページの『19=Grant authority』	IBM MQ オブジェクト権限を付与します。
228 ページの『20=Revoke authority』	IBM MQ オブジェクト権限を取り消します。
228 ページの『21=Recover object』	IBM MQ オブジェクトをリカバリーします。
228 ページの『22=Record image』	IBM MQ オブジェクト・イメージを記録します。

## IBM i 2=Change

「Change (変更)」オプションは、既存のチャンネル定義を変更するときに使用します。

「Change (変更)」オプションまたは CHGMQMCHL コマンドは、チャンネル名以外の既存のチャンネル定義を変更します。チャンネル定義パネル内で変更したいフィールドを上書きし、Enter キーを押して更新済みの定義を保存してください。

## IBM i 3=Copy

「Copy (コピー)」オプションは、既存のチャンネルをコピーするときに使用します。

「Copy (コピー)」オプションは CPYMQMCHL コマンドを使用して、既存のチャンネルをコピーします。「Copy (コピー)」パネルを使用すると、新しいチャンネル名を定義できます。ただし、IBM i オブジェクト名として有効な文字だけを使用するようにしてください。IBM MQ for IBM i の管理を参照してください。

「Copy (コピー)」パネルで Enter キーを押して、現行設定値の詳細を表示してください。任意の新規チャンネル設定値を変更できます。Enter キーを押して、新しいチャンネル定義を保管してください。

## IBM i 4=Delete

「Delete (削除)」オプションは、特定のチャンネルを削除するために使用します。

要求の確認または取り消しを行うためのパネルが表示されます。

## IBM i 5=Display

「Display (表示)」オプションは、チャンネルの現行定義を表示するために使用します。

これを選択すると、フィールドにパラメーターの現行値が入っているパネルが表示されます。これらのフィールドは、ユーザー入力が行われないように保護されています。

## IBM i 6=Create

「Create (作成)」オプションは、「Create channel (チャンネルの作成)」パネルを表示するときに使用します。

「Create (作成)」オプションを使用するか、またはコマンド行で CRTMQMCHL コマンドを入力して、「Create Channel (チャンネルの作成)」パネルを表示します。217 ページの図 26 以降には、「Create Channel (チャンネルの作成)」パネルの例があります。

このパネルを使用すると、IBM MQ for IBM i によって提供されるデフォルト値がフィールドに入った画面に基づいて、チャンネル定義を作成できます。チャンネルの名前を入力し、作成するチャンネルのタイプおよび使用する通信方式を入力してください。

Enter キーを押すと、パネルが表示されます。このパネルとその他のパネル内の必要なすべてのフィールドに情報を入力してから、Enter キーを押して定義を保存します。

チャンネル名はチャンネルの両端で同じ名前になっていなければならない、またネットワーク内で固有の名前でなければなりません。ただし、IBM MQ for IBM i オブジェクト名として有効な文字だけを使用するようにしてください。

どのパネルでも、いくつかのフィールドには IBM MQ for IBM i によってデフォルト値が提供されます。これらの値をカスタマイズしたり、またはチャンネルの作成もしくはコピー時に値を変更したりできます。値のカスタマイズについては、「IBM MQ for IBM i システム管理ガイド」を参照してください。

独自のチャンネル・デフォルト値のセットを作成するには、各チャンネル・タイプに必要なデフォルト値を指定したダミー・チャンネルを設定しておいて、新しいチャンネル定義の作成が必要になるたびにそれらのチャンネルをコピーすることができます。

## 関連情報

チャンネルの属性

### IBM i 8=Work with Status

「状況の処理」は、詳細なチャンネルの状況情報を表示するときに使用します。

「status (状況)」列は、チャンネルがアクティブ、または非アクティブかを示し、「Work with MQM channels (MQM チャンネルの処理)」パネルでも引き続き表示されます。さらに状況情報を表示するには、オプション「8 (Work with Status (状況の処理))」を選択します。あるいは、コマンド行から WRKMQMCHST コマンドを使用してこの情報を表示させることもできます。[221 ページの『チャンネル状況の操作 \(IBM i\)』](#)を参照してください。

- チャンネル名
- チャンネル・タイプ
- チャンネル状況
- チャンネル・インスタンス
- リモート・キュー・マネージャー
- 伝送キュー名
- 通信接続名
- チャンネルの未確定状況
- 最終シーケンス番号
- 未確定メッセージ数
- 未確定シーケンス番号
- 伝送キュー上のメッセージ数
- 論理作業単位 ID
- 未確定作業論理単位 ID
- チャンネル副状態
- チャンネル・モニター
- ヘッダー圧縮
- メッセージ圧縮
- 圧縮時間標識
- 圧縮率標識
- 伝送キュー時間標識
- ネットワーク時間標識

- 終了時間標識
- バッチ・サイズ標識
- 現行共有会話
- 最大共有会話

## IBM i 13=Ping

固定データ・メッセージをリモート側に交換するには、Ping オプションを使用します。

IBM MQ の Ping に成功すると、システム監視プログラムは、チャンネルが使用可能であって機能していることをある程度確認することができます。

Ping を選択しても、伝送キューとターゲット・キューが使用されることはありません。チャンネル定義、関連する通信リンク、およびネットワーク設定が使用されます。

Ping は、送信側チャンネル、およびサーバー・チャンネルからのみ使用できます。リンクの反対側で対応チャンネルが開始され、開始パラメーターの折衝が実施されます。エラーがある場合には、通常どおり通知されます。

メッセージ交換の結果は Ping パネルに表示されます。これには、戻されたメッセージ・テキスト、メッセージの送信時刻、および応答の受信時刻も表示されます。

## LU 6.2 での ping

IBM MQ for IBM i で ping が呼び出されるときには、その機能を要求したユーザーのユーザー ID で実行されます。これに対し、チャンネル・プログラムの通常の実行方法では、QMQM ユーザー ID がチャンネル・プログラム用に使用されます。ユーザー ID は受信側まで流れるため、これが受信側で有効でなければ LU 6.2 の会話を割り振ることはできません。

## IBM i 14=Start

「Start (開始)」オプションは、チャンネルを手動で開始するときに使用します。

「Start (開始)」オプションは送信側、サーバー、および要求側のチャンネルで利用できます。これは、チャンネルがキュー・マネージャーによるトリガー操作でセットアップされている場合には必要ありません。

「Start (開始)」オプションは、受信側、サーバー接続、クラスター送信側、およびクラスター受信側の各チャンネルでも使用されます。STOPPED 状態になっている受信側チャンネルの開始は、リモート・チャンネルからそのチャンネルを開始できることを意味しています。

送信側 MCA は、開始されると、チャンネル定義ファイルを読み取り、伝送キューをオープンします。チャンネル開始シーケンスが実行され、リモートで受信側またはサーバー側のチャンネルの対応する MCA が開始されます。それらのチャンネルが開始すると、送信側プロセスおよびサーバー・プロセスはメッセージが伝送キューに到達するのを待ち、到達したメッセージを伝送します。

トリガー操作を使用する場合には、絶えず実行されているトリガー・プロセスを開始して、開始キューをモニターする必要があります。プロセスを開始するために、STRMQMCHLI コマンドを使用します。

チャンネルの反対側では、送信側からのチャンネル開始に応答して受信側プロセスが開始することがあります。そのために使用される方式は、以下のように、LU 6.2 接続チャンネルと TCP/IP 接続チャンネルとは異なります。

- LU 6.2 接続チャンネルの場合、チャンネルの受信側では明示的なアクションは不要です。
- TCP 接続チャンネルの場合は、リスナー・プロセスが絶えず実行されている必要があります。このプロセスはリンクのリモート側からのチャンネル開始要求を待ち、その接続に関してチャンネル定義で定義されたプロセスを開始します。

リモート・システムが IBM i の場合には、STRMQMLSR コマンドを使用できます。

「Start (開始)」オプションを使用すると、必要な場合には必ずチャンネルが再同期化されます。

開始手続きを正常に行うためには、次の条件が必要です。

- ローカルおよびリモートのチャンネル定義が存在している。受信側またはサーバー接続のチャンネルに該当するチャンネル定義がない場合は、チャンネルが自動定義されていると、自動的にデフォルトのチャンネル定義が作成されます。 [チャンネル自動定義出口プログラム](#)を参照してください。
- 伝送キューが存在していて、読み取りのために使用することが可能になっており、しかも他のチャンネルで使用されていない。
- ローカルおよびリモートの MCA が存在している。
- 通信リンクが利用可能になっている。
- ローカルおよびリモートのキュー・マネージャーが実行されている。
- メッセージ・チャンネルが非アクティブになっている。

メッセージを転送するには、リモート・キューとリモート・キュー定義が存在していなければなりません。

チャンネル開始要求が受け入れられたことを示すメッセージがパネルに表示されます。開始プロセスが成功したかどうかを確認するためには、システム・ログを調べるか、あるいは F5 (画面最新表示) キーを押してください。

## IBM i 15=End

「End (終了)」は、チャンネル活動を停止するために使用します。

「End (終了)」オプションは、チャンネル活動の停止を要求するために使用します。チャンネルは、これ以上メッセージを送信しません。

Enter キーを押す前に F4 を選択し、チャンネルが STOPPED と INACTIVE のどちらになるか、さらに CONTROLLED と IMMEDIATE のどちらの停止を使用してチャンネルを停止するかを選択します。停止状態のチャンネルを再度アクティブにするは、オペレーターが再開する必要があります。非アクティブのチャンネルはトリガーできます。

## 即時停止

「即時停止」を使用すると、作業単位がある場合でもそれを完了せずにチャンネルを停止します。

このオプションは、チャンネル・プロセスを終了してしまいます。チャンネルはメッセージの現行バッチの処理を完了しないため、そのチャンネルを未確定のままにしておくことはできません。一般に、オペレーターは制御された停止オプションを使用する方が良いです。

## 制御された停止

「制御された停止」を使用すると、現在の作業単位が終わったらチャンネルを停止します。

これを選択すると、チャンネルが正常にクローズし、メッセージの現行バッチが完了するとともに、チャンネルの反対側の端点で同期点手順が実行されます。

## 停止したチャンネルの再始動

チャンネルが STOPPED 状態になるときは、手動でチャンネルを再始動しなければなりません。以下の方法でチャンネルを再始動できます。

- START CHANNEL** MQSC コマンドを使用します。
- Start Channel** PCF コマンドを使用します。
- IBM MQ Explorer を使用します。
-  z/OS で、「チャンネルの開始 (Start a channel)」パネルを使用します。
-  IBM i で、**STRMQMCHL CL** コマンドまたは WRKMQMCHL パネルの **START** オプションを使用します。

送信側またはサーバー・チャンネルでは、チャンネルが停止状態になった場合、関連する伝送キューは GET(DISABLED) に設定され、トリガー操作はオフに設定されます。開始要求が受信されたとき、これらの属性は自動的にリセットされます。

**z/OS** チャンネルが RETRYING または STOPPED の状況の間にチャンネル・イニシエーターが停止した場合、チャンネル・イニシエーターの再開始時にチャンネル状況が記憶されています。ただし、SVRCONN チャンネル・タイプのチャンネル状況は、チャンネルが STOPPED の状況の間にチャンネル・イニシエーターが停止すると再設定されます。

**Multi** チャンネルが RETRYING または STOPPED の状況の間にキュー・マネージャーが停止した場合、キュー・マネージャーの再開始時にチャンネル状況が記憶されています。IBM MQ 8.0 以降、これは SVRCONN チャンネルにも当てはまります。以前は、SVRCONN チャンネル・タイプのチャンネル状況は、チャンネルが STOPPED の状況の間にチャンネル・イニシエーターが停止するとリセットされていました。

## **IBM i** 16=Reset

「Reset (リセット)」オプションは、メッセージ・シーケンスを強制的に新しくするときを使用します。

「Reset (リセット)」オプションは、メッセージ・シーケンス番号を変更するために使用します。このオプションを使用するには注意が必要です。未確定状態がある場合は、必ず、「Resolve (解決)」オプションを使用してそれを解決してからこのオプションを使用するようにしてください。このオプションは、送信側チャンネルまたはサーバー・チャンネルだけで使用可能です。次にそのチャンネルが開始されると、最初のメッセージは新しいシーケンス番号になります。

## **IBM i** 17=Resolve

「Resolve (解決)」オプションは、伝送キューに保持される未確定メッセージのローカル側でのコミットまたはバックアウトを強制するときを使用します。

リンクの送信側かサーバー側が終了してしまいリカバリーの見込みがないなどの理由で、メッセージがいずれかの側で未確定のままの場合は、「(Resolve (解決))」オプションを使用します。「Resolve (解決)」オプションには、BACKOUT か COMMIT のどちらかのパラメーターを設定できます。バックアウトではメッセージは伝送キューに復元され、コミットではメッセージは廃棄されます。

チャンネル・プログラムは、パートナーとのセッションの確立を試みません。その代わりに、未確定メッセージを表す作業論理単位 ID (LUWID) を判別します。そのあとで、要求に応じて次のいずれかを発行します。

- メッセージを伝送キューに復元するための BACKOUT
- メッセージを伝送キューから削除するための COMMIT

解決を正常に行うためには、次の条件が必要です。

- チャンネルが非アクティブになっている。
- チャンネルが未確定になっている。
- チャンネル・タイプが送信側またはサーバーである。
- ローカルのチャンネル定義が存在している。
- ローカルのキュー・マネージャーが実行されている。

## **IBM i** 18=Display authority

「Display authority (権限の表示)」オプションは、特定の IBM MQ オブジェクトに対してユーザーがどのアクションの実行を許可されているかを表示するときを使用します。

DSPMQAUT コマンドは、選択したオブジェクトおよびユーザーについて、IBM MQ オブジェクトに対してアクションを実行するための、ユーザーが持つ権限を表示します。そのユーザーが複数のグループのメンバーである場合は、そのオブジェクトへのすべてのグループの権限を組み合わせたものがこのコマンドで表示されます。

## **IBM i** 19=Grant authority

「Grant authority (権限の付与)」オプションは、IBM MQ オブジェクトに対するアクションを実行する権限を別のユーザーまたはユーザー・グループに付与するときを使用します。

GRTMQMAUT コマンドは、QMADM グループのユーザーのみに使用できます。QMADM のユーザーは、ユーザーを名前前で指定するか、または \*PUBLIC のすべてのユーザーに権限を付与するかのいずれかに

より、コマンドで指定された IBM MQ オブジェクトに対してアクションを実行する権限を他のユーザーに付与します。

### IBM i 20=Revoke authority

「Revoke authority (権限の取り消し)」は、オブジェクトに対するアクション実行の許可をユーザーから除去するときに使用します。

RVKMQMAUT コマンドは、QMADM グループのユーザーのみに使用できます。QMADM グループのユーザーは、ユーザーを名前で指定するか、または \*PUBLIC のすべてのユーザーから権限を取り消すかのいずれかにより、コマンドで指定された IBM MQ オブジェクトに対してアクションを実行する権限を他のユーザーから除去します。

### IBM i 21=Recover object

「Recover object (オブジェクトのリカバリー)」は、損傷したオブジェクトを、IBM MQ ジャーナルに保管されている情報を使用してリカバリーするときに使用します。

「Recover object (オブジェクトのリカバリー)」では、「MQ オブジェクトの再作成」コマンド (RCRMQMOBJ) を使用して、コマンドで指定された損傷したすべてのオブジェクトをリカバリーします。オブジェクトが損傷していなければ、そのオブジェクトに対してはアクションは実行されません。

### IBM i 22=Record image

「Record image (イメージの記録)」は、一連のオブジェクトのリカバリーのために必要なジャーナル・レシーバーの数を削減するため、さらにリカバリー時間を最小化するために使用します。

RCDMQMIMG コマンドは、コマンドで選択されたすべてのオブジェクトでチェックポイントを取ります。このコマンドは、統合ファイル・システム (IFS) 内のオブジェクトの現行値を、ジャーナル・レシーバーに記録された MQPUT や MQGET などの、そのオブジェクトに関する後の情報と同期化します。

コマンドが完了すると、IFS 内のオブジェクトは最新になり、このオブジェクトをリカバリーするためのそれらジャーナル・レシーバーは、もはや必要なくなります。切断されたジャーナル・レシーバーは切り離すことができます (これらがその他のオブジェクトをリカバリーするために必要である場合を除く)。

## IBM i IBM i における通信のセットアップ

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。これを正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

DQM は IBM MQ for IBM i 用のリモート・キューイング機能です。これは IBM MQ for IBM i キュー・マネージャーにチャネル制御プログラムを提供し、このプログラムによって、通信リンクに対する、システム・オペレーターからの制御が可能なインターフェースが形成されます。

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。これを正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。このセクションでは、確実に接続が定義され、使用可能になる方法について説明します。

チャネルを開始する前に、このセクションの説明に従って伝送キューを定義して、メッセージ・チャネル定義に組み込んでおく必要があります。

IBM MQ for IBM i システム間の通信の次の 2 つの形式から選択することができます。

#### • [229 ページの『IBM i での TCP 接続の定義』](#)

TCP の場合、ホスト・アドレスを使用することができ、これらの接続は「*IBM i Communication Configuration Reference*」で説明されているようにセットアップされます。

TCP 環境では、各分散サービスには、そのサービスにアクセスするためにリモートのマシンで使用される可能性がある、固有の TCP アドレスが割り当てられます。TCP アドレスは、ホストの名前/番号およびポート番号で構成されます。すべてのキュー・マネージャーは、この番号を使用して、TCP を介して相互に通信します。

#### • [229 ページの『TCP での受信』](#)

この通信を行うには、ローカル・キュー・マネージャーを備えた IBM i システムとリモート・キュー・マネージャーを備えたシステムとの間の物理リンクを使用できるように、IBM i SNA 論理装置タイプ 6.2

(LU 6.2) を定義しなければなりません。IBM i での通信の構成について詳しくは、「*IBM i Communication Configuration Reference*」を参照してください。

さらに、必要に応じて、必要なプロセスとキューの定義を行ってトリガー操作の準備も整えておく必要があります。

## 関連タスク

214 ページの『[IBM i 上のチャネルのモニターおよび制御](#)』

リモート・キュー・マネージャーにつながるチャネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。各キュー・マネージャーには、互換性のあるリモート・キュー・マネージャーとの相互接続を制御するための DQM プログラムが備わっています。

## 関連情報

[構成例 - IBM MQ for IBM i](#)

[IBM MQ for IBM i におけるメッセージ・チャネルの計画例](#)

[IBM i における相互通信ジョブ](#)

[IBM i におけるチャネルの状態](#)

## IBM i IBM i での TCP 接続の定義

「接続名」フィールドを使用して、チャネル定義内に TCP 接続を定義できます。

チャネル定義には CONNECTION NAME フィールドがあり、ターゲットの TCP ネットワーク・アドレスか、ホスト名 (例えば ABCHOST) が含まれています。TCP ネットワーク・アドレスは、IPv4 小数点付き 10 進数形式 (例えば、127.0.0.1) または IPv6 16 進数形式 (例えば、2001:DB8:0:0:0:0:0) になります。CONNECTION NAME がホスト名またはネーム・サーバーである場合、ホスト名を TCP ホスト・アドレスに変換するために、IBM i ホスト・テーブルが使用されます。

TCP アドレスを完成させるには、ポート番号が必要です。ポート番号を指定しなければ、デフォルト・ポート番号 1414 が使用されます。接続の開始側 (送信側、要求側、およびサーバーのチャネル・タイプ) では、例えば、次のようにオプションの接続用ポート番号を指定できます。

```
Connection name 127.0.0.1 (1555)
```

この場合、接続の開始側ではポート 1555 で受信プログラムに接続しようとしています。

## TCP リスナー・バックログ・オプションの使用

TCP では、サーバーとクライアントとの間で 3 ウェイ・ハンドシェイクが実行されない限り、接続は不完全として扱われます。これらの接続は、未解決接続要求と呼ばれます。最大値をこれらの未解決接続要求に設定し、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

詳細については、211 ページの『[UNIX and Linux での TCP リスナー・バックログ・オプションの使用](#)』を参照してください。また、IBM i の特定の値を参照してください。

## 関連概念

229 ページの『[TCP での受信](#)』

受信側のチャネル・プログラムは、送信側チャネルからの開始要求に応じて開始されます。開始要求に応答するには、リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャネルを開始する必要があります。このリスナー・プログラムは、STRMQMLSR コマンドで開始してください。

## IBM i TCP での受信

受信側のチャネル・プログラムは、送信側チャネルからの開始要求に応じて開始されます。開始要求に応答するには、リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャネルを開始する必要があります。このリスナー・プログラムは、STRMQMLSR コマンドで開始してください。

キュー・マネージャーごとに複数のリスナーを開始することができます。デフォルトでは、STRMQMLSR コマンドはポート 1414 を使用しますが、この値は指定変更できます。デフォルト設定を指定変更するには、選択したキュー・マネージャーの qm.ini ファイルに以下のステートメントを追加します。次の例では、リスナーはポート 2500 を使用する必要があります。

```
TCP:
Port=2500
```

qm.ini ファイルは次の IFS ディレクトリーにあります。/QIBM/UserData/mqm/qmgrs/*queue manager name*

この新しい値は、TCP リスナー・プログラムが開始するときだけ読み取られます。リスナーがすでに開始されている場合、この変更はそのプログラムからは見えません。新しい値を使用するためには、リスナーを停止して STRMQMLSR コマンドを実行し直します。現時点では、STRMQMLSR コマンドを使用した場合、リスナーは常に新しいポートにデフォルト解釈されます。

また、STRMQMLSR コマンドに異なるポート番号を指定することもできます。以下に例を示します。

```
STRMQMLSR MQMNAME( queue manager name ) PORT(2500)
```

このように変更すると、リスナー・ジョブが続く限りリスナーのデフォルト値は新しいポートになります。

## TCP SO\_KEEPALIVE オプションの使用

SO\_KEEPALIVE オプション (詳細は、[176 ページの『チャネルの相手側がまだ使用可能であるかどうかの検査』](#)を参照) を使用するためには、キュー・マネージャー構成ファイル (IFS ディレクトリー /QIBM/UserData/mqm/qmgrs/*queue manager name* にある qm.ini) に次の項目を追加する必要があります。

```
TCP:
KeepAlive=yes
```

そのあと、次のコマンドを実行する必要があります。

```
CFGTCP
```

オプション「3 (Change TCP Attributes (TCP 属性の変更))」を選択します。ここで、時間間隔を分単位で指定できます。値は、1 から 40320 分の範囲内で指定できます。デフォルトは 120 です。

## TCP リスナー・バックログ・オプションの使用

TCP 上で受信する場合、未解決の接続要求の最大数が設定されます。この数は、リスナーが要求を受け入れるのを TCP ポート上で待っている要求のバックログと考えることができます。

IBM i でのデフォルト・リスナー・バックログ値は 255 です。バックログがこの値に到達すると、TCP 接続は拒否され、チャネルを開始できません。

MCA チャネルの場合は、チャネルが RETRY 状態になり、後で接続が再試行されます。

クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC\_Q\_MGR\_NOT\_AVAILABLE を受け取り、後で接続を再試行することができます。

しかし、このエラーを回避するため、qm.ini ファイルに次の項目を追加できます。

```
ListenerBacklog = n
```

この項目は、TCP リスナーのデフォルトの最大未解決要求数 (255) を上書きします。

注：一部のオペレーティング・システムでは、デフォルトより大きい値がサポートされています。必要であれば、この値を使用して接続限度に到達するのを回避できます。

## IBM i IBM i での LU 6.2 接続の定義

完全修飾された LU 6.2 接続のモード名、TP 名、および接続名を使用して、LU 6.2 通信の詳細を定義します。

リンクが開始される側では、この CSI オブジェクトを完成するための経路指定項目の定義が必要です。リモート LU 6.2 システムからの作業要求の管理についての詳細は、「*IBM i Programming: Work Management Guide*」に記載されています。

詳細については、「*Multiplatform APPC Configuration Guide*」と次の表を参照してください。

表 23. ローカル IBM i システムにおけるリモート・キュー・マネージャー・プラットフォーム用の設定値	
リモート・プラットフォーム	TPNAME
z/OS または MVS	リモート・キュー・マネージャーの対応するサイド情報内と同じ。
IBM i	IBM i システムにおいて経路指定項目の比較値と同じ。
UNIX and Linux システム	リモート LU 6.2 構成で定義された呼び出し可能なトランザクション・プログラム。
Windows	Windows Run Listener コマンドで指定されたとおり、または Windows において TpSetup を使用して定義された呼び出し可能なトランザクション・プログラム。

同一のコンピューター上に複数のキュー・マネージャーがある場合は、チャンネル定義内の TPname をそれぞれ固有にする必要があります。

### 関連概念

#### 231 ページの『開始される側 (送信側)』

トランスポート・タイプ \*LU62 のチャンネルを定義するには、CRTMQMCHL コマンドを使用してください。

#### 234 ページの『開始される側 (受信側)』

トランスポート・タイプ \*LU62 のメッセージ・チャンネル・リンクの受信側を定義するには、CRTMQMCHL コマンドを使用してください。

#### IBM i 開始される側 (送信側)

トランスポート・タイプ \*LU62 のチャンネルを定義するには、CRTMQMCHL コマンドを使用してください。

CSI オブジェクトの使用は、IBM MQ for IBM i 5.3 以降ではオプションです。

開始される側のパネルを、[図 LU 6.2 通信セットアップ・パネル - 開始側](#)に示しています。図示されているパネルの全体を確認するには、最初のパネルで F10 を押してください。

```

Create Comm Side Information (CRTCSI)

Type choices, press Enter.

Side information . . . . . > WINSDOA1  Name
Library . . . . . > QSYS      Name, *CURLIB
Remote location . . . . . > WINSDOA1  Name
Transaction program . . . . . > MQSERIES

Text 'description' . . . . . *BLANK

Additional Parameters

Device . . . . . *LOC      Name, *LOC
Local location . . . . . *LOC      Name, *LOC, *NETATR
Mode . . . . . JSTMOD92  Name, *NETATR
Remote network identifier . . . *LOC      Name, *LOC, *NETATR, *NONE
Authority . . . . . *LIBCRTAUT Name, *LIBCRTAUT, *CHANGE...

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

図 35. LU 6.2 通信セットアップ - 開始側

次のようにして、開始側のフィールドを完成させてください。

## サイド情報

作成するサイド情報オブジェクトを保管するために使用する名前 (例えば、WINSDOA1) を、この定義に付けてください。

**注:** LU 6.2 の場合、メッセージ・チャンネル定義と通信接続の間のリンクは、送信側のメッセージ・チャンネル定義における「**Connection name (接続名)**」フィールドになります。このフィールドには、CSI オブジェクトの名前が入ります。

## ライブラリー

この定義が保管されるライブラリーの名前です。

CSI オブジェクトは、メッセージ・チャンネルをサービスするプログラムからアクセスできるライブラリー (例えば、QSYS、QMQM、および QGPL) で使用可能でなければなりません。

この名前が誤っていたり、欠落していたり、検出できなかったりする場合には、チャンネル開始時にエラーが発生します。

## リモート・ロケーション (Remote location)

プログラムが通信するリモート・ロケーションの名前を指定してください。

この必須パラメーターには、2つのシステム間の通信リンクに使用される装置記述で定義された、リモート・システムのパートナーの論理装置名が入ります。

「**Remote location (リモート・ロケーション)**」名は、リモート・システムで DSPNETA コマンドを実行し、デフォルト・ローカル・ロケーション名を調べることによって得られます。

## トランザクション・プログラム (Transaction program)

開始するリモート・システムのトランザクション・プログラムの名前を指定します (64 文字まで)。トランザクション・プロセス名、プログラム名、チャンネル名、または経路指定項目の「**Compare value (値の比較)**」に一致する文字ストリングを指定することができます。

このパラメーターは必須です。

**注:** SNA サービス・トランザクション・プログラム名を指定するには、サービス・トランザクション・プログラム名を 16 進表記で入力してください。例えば、16 進表記で 21F0F0F1 となるサービス・トランザクション・プログラム名を指定する場合には、X'21F0F0F1' と入力します。

SNA サービス・トランザクション・プログラム名の詳細については、LU タイプ 6.2 用の「*SNA Transaction Programmer's Reference*」に記載されています。

受信側が別の IBM i システムである場合、送信側の CSI オブジェクトと受信側の経路指定項目を突き合わせるために「**Transaction program (トランザクション・プログラム)**」名が使用されます。この名前は、宛先 IBM i システム上のキュー・マネージャーごとに固有でなければなりません。開始される側 (受信側)の「**呼び出すプログラム**」パラメーターを参照してください。「経路指定項目の追加」パネルの「**比較データ: 値の比較**」パラメーターも参照してください。

## テキスト記述 (Text description)

この接続の使用目的を記録しておくための記述です (50 文字まで)。

## デバイス

リモート・システム用の装置記述の名前を指定します。指定できる値は以下のとおりです。

### \*LOC

この装置はシステムによって決められます。

### 装置名 (Device-name)

リモート・ロケーションと関連付けられる装置の名前を指定します。

## ローカル・ロケーション (Local location)

ローカル・ロケーション名を指定します。指定できる値は以下のとおりです。

### \*LOC

ローカル・ロケーション名はシステムによって決定されます。

### \*NETATR

システム・ネットワーク属性で指定した LCLLOCNAME 値が使用されます。

### ローカル・ロケーション名 (Local-location-name)

ローカル・ロケーションの名前を指定します。リモート・ロケーションとして特定のロケーション名を指示したい場合には、ローカル・ロケーション名を指定してください。ロケーション名は、DSPNETA コマンドを使用して調べることができます。

### モード (Mode)

セッションの制御に使用するモードを指定します。この名前は、共通プログラミング・インターフェース (CPI) - 通信 Mode\_Name と同じです。指定できる値は以下のとおりです。

#### \*NETATR

ネットワーク属性で指定されたモードが使用されます。

#### BLANK

8つの空白文字が使用されます。

#### モード名

リモート・ロケーションのモード名を指定します。

**注:** 通信セッションの伝送優先順位はモードによって決まるため、送られるメッセージの優先順位に基づいて、例えば MQMODE\_HI、MQMODE\_MED、MQMODE\_LOW のように、さまざまなモードを定義しておくとう便な場合があります (同じロケーションを指す複数の CSI を持つことができます)。

### リモート・ネットワーク ID (Remote network identifier)

リモート・ロケーションで使用されるリモート・ネットワーク ID を指定します。指定できる値は以下のとおりです。

#### \*LOC

リモート・ロケーションのリモート・ネットワーク ID が使用されます。

#### \*NETATR

ネットワーク属性で指定されたリモート・ネットワーク ID が使用されます。

#### \*NONE 値

このリモート・ネットワークには名前がありません。

#### リモート・ネットワーク ID (Remote-network-id)

リモート・ネットワーク ID を指定します。このネットワーク ID を調べるには、リモート・ロケーションで DSPNETA コマンドを使用してください。これはリモート・ロケーションの「ローカル・ネットワーク ID」です。

### Authority

このオブジェクトに対する特別な権限をもたないユーザー、許可リストに記載されていないユーザー、およびグループ・プロファイルでそのオブジェクトに対する特別な権限が指定されていないユーザーに対して与える権限を指定します。指定できる値は以下のとおりです。

#### \*LIBCRTAUT

このオブジェクトに関する共通権限は、指定されたライブラリーの CRTAUT パラメーターから得られます。この値は作成時に決定されます。オブジェクトの作成後にライブラリーに関する CRTAUT 値が変更された場合には、新しい値は既存のオブジェクトに影響を与えません。

#### \*CHANGE

変更権限を与えられたユーザーは、そのオブジェクトに対して基本機能を実行できます。ただし、ユーザーがオブジェクトを変更することはできません。変更権限は、オブジェクト操作権限およびすべてのデータ権限が与えられます。

#### \*ALL

ユーザーは、所有者に限定されている操作や権限リスト管理権限によって制御される操作を除き、すべての操作を実行できます。ユーザーは、そのオブジェクトの存在を制御したり、オブジェクトのセキュリティーを指定したり、オブジェクトを変更したり、オブジェクトに対する基本機能を実行したりすることができます。また、オブジェクトの所有権を変更できます。

#### \*USE

使用権限は、オブジェクト操作権限と読み取り権限を提供します。

#### \*EXCLUDE

除外権限は、そのユーザーがオブジェクトにアクセスできないようにします。

## 許可リスト (Authorization-list)

サイド情報として使用する権限の入った許可リストの名前を指定します。

### IBM i 開始される側 (受信側)

トランスポート・タイプ\*LU62 のメッセージ・チャネル・リンクの受信側を定義するには、CRTMQMCHL コマンドを使用してください。

「Connection name (接続名)」フィールドはブランクのままにし、また対応する詳細がチャネルの送信側と一致するようにしてください。詳細については、[チャネルの作成](#)を参照してください。

開始する側が受信側チャネルを開始できるように、開始される側のサブシステムに経路指定項目を追加してください。このサブシステムは、LU 6.2 セッションで使用する APPC デバイスを割り振るものでなければなりません。したがって、そのデバイスの有効な通信項目を保持している必要があります。経路指定項目は、メッセージ・チャネルの受信側を開始させるプログラムを呼び出します。

リンク上で通信セッションによって開始される側を定義するには、IBM i コマンド (例えば、ADDRTE) を使用してください。

開始される側のパネルを、[図 LU 6.2 通信設定パネル - Add Routing Entry](#) に示しています。

#### Add Routing Entry (ADDRTE)

Type choices, press Enter.

```
Subsystem description . . . . . QCMN      Name
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Routing entry sequence number . 1      1-9999
Comparison data:
Compare value . . . . . MQSERIES

Starting position . . . . . 37      1-80
Program to call . . . . . AMQCRC6B  Name, *RTGDTA
Library . . . . . QMAS400      Name, *LIBL, *CURLIB
Class . . . . . *SBSD      Name, *SBSD
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Maximum active routing steps . . *NOMAX 0-1000, *NOMAX
Storage pool identifier . . . . . 1      1-10
```

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys

図 36. LU 6.2 通信設定パネル - 開始される側

### サブシステム記述

この定義が置かれるサブシステムの名前です。経路指定項目に該当するサブシステム記述を表示および更新するには、IBM i WRKSBSD コマンドを使用してください。

### 経路指定項目のシーケンス番号 (Routing entry sequence number)

この通信定義を識別するための、サブシステム内で固有の番号です。値は 1 から 9999 の範囲内で指定できます。

### 比較データ: 値の比較 (Comparison data: Compare value)

[Figure 1](#) に示されているように、トランザクション・プログラムパラメーターによって、セッションが開始されたときに受信されるストリングと比較するテキスト・ストリング。文字ストリングは、送信側 CSI の「トランザクション・プログラム」フィールドから導出されます。

### 比較データ: 開始位置 (Comparison data: Starting position)

ストリング内の、比較が開始される文字位置です。

注: この開始位置フィールドは、ストリング内の比較を行うための文字位置であり、常に 37 です。

## 呼び出すプログラム (Program to call)

セッションを開始するために呼び出されるインバウンド・メッセージ・プログラムを実行する、プログラムの名前です。

プログラム AMQCRC6A は、デフォルト・キュー・マネージャーの場合に呼び出されます。このプログラムは、IBM MQ for IBM i に付属のプログラムであり、環境をセットアップしてから AMQCRS6A を呼び出します。

追加キュー・マネージャーの場合:

- 各キュー・マネージャーには、そのライブラリーに置かれた特定の LU 6.2 呼び出し可能プログラムが備わっています。このプログラムは AMQCRC6B と呼ばれ、キュー・マネージャーの作成時に自動的に生成されます。
- 各キュー・マネージャーには、追加する固有経路指定データを指定した特定の経路指定項目が必要です。この経路指定データは、要求側のシステムで指定する **トランザクション・プログラムの名前 (開始する側 (送信側) を参照)** と一致している必要があります。

例を以下の [LU 6.2 通信設定パネル - Display Routing Entries](#) に示します。

```
Display Routing Entries
System: MY400
Subsystem description: QCMN      Status: ACTIVE

Type options, press Enter.
5=Display details

Start
Opt Seq Nbr  Program      Library      Compare Value  Pos
10  *RTGDTA          'QZSCSRVR'   37
20  *RTGDTA          'QZRCSRVR'   37
30  *RTGDTA          'QZHQTRG'    37
50  *RTGDTA          'QVPPRINT'   37
60  *RTGDTA          'QNPSRVR'    37
70  *RTGDTA          'QNMAPPINGD' 37
80  QNMAREXECD  QSYS      'AREXECD'    37
90  AMQCRC6A     QMQMBW    'MQSERIES'   37
100 *RTGDTA       'QTFDWNLD'   37
150 *RTGDTA       'QMFRCVR'    37

F3=Exit  F9=Display all detailed descriptions  F12=Cancel
```

図 37. LU 6.2 通信設定パネル - 開始される側

[LU 6.2 通信設定パネル - Display Routing Entries](#) のシーケンス番号 90 はデフォルト・キュー・マネージャーを表しており、これによって、IBM MQ for IBM i の旧リリース (すなわち、V3R2、V3R6、V3R7、および V4R2) の構成との互換性を実現しています。これらのリリースは、1つのキュー・マネージャーだけを使用できます。シーケンス番号 92 および 94 は、ライブラリー QMALPHA および QMBETA を使用して作成された ALPHA および BETA という 2つの追加キュー・マネージャーを表します。

**注:** 異なる経路指定データを使用することにより、それぞれのキュー・マネージャーごとに別々の経路指定項目を指定できます。これらの項目によって、使用するクラスに応じてジョブ優先順位を変えることができます。

## Class

この経路指定項目を使用して開始されるステップに使用されるクラスの、名前とライブラリーです。このクラスは、経路指定ステップの実行環境の属性を定義し、ジョブ優先順位を指定します。該当のクラス項目を指定しなければなりません。既存クラスの表示または新規クラスの作成には、WRKCLS コマンドなどを使用してください。リモート LU 6.2 システムからの作業要求の管理についての詳細は、「*IBM i Programming: Work Management Guide*」に記載されています。

## ワーク・マネジメントに関する注意

AMQCRS6A ジョブは、[ワーク・マネジメント](#)に記載している標準的な IBM i ワーク・マネジメント機能を利用できません。このジョブは、他の IBM MQ ジョブと同じ方法では開始されないためです。LU 62 受信側ジョブの実行時プロパティを変更するには、以下のいずれかの変更を行います。

- AMQCRS6A ジョブの経路指定項目に指定されているクラス記述の変更
- 通信項目に対するジョブ記述の変更

通信ジョブの構成についての詳細は、「[IBM i Programming: Work Management Guide](#)」を参照してください。

## キュー・マネージャー・クラスターの構成

クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

### 始める前に

クラスターリングの概念の導入については、[クラスター](#)を参照してください。

キュー・マネージャー・クラスターを設計するときは、いくつかの決定を行う必要があります。[サンプル・クラスターおよびクラスターの設計](#)を参照してください。

### 関連タスク

[366 ページの『別のキュー・マネージャーへのクラスター・トピック定義の移動』](#)

トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

### 関連情報

[DELETE TOPIC](#)

## クラスターのコンポーネントの定義

クラスターは、キュー・マネージャー、クラスター・チャンネル、およびクラスター・キューで構成されます。クラスター・キューを定義したり、デフォルトのクラスター・オブジェクトのいくつかの側面を変更したりできます。個別のクラスター送信側チャンネルと伝送キューの関係に関して、さらに自動定義チャンネルに関しての構成情報や状況情報を取得できます。

各クラスター・コンポーネントの定義については、以下のサブトピックを参照してください。

### 関連タスク

[248 ページの『新規クラスターのセットアップ』](#)

クラスター例をセットアップするには、これらの手順に従ってください。クラスターを設定する方法は、TCP/IP、LU 6.2、および単一の伝送キューまたは複数の伝送キューを使用する場合で別々に説明されています。あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信することによって、クラスターの動作をテストします。

[260 ページの『クラスターにキュー・マネージャーを追加する』](#)

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、単一のクラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE を使用して転送されます。

### 関連情報

[クラスターのコンポーネント](#)

[クラスター・チャンネル](#)

[クラスター・トピックの定義](#)

## クラスター・キューの定義

クラスター・キューとは、クラスター・キュー・マネージャーでホストされ、同じクラスター内の別のキュー・マネージャーで使用できるキューです。クラスター・キューは、キューをホストするクラスター・キュー・マネージャーのローカル・キューとして定義します。キューが属するクラスターの名前を指定します。

次に示すのは、CLUSTER オプションを指定してクラスター・キューを定義する **runmqsc** コマンドの例です。

```
DEFINE QLOCAL(Q1) CLUSTER(SALES)
```

クラスター・キュー定義は、クラスター内の他のキュー・マネージャーに通知されます。クラスター内の他のキュー・マネージャーは、対応するリモート・キュー定義がなくても、クラスター・キューにメッセージを書き込むことができます。クラスター名前リストを使用して、クラスター・キューを複数のクラスターに通知できます。

キューが通知されると、クラスター内のキュー・マネージャーはそのキューにメッセージを書き込めるようになります。メッセージを書き込むときには、キュー・マネージャーが、フルリポジトリで、そのキューがホストされている場所を調べる必要があります。格納場所が分かったら、宛先情報をメッセージに追加して、クラスター伝送キューにメッセージを書き込みます。

 クラスター・キューは、IBM MQ for z/OS でのキュー共用グループのメンバーによって共有されるキューにすることができます。

## バインド中

クラスターを構築するときには、同じクラスター・キューをホストしている複数のキュー・マネージャーで構成されるクラスターを構築することができます。シーケンスに含まれるすべてのメッセージがキューの同一インスタンスに送信されるようにします。このとき、MQOPEN 呼び出しの MQOO\_BIND\_ON\_OPEN オプションを使用すると、一連のメッセージを特定のキューにバインドすることができます。

## クラスター伝送キュー

キュー・マネージャーは、同じクラスター内の他のキュー・マネージャーのメッセージを複数の伝送キューに格納することができます。複数のクラスター伝送キューにメッセージを格納するようキュー・マネージャーを構成する方法は2つあります。キュー・マネージャー属性 **DEFCLXQ** を CHANNEL に設定すると、SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE から、クラスター送信側チャンネルごとに異なるクラスター伝送キューが自動的に作成されます。CLCHNAME 伝送キュー・オプションを1つ以上のクラスター送信側チャンネルに一致するように設定すると、キュー・マネージャーは、一致しているチャンネルのメッセージを、そのチャンネルの伝送キューに格納できます。

 **重要:** IBM WebSphere MQ 7.5 より前のバージョンの製品からアップグレードされたキュー・マネージャーで専用の SYSTEM.CLUSTER.TRANSMIT.QUEUES を使用する場合は、SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE の **SHARE/NOSHARE** オプションが **SHARE** に設定されている必要があります。

別のキュー・マネージャー上のクラスター・キューへのメッセージは、クラスター伝送キューに配置されてから送信されます。クラスター送信側チャンネルは、メッセージをクラスター伝送キューから他のキュー・マネージャーのクラスター受信側チャンネルに転送します。デフォルトでは、システムで定義された1つのクラスター伝送キューが、他のクラスター・キュー・マネージャーに転送するすべてのメッセージを保持します。このキューを SYSTEM.CLUSTER.TRANSMIT.QUEUE といいます。クラスターの一部であるキュー・マネージャーは、このクラスター伝送キューに入っているメッセージを、同じクラスター内の他のキュー・マネージャーに送信できます。

単一の SYSTEM.CLUSTER.TRANSMIT.QUEUE キューの定義は、z/OS の場合を除いて、デフォルトではすべてのキュー・マネージャーに作成されます。 z/OS では、提供サンプル **CSQ4INSX** を使用して定義できます。

複数の伝送キューを使用して他のクラスター化されたキュー・マネージャーにメッセージを転送するように、キュー・マネージャーを構成できます。追加のクラスター伝送キューを手動で定義するか、またはキュー・マネージャーが自動的にキューを作成するように設定することもできます。

キューがキュー・マネージャーによって自動的に作成されるようにするには、キュー・マネージャー属性 `DEFCLXQ` を `SCTQ` から `CHANNEL` に変更します。その結果、キュー・マネージャーはクラスター送信側チャンネルが作成されるたびに別個のクラスター伝送キューを作成します。伝送キューは、`MODEL.QUEUE` から永続動的キューとして作成されます。各永続動的キューの名前は `SYSTEM.CLUSTER.TRANSMIT.ChannelName` です。各永続動的クラスターの伝送キューが関連付けられたクラスター送信側チャンネルの名前は、ローカル伝送キュー属性 `CLCHNAME` に設定されます。リモート・クラスター化されたキュー・マネージャーのメッセージは、`SYSTEM.CLUSTER.TRANSMIT.QUEUE` ではなく、関連付けられたクラスター送信側チャンネルの永続動的クラスターの伝送キューに置かれます。

クラスター伝送キューを手動で作成するには、`USAGE` 属性を `XMITQ` に設定し、`CLCHNAME` 属性を1つ以上のクラスター送信側チャンネルに解決する汎用チャンネル名に設定して、ローカル・キューを作成します。`ClusterChannelName` を参照してください。クラスター伝送キューを手動で作成する場合は、伝送キューを1つのクラスター送信側チャンネルに関連付けるかまたは複数のクラスター送信側チャンネルに関連付けるかを選択できます。`CLCHNAME` の属性は総称名です。これは、名前に複数のワイルドカード文字 "\*" を入れることができることを意味します。

キュー・マネージャーを完全リポジトリに接続するために手動で作成する初期クラスター送信側チャンネルを除いて、クラスター送信側チャンネルは自動的に作成されます。自動的に作成されるのは、クラスター・キュー・マネージャーに転送するメッセージがあるときです。クラスター送信側チャンネルは、宛先キュー・マネージャーで特定のクラスターに対するクラスター・メッセージを受信するクラスター受信側チャンネルの名前と同じ名前で作成されます。

クラスター受信側チャンネルの命名規則に従うと、`CLCHNAME` の総称値を定義して、さまざまなクラスター・メッセージをそれぞれの伝送キューにフィルタリングして転送することができます。例えば、`ClusterName.QmgrName` のクラスター受信側チャンネルの命名規則に従う場合、総称名 `ClusterName.*` は、異なるクラスターのメッセージを異なる伝送キューにフィルタリングします。伝送キューを手動で定義し、各伝送キューの `CLCHNAME` を `ClusterName.*` に設定してください。

クラスター伝送キューからクラスター送信側チャンネルへの関連付けを変更しても、即時には有効になりません。現在関連付けられていて、クラスター送信側チャンネルがサービス中の伝送キューには、クラスター送信側チャンネルが転送処理中のメッセージが含まれている可能性があります。キュー・マネージャーがクラスター送信側チャンネルの関連付けを別の伝送キューに変更できるのは、現在関連付けられている伝送キューにクラスター送信側チャンネルで処理中のメッセージがないときのみです。それに該当するのは、クラスター送信側チャンネルで処理するメッセージが伝送キューに残っていないとき、またはメッセージの処理が中断していてクラスター送信側チャンネルに"未完了"メッセージがないときです。この状態になると、クラスター送信側チャンネルの未処理メッセージはどれも新たに関連付けられた伝送キューに転送され、クラスター送信側チャンネルの関連付けが変更されます。

解決結果がクラスター伝送キューになるリモート・キュー定義を作成できます。この定義では、ローカル・キュー・マネージャーと同じクラスター内にキュー・マネージャー `QMX` があり、伝送キュー `QMX` はありません。

```
DEFINE QREMOTE(A) RNAME(B) RQMNAME(QMX)
```

キュー名の解決時は、デフォルト伝送キューよりクラスター伝送キューが優先されます。A に書き込まれたメッセージは、クラスター伝送キューに保管されてから、`QMX` 上のリモート・キュー B に送信されます。

また、キュー・マネージャーは、クラスターの一部ではない他のキュー・マネージャーと通信することもできます。分散キューイング環境の場合と同じように、他のキュー・マネージャーへのチャンネルと伝送キューを定義する必要があります。

**注:** アプリケーションでは、解決結果がクラスター伝送キューになるキューに書き込むようにし、クラスター伝送キューには直接書き込まないようにしてください。

## リモート・キューの自動定義

クラスター内のキュー・マネージャーは、クラスター内のリモート・キューに対して、リモート・キュー定義を必要としません。クラスター・キュー・マネージャーは、完全リポジトリからリモート・キューのロケーションを見つけます。メッセージにルーティング情報を追加し、それをクラスター伝送キューに書き込みます。IBM MQ は、リモート・キュー定義と同じ内容の定義を自動的に作成するため、メッセージを送信することができます。

自動作成されたリモート・キュー定義を修正したり削除したりすることはできません。しかし、CLUSINFO 属性を指定した DISPLAY QUEUE **runmqsc** コマンドを使用することにより、キュー・マネージャー上のローカル・キューをすべて表示できるだけでなく、リモート・キュー・マネージャー上のクラスター・キューも含めてすべてのクラスター・キューを表示できます。以下に例を示します。

```
DISPLAY QUEUE(*) CLUSINFO
```

### 関連情報

[クラスター・キュー](#)

[ClusterChannelName \(MQCHAR20\)](#)

### 自動定義クラスター送信側チャンネルの処理

最初の CLUSSDR 定義と CLUSRCVR 定義を作成してキュー・マネージャーをクラスターに導入した後、IBM MQ は、クラスター内の別のキュー・マネージャーにメッセージを移動する必要があるときに、他のクラスター送信側チャンネル定義を自動的に作成します。自動定義されたクラスター送信側チャンネルについては、情報を表示することはできますが、それを変更することはできません。その動作を変更するには、チャンネル自動定義出口を使用することができます。

### 始める前に

自動定義チャンネルの概要については、[自動定義されたクラスター送信側チャンネル](#)を参照してください。

### このタスクについて

自動定義クラスター送信側チャンネルは、クラスターにより必要に応じて作成されます。その後、通常の切断間隔規則を使ってシャットダウンされるまでアクティブ状態になります。

クラスター送信側チャンネル (CLUSSDR) は、アプリケーション・メッセージと内部クラスター管理メッセージの両方を移動するために自動定義できます。例えば、[パブリッシュ/サブスクライブ・クラスター](#) (クラスター・トピックが定義されているもの) では、部分リポジトリ間でチャンネルを定義して「プロキシ・サブスクリプション」状態の交換を許可できます。長期間にわたって必要ではない (非アクティブ) 場合、自動定義 CLUSSDR が部分リポジトリのクラスター情報のキャッシュから削除され、そのキュー・マネージャーに表示されなくなります。

**Multi** マルチプラットフォームでは、OAM (オブジェクト権限マネージャー) は、自動定義クラスター送信側チャンネルの存在を認識しません。自動定義クラスター送信側チャンネルに対してコマンド **start**、**stop**、**ping**、**reset**、または **resolve** を発行すると、OAM はコマンドを発行したユーザーが対応するクラスター受信側チャンネルに対して同じアクションを実行する権限があるかどうかを検査します。

**z/OS** z/OS では、自動定義クラスター送信側チャンネルを、他のチャンネルと同じように保護することができます。

### 手順

- 特定のクラスター・キュー・マネージャーの自動定義チャンネルに関する情報を表示します。

自動的に定義されたチャンネルは、DISPLAY CHANNEL **runmqsc** コマンドで表示することができません。自動定義チャンネルを調べるには、次のコマンドを使用します。

```
DISPLAY CLUSQMGR(qMgrName)
```

- 特定の CLUSRCVR の自動定義チャンネルの状況を表示します。

作成した CLUSRCVR チャンネル定義に対応する自動定義チャンネル CLUSSDR の状況を表示するには、次のコマンドを使用します。

```
DISPLAY CHSTATUS(channelname)
```

- チャンネル自動定義出口を使用して、自動定義チャンネルの動作を変更します。

クラスター送信側チャンネルまたはクラスター受信側チャンネルをユーザーの要求に合わせてカスタマイズするために、ユーザー出口プログラムを作成したい場合、IBM MQ のチャンネル自動定義出口を使用することができます。例えば、クラスター環境でチャンネル自動定義出口を使用すると、以下のような変更を行えます。

- 通信の定義 (SNA LU6.2 名) を変更する場合。
- 他の出口 (セキュリティー出口など) を追加または削除する場合。
- チャンネル出口の名前を変更する場合。

CLUSSDR チャンネル出口の名前は CLUSRCVR チャンネル定義から自動生成されるため、希望の名前にならない場合があります。異なるプラットフォーム上にチャンネルの両端がある場合は特にそう言えます。

出口名の形式は、プラットフォームによって異なります。以下に例を示します。

- **z/OS** z/OS プラットフォームでは、SCYEXIT (*security exit name*) パラメーターの形式は SCYEXIT('SECEXIT') になります。
- **Windows** Windows プラットフォームでは、SCYEXIT (セキュリティー出口名) パラメーターの形式は SCYEXIT('drive:\path\library (secexit)') です。

**注:** **z/OS** チャンネル自動定義出口がない場合、z/OS キュー・マネージャーは CLUSSDR チャンネル出口名をチャンネルの他方の端の CLUSRCVR チャンネル定義から派生するものにします。z/OS 以外に派生する名前から z/OS 出口名を付けるために、以下のアルゴリズムが使用されます。

- マルチプラットフォーム における出口名の一般的な形式は、*path/library(function)* です。
- *function* が存在する場合は、8 文字までが使用されます。
- 存在しない場合は、*library* に含まれる最大 8 文字が使用されます。

以下に例を示します。

- /var/mqm/exits/myExit.so(MsgExit) は MSGEXIT に変わります。
- /var/mqm/exits/myExit は MYEXIT に変わります。
- /var/mqm/exits/myExit.so(ExitLongName) は EXITLONG に変わります。

- IBM WebSphere MQ 7 より前のキュー・マネージャーの場合、**PROPCTL** 属性を値 NONE に設定します。

それぞれの自動定義クラスター送信側チャンネルは、対応するクラスター受信側チャンネルに基づいて定義されます。IBM MQ バージョン 7 より前のクラスター受信側チャンネルには **PROPCTL** 属性がないため、この属性は自動定義クラスター送信側チャンネルで COMPAT に設定されます。

クラスターで **PROPCTL** を使用して、IBM WebSphere MQ 7 以降のキュー・マネージャーからそれより前のバージョンの IBM MQ のキュー・マネージャーに送られるメッセージから RFH2 などのアプリケーション・ヘッダーを除去する必要がある場合は、**PROPCTL** を値 NONE に設定するチャンネル自動定義出口を作成する必要があります。

- チャンネル属性 LOCLADDR を使用して、アドレッシングの側面を制御します。
  - アウトバウンド (TCP) チャンネルが特定の IP アドレス、ポート、またはポートの範囲を使用できるようにするには、チャンネル属性 LOCLADDR を使用します。これは、複数のネットワーク・カードがあり、チャンネルでアウトバウンド通信に特定のネットワーク・カードを使用する場合に有益です。

- CLUSSDR チャンネルの仮想 IP アドレスを指定するには、手動で定義した CLUSSDR の LOCLADDR での IP アドレスを使用します。ポート範囲を指定するには、CLUSRCVR でのポート範囲を使用します。
- クラスターで LOCLADDR を使用して、アウトバウンド通信チャンネルを特定の IP アドレスにバインドする必要がある場合は、自動的に定義された CLUSSDR チャンネルに LOCLADDR 値を強制的に指定するために、チャンネル自動定義出口を作成することができます。また、手動で定義された CLUSSDR チャンネルにもこの値を指定する必要があります。
- クラスター内のすべてのキュー・マネージャーで、すべてのアウトバウンド通信に特定のポートまたはポートの範囲を使用するには、CLUSRCVR チャンネルの LOCLADDR にポート番号またはポート範囲を入力します。

**注:** すべてのキュー・マネージャーが同一サーバー上にない限り、CLUSRCVR チャンネルの LOCLADDR フィールドに IP アドレスを入力しないでください。LOCLADDR IP アドレスは、CLUSRCVR チャンネルを使用して接続するすべてのキュー・マネージャーの自動定義 CLUSSDR チャンネルに伝搬されます。

**Multi** マルチプラットフォームでは、ローカル・アドレスが定義されていないすべての送信側チャンネルで使用される、デフォルトのローカル・アドレス値を設定することができます。このデフォルト値を定義するには、キュー・マネージャーの開始前に MQ\_LCLADDR 環境変数を設定します。この値の形式は、MQSC 属性の LOCLADDR の形式と同じです。

## 関連情報

[ローカル・アドレス \(LOCLADDR\)](#)

## デフォルト・クラスター・オブジェクトの処理

デフォルト・チャンネル定義は、他のすべてのチャンネル定義と同様に、MQSC または PCF コマンドを実行することで変更できます。SYSTEM.CLUSTER.HISTORY.QUEUE を除き、デフォルト・キュー定義は変更しないでください。

これらのオブジェクトの完全なリストについては、[デフォルトのクラスターオブジェクト](#)を参照してください。以下のリストには、変更可能なオブジェクトしか含まれていません。

### SYSTEM.CLUSTER.HISTORY.QUEUE

クラスター内のそれぞれのキュー・マネージャーには、SYSTEM.CLUSTER.HISTORY.QUEUE というローカル・キューがあります。SYSTEM.CLUSTER.HISTORY.QUEUE は、サービス目的でクラスター状態情報の履歴を保管するために使用されます。

デフォルトのオブジェクト設定では、SYSTEM.CLUSTER.HISTORY.QUEUE は PUT (ENABLED) に設定されます。履歴収集を抑止するには、設定を PUT (DISABLED) に変更します。

### SYSTEM.CLUSTER.TRANSMIT.QUEUE

それぞれのキュー・マネージャーには、SYSTEM.CLUSTER.TRANSMIT.QUEUE というローカル・キューの定義があります。SYSTEM.CLUSTER.TRANSMIT.QUEUE は、クラスター内のすべてのキューおよびキュー・マネージャーに送信する、すべてのメッセージのためのデフォルト伝送キューです。キュー・マネージャー属性 DEFXTMQ **z/OS** (z/OS を除く) を変更することにより、各クラスター送信側チャンネルのデフォルト伝送キューを SYSTEM.CLUSTER.TRANSMIT.ChannelName に変更できます。SYSTEM.CLUSTER.TRANSMIT.QUEUE を削除することはできません。これは、使用されるデフォルト伝送キューが SYSTEM.CLUSTER.TRANSMIT.QUEUE であるのか、SYSTEM.CLUSTER.TRANSMIT.ChannelName であるのかを検証する許可検査の定義にも使用されます。

## 関連情報

[デフォルトのクラスター・オブジェクト](#)

## クラスター伝送キューとクラスター送信側チャンネルの操作

クラスター・キュー・マネージャー間のメッセージは、クラスター伝送キューに保管されてから、クラスター送信側チャンネルによって転送されます。どの時点でも、クラスター送信側チャンネルが関連付けられる伝送キューは 1 つです。チャンネルの構成を変更すると、そのチャンネルは次回開始したときに、別の伝送キューに切り替わる可能性があります。この切り替え処理は自動化されていて、トランザクションで実行されます。

クラスター送信側チャンネルが関連付けられている伝送キューを表示するには、以下の MQSC コマンドを実行します。

```
DISPLAY CHSTATUS(*) WHERE(CHLTYPE EQ CLUSSDR)
```

```
AMQ8417: Display Channel Status details.  
CHANNEL(TO.QM2)          CHLTYPE(CLUSSDR)  
CONNNAME(9.146.163.190(1416))  CURRENT  
RQMNAME(QM2)             STATUS(STOPPED)  
SUBSTATE( )              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

停止したクラスター送信側チャンネルの保存されたチャンネル状況に示される伝送キューは、チャンネルが再び開始すると変更される可能性があります。デフォルト伝送キューの選択プロセスについては、[242 ページの『クラスター送信側チャンネルによるデフォルト伝送キューの選択』](#)で説明しています。手動で定義した伝送キューの選択プロセスについては、[243 ページの『クラスター送信側チャンネルによる手動で定義された伝送キューの選択』](#)で説明しています。

すべてのクラスター送信側チャンネルは、開始時にその伝送キューとの関連付けを再チェックします。伝送キューの構成またはキュー・マネージャーのデフォルトが変更されている場合、チャンネルと伝送キューの関連付けが変更される可能性があります。構成変更の結果として、チャンネルが異なる伝送キューで再始動した場合、新しく関連付けられた伝送キューにメッセージを転送するプロセスが発生します。ある伝送キューから別の伝送キューへのクラスター送信側チャンネルの転送プロセスについては、[244 ページの『クラスター送信側チャンネルを異なる伝送キューに切り替えるプロセスの仕組み』](#)で説明しています。

クラスター送信側チャンネルの動作は、送信側およびサーバー・チャンネルとは異なります。これらのチャンネルは、チャンネル属性 **XMITQ** が変更されるまで、同じ伝送キューとの関連付けを維持します。送信側またはサーバー・チャンネルで伝送キュー属性を変更して、チャンネルを再始動しても、メッセージは古い伝送キューから新しい伝送キューには転送されません。

クラスター送信側チャンネルと送信者、またはサーバー・チャンネルの相違点の一つは、複数のクラスター送信側チャンネルがクラスター伝送キューを開くことができるという点ですが、通常の伝送キューをオープンできる送信側チャンネルまたはサーバー・チャンネルだけです。IBM WebSphere MQ 7.5 までは、クラスター接続は単一のクラスター伝送キュー **SYSTEM.CLUSTER.TRANSMIT.QUEUE** を共有していました。IBM WebSphere MQ 7.5 以降では、クラスター送信側チャンネルに伝送キューを共有させないことを選択できます。排他性は強制されません。これは、構成の結果です。クラスター内でメッセージが通るパスを構成して、そのメッセージが他のアプリケーション間を流れるメッセージと伝送キューやチャンネルを共有しないようにすることができます。[クラスター化: クラスター伝送キューの構成方法の計画および 297 ページの『クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』](#)を参照してください。

z/OS で **SYSTEM.CLUSTER.TRANSMIT.QUEUE** 以外の伝送キューを使用するようにクラスター送信チャンネルを構成するには、**CSQ6SYSP** マクロの操作モード (**OPMODE**) システム・パラメーターを使って、バージョン 8 の新機能を有効にする必要があります。

## クラスター送信側チャンネルによるデフォルト伝送キューの選択

クラスター伝送キューは、名前が **SYSTEM.CLUSTER.TRANSMIT** で始まるシステム・デフォルト・キュー、または手動で定義されたキューのいずれかです。クラスター送信側チャンネルは、2つの方法のいずれかでクラスター伝送キューに関連付けられます。一方の方法はデフォルト・クラスター伝送キュー・メカニズム、もう一方の方法は手動による構成です。

デフォルト・クラスター伝送キューは、キュー・マネージャー属性 **DEFCLXQ** として設定されます。その値は、**SCTQ** または **CHANNEL** です。新しいキュー・マネージャーおよびマイグレーションされたキュー・マネージャーは、**SCTQ** に設定されます。この値を **CHANNEL** に変更することができます。

**SCTQ** が設定されている場合、デフォルト・クラスター伝送キューは **SYSTEM.CLUSTER.TRANSMIT.QUEUE** となります。このキューは、すべてのクラスター送信側チャンネルがオープンできます。このキューをオープンしないクラスター送信側チャンネルは、手動で定義されたクラスター伝送キューに関連付けられていないものです。

CHANNEL が設定されている場合には、キュー・マネージャーがクラスター送信側チャンネルごとに個別の永続動的伝送キューを作成できます。各キューは `SYSTEM.CLUSTER.TRANSMIT.ChannelName` という名前で、モデル・キュー `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` から作成されます。手動で定義されたクラスター伝送キューに関連付けられていないクラスター送信側チャンネルは、それぞれ永続動的クラスター伝送キューに関連付けられます。このキューは、このクラスター送信側チャンネルが処理するクラスター宛先に別個のクラスター伝送キューが必要であるものの、キューが存在していない場合に、キュー・マネージャーによって作成されます。

一部のクラスター宛先は、手動で定義された伝送キューに関連付けられたクラスター送信側チャンネルによって処理できます。その他のクラスター宛先は、デフォルト・キューによって処理されます。クラスター送信側チャンネルと伝送キューの関連付けでは、常に、手動で定義された伝送キューがデフォルト伝送キューに優先されます。

クラスター伝送キューの優先順位については、243 ページの図 38 で説明されています。手動で定義されたクラスター伝送キューに関連付けられていないクラスター送信側チャンネルは、`CS.QM1` だけです。これが手動で定義された伝送キューに関連付けられない理由は、伝送キューの `CLCHNAME` 属性に設定されたチャンネル名の中に、`CS.QM1` と一致する名前がないためです。

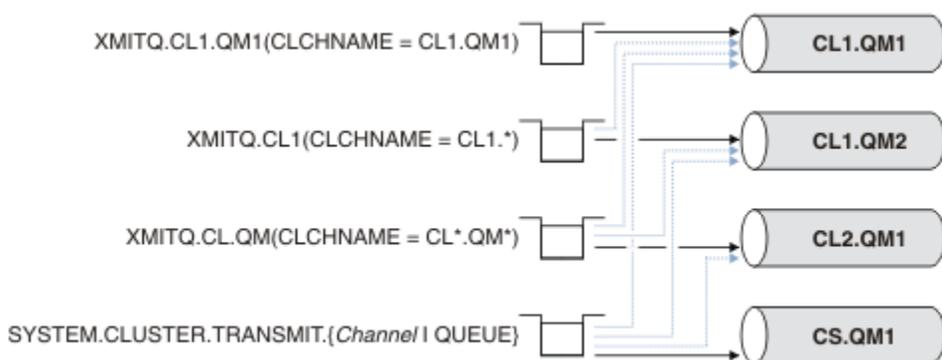


図 38. 伝送キューとクラスター送信側チャンネルの優先順位

## クラスター送信側チャンネルによる手動で定義された伝送キューの選択

手動で定義されたキューの伝送キュー属性 `USAGE` は `XMITQ` に設定され、クラスター・チャンネル名属性 `CLCHNAME` は特定のチャンネル名または総称チャンネル名に設定されています。

`CLCHNAME` キュー属性に設定された名前がクラスター送信側チャンネル名と一致する場合、そのチャンネルがキューに関連付けられます。この名前は、完全一致 (名前にワイルドカードが含まれていない場合)、または最適一致 (名前にワイルドカードが含まれている場合) のいずれかです。

複数の伝送キューの `CLCHNAME` 定義が同じクラスター送信側チャンネルに一致する場合、「定義がオーバーラップしている」と言います。あいまいさを解決するため、複数の一致の間には優先順位があります。完全一致が常に優先されます。243 ページの図 38 に、伝送キューとクラスター送信側チャンネルとの間の関連付けを示します。黒の矢印は実際の関連付けを表し、グレイの矢印は潜在的な関連付けを表します。伝送キューの優先順位については、243 ページの図 38 で説明されています。

### XMITQ.CL1.QM1

伝送キュー `XMITQ.CL1.QM1` の `CLCHNAME` 属性は `CL1.QM1` に設定されています。`CLCHNAME` 属性の定義である `CL1.QM1` はワイルドカードを取らず、他の伝送キューに定義されたワイルドカードに一致する他のすべての `CLCHNAME` 属性よりも優先されます。キュー・マネージャーは、`CL1.QM1` クラスター送信側チャンネルによって転送されるすべてのクラスター・メッセージを `XMITQ.CL1.QM1` 伝送キューに格納します。唯一の例外は、複数の伝送キューで `CLCHNAME` 属性が `CL1.QM1` に設定されている場合です。その場合、キュー・マネージャーは `CL1.QM1` クラスター送信側チャンネルのメッセージをそれらの伝送キューのいずれかに格納します。チャンネルが開始すると、キュー・マネージャーはキューを任意に選択します。再びチャンネルが開始して、別のキューが選択されることもあります。

## XMITQ.CL1

伝送キュー XMITQ.CL1 の **CLCHNAME** 属性は CL1.\* に設定されています。 **CLCHNAME** 属性の定義である CL1.\* は末尾に 1 つのワイルドカードを取り、これは CL1. で始まる任意のクラスター送信側チャンネルの名前に一致します。キュー・マネージャーは、CL1. で始まる名前を持つクラスター送信側チャンネルによって転送されるすべてのクラスター・メッセージを伝送キュー XMITQ.CL1 に格納します。ただし、キュー XMITQ.CL1.QM1 などのように、より具体的に一致する伝送キューがあれば、そのキューが使用されます。末尾に 1 つのワイルドカードがある定義は、具体性の点で、ワイルドカードがない定義よりも劣り、複数のワイルドカードがある定義や、ワイルドカードの後にさらに文字が続く定義よりも優れています。

## XMITQ.CL.QM

XMITQ.CL.QM は、**CLCHNAME** 属性が CL\*.QM\* に設定された伝送キューの名前です。CL\*.QM\* という定義には 2 つのワイルドカードがあり、それらは任意のクラスター送信側チャンネルの名前が CL. で始まり、その名前が QM を含むかまたはそれで終わることに一致します。ワイルドカードが 1 文字であるときの一致と比較すると、この一致は具体性の点で劣ります。

## SYSTEM.CLUSTER.TRANSMIT. channelName|QUEUE

伝送キューの **CLCHNAME** 属性にキュー・マネージャーが使用する予定のクラスター送信側チャンネルの名前と一致するものがない場合、キュー・マネージャーはデフォルトのクラスター伝送キューを使用します。デフォルトのクラスター伝送キューは、単一システム・クラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE、またはキュー・マネージャーが特定のクラスター送信側チャンネル SYSTEM.CLUSTER.TRANSMIT. channelName 用に作成したシステム・クラスター伝送キューのいずれかです。どちらがデフォルトであるかは、キュー・マネージャーの **DEFXMITQ** 属性の設定によって決まります。

**ヒント**：オーバーラップする定義が明らかに必要であるのでなければ、構成がわかりにくくて複雑にならないようにするために、そのような定義は使わないようにしてください。

## クラスター送信側チャンネルを異なる伝送キューに切り替えるプロセスの仕組み

クラスター送信側チャンネルとクラスター伝送キューとの関連付けを変更するには、随時、任意の伝送キューの **CLCHNAME** パラメーターを変更するか、キュー・マネージャー・パラメーター **DEFCLXQ** を変更します。すぐには何も起こりません。変更はチャンネルが開始した時にのみ行われます。開始時に、引き続き同じ伝送キューからメッセージを転送するかどうかをチェックします。次の 3 種類の変更によって、クラスター送信側チャンネルと伝送キューとの関連付けは変更されます。

1. クラスター送信側チャンネルが現在関連付けられている伝送キューの **CLCHNAME** パラメーターをより限定的でなくなるように再定義するか、またはブランクに再定義する。あるいは、チャンネルが停止されているときに、クラスター伝送キューを削除する。

他のクラスター伝送キューが、チャンネル名により一致するようになる可能性があります。あるいは、クラスター送信側チャンネルの名前と一致する伝送キューが他にない場合、関連付けはデフォルト伝送キューに戻る必要があります。

2. 他の任意のクラスター伝送キューの **CLCHNAME** パラメーターを再定義するか、クラスター伝送キューを追加する。

クラスター送信側チャンネルに現在関連付けられている伝送キューよりも、別の伝送キューの **CLCHNAME** パラメーターのほうがクラスター送信側チャンネルに一致するようになる可能性があります。クラスター送信側チャンネルが現在デフォルト・クラスター伝送キューに関連付けられている場合、手動で定義されたクラスター伝送キューに関連付けられることになる可能性があります。

3. クラスター送信側チャンネルが現在デフォルト・クラスター伝送キューに関連付けられている場合、**DEFCLXQ** キュー・マネージャー・パラメーターを変更する。

クラスター送信側チャンネルの関連付けが変更された場合、チャンネルは開始した時点で、その関連付けを新しい伝送キューに切り替えます。この切り替えの間、メッセージが失われないようにします。メッセージは、チャンネルによってリモート・キュー・マネージャーに転送される順序に従って、新しい伝送キューに転送されます。

**要確認:** クラスター内でメッセージを転送する場合と同様、メッセージをグループに書き出して、順番に送信する必要のあるメッセージが順番に送信されているかどうかを確認します。まれに、クラスター内でメッセージの順序が狂うことがあります。

切り替えプロセスは、以下のトランザクション・ステップに従って行われます。切り替えプロセスが中断された場合は、チャンネルが再始動した時点で、現行のトランザクション・ステップが再開されます。

#### ステップ 1 - 元の伝送キューのメッセージの処理

クラスター送信側チャンネルが関連付けられた新しい伝送キューは、他のクラスター送信側チャンネルと共有している可能性があります。クラスター送信側チャンネルへのメッセージは、引き続き元の伝送キューに配置されます。トランザクション切り替えプロセスにより、メッセージは元の伝送キューから新しい伝送キューに転送されます。クラスター送信側チャンネルは、メッセージを新しい伝送キューからクラスター受信側チャンネルに転送します。チャンネル状況には、まだ古い伝送キューに関連付けられたクラスター送信側チャンネルが示されます。

切り替えプロセスは、新しく到着したメッセージの転送も継続します。このステップは、切り替えプロセスによって転送される残りのメッセージの数がゼロに達するまで続きます。メッセージの数がゼロに達すると、プロシージャラーはステップ 2 に移ります。

ステップ 1 の間、チャンネルのディスク・アクティビティーは増加します。持続メッセージは、最初の伝送キューからコミットされ、2 番目の伝送キューに渡されます。このディスク・アクティビティーは、通常メッセージ転送の一環として、メッセージが伝送キューに配置され、そこから削除される際に、コミットされるメッセージに追加されます。概念的には、切り替えプロセス中に到着するメッセージはないため、この遷移は可能な限り短時間で行われるはずですが、メッセージが到着した場合は、切り替えプロセスによって処理されます。

#### ステップ 2 - 新しい伝送キューのメッセージの処理

クラスター送信側チャンネルの元の伝送キューに残っているメッセージがなくなると同時に、新しいメッセージは直接新しい伝送キューに配置されるようになります。チャンネル状況には、クラスター送信側チャンネルが新しい伝送キューに関連付けられていることが示されます。以下のメッセージがキュー・マネージャーのエラー・ログに書き込まれます:"AMQ7341ChannelName の伝送キューは QueueName です。"

### 複数のクラスター伝送キューとクラスター伝送キュー属性

クラスター・メッセージを別のキュー・マネージャーへ転送する場合、1つのクラスター伝送キューに格納するか、または複数のキューに格納するかを選択できます。1つのキューを使用する場合は、クラスター伝送キュー属性の1つのセットを設定および照会し、複数のキューを使用する場合は、複数のセットを設定および照会します。一部の属性では、複数のセットを設定すると有利です。例えば、キュー項目数を照会することによって、すべてのチャンネルではなく、1つのチャンネルまたはチャンネルのセットについて、転送を待機中のメッセージ数が分かります。その他の属性については、複数のセットを設定すると不利になります。例えば、すべてのクラスター伝送キューに同じアクセス権を構成することはしたくないはずですが、この理由から、アクセス権は特定のクラスター伝送キューのプロファイルではなく、常に `SYSTEM.CLUSTER.TRANSMIT.QUEUE` のプロファイルに対してチェックされます。より粒度の細かいセキュリティ検査を適用する必要がある場合には、[アクセス制御と複数のクラスター伝送キュー](#)を参照してください。

### 複数のクラスター送信側チャンネルと複数の伝送キュー

キュー・マネージャーは、メッセージをクラスター送信側チャンネルに転送する前に、クラスター伝送キューに保管します。そして、メッセージの宛先に接続されたクラスター送信側チャンネルを選択します。共通して同じ宛先に接続された複数のクラスター送信側チャンネルから選択できる場合もあります。宛先は、複数のクラスター送信側チャンネルによって単一のキュー・マネージャーに接続された同じ物理的キューである場合があります。宛先が、同じクラスター内の異なる複数のキュー・マネージャーでホストされた、同じキュー名を持つ多数の物理的キューである場合もあります。宛先に接続されたクラスター送信側チャンネルから選択できる場合には、ワークロード・バランシング・アルゴリズムがクラスター送信側チャンネルを選択します。この選択は、複数の要因に基づいて行われます。[クラスター・ワークロード管理アルゴリズム](#)を参照してください。

246 ページの図 39 では、CL1.QM1、CL1.QM2、および CS.QM1 は、いずれも同じ宛先に至る可能性のあるチャンネルです。例えば、CL1 内での Q1 を QM1 および QM2 に定義した場合、CL1.QM1 と CL1.QM2 の両

方が、2つの異なるキュー・マネージャー上にある同じ宛先 Q1 への経路を提供します。チャンネル CS.QM1 も CL1 内にある場合、Q1 へのメッセージはこのチャンネルも使用できます。CS.QM1 のクラスター・メンバーシップはクラスター名前リストで定義できます。チャンネル名がその構造にクラスター名を含めないのは、このためです。ワークロード・バランシング・パラメーターおよび送信側アプリケーションによっては、Q1 への一部のメッセージが伝送キュー XMITQ.CL1.QM1、XMITQ.CL1、および SYSTEM.CLUSTER.TRANSMIT.CS.QM1 のそれぞれに配置されることもあります。

メッセージ・トラフィックを分離して、同じ宛先へのメッセージが、それとは異なる宛先へのメッセージとキューやチャンネルを共有しないようにする場合には、トラフィックを複数の異なるクラスター送信側チャンネルに分割する方法をまず検討してから、特定のチャンネルのメッセージを別個の伝送キューに分離する方法を検討する必要があります。同じクラスターの同じキュー・マネージャー上にあるクラスター・キューは、一般に同じクラスター・チャンネルを共有します。複数のクラスター伝送キューを定義するだけでは、クラスター・メッセージ・トラフィックを異なるキューに分離するには不十分です。異なる宛先キューへのメッセージをそれぞれに異なるチャンネルに分離しない限り、メッセージは同じクラスター伝送キューを共有します。

メッセージが使用するチャンネルを分離する簡単な方法は、複数のクラスターを作成することです。各クラスターのあらゆるキュー・マネージャーには、1つのクラスター・キューだけを定義します。そうしてから、クラスターとキュー・マネージャーの組み合わせごとに異なるクラスター受信側チャンネルを定義すれば、各クラスター・キューへのメッセージがクラスター・チャンネルを他のクラスター・キューへのメッセージと共有することはありません。クラスター・チャンネルに個々の伝送キューを定義すると、送信側キュー・マネージャーは、各伝送キューに1つのクラスター・キューへのメッセージだけを保管します。例えば、2つのクラスター・キューにリソースを共有させないようにするには、これらのクラスター・キューを同じキュー・マネージャーでそれぞれに異なるクラスターに配置するか、あるいは同じクラスター内の異なるキュー・マネージャーに配置することができます。

クラスター伝送キューの選択は、ワークロード・バランシング・アルゴリズムには影響しません。ワークロード・バランシング・アルゴリズムが、メッセージを転送するクラスター送信側チャンネルを選択します。そして、そのチャンネルによって処理される伝送キューにメッセージを配置します。チャンネルを選択し直すためにワークロード・バランシング・アルゴリズムが必要になった場合(例えば、チャンネルが停止した場合)、アルゴリズムはメッセージを転送するための別のチャンネルを選択できる場合があります。別のチャンネルを選択し、その新しいチャンネルが別のクラスター伝送キューからのメッセージを転送する場合、ワークロード・バランシング・アルゴリズムはメッセージをその別の伝送キューに転送します。

246 ページの図 39 では、2つのクラスター送信側チャンネル CS.QM1 と CS.QM2 が、デフォルト・システム伝送キューに関連付けられています。ワークロード・バランシング・アルゴリズムがメッセージを SYSTEM.CLUSTER.TRANSMIT.QUEUE、または他のクラスター伝送キューに保管すると、そのメッセージの転送先クラスター送信側チャンネルの名前が、メッセージの相関 ID に保管されます。各チャンネルは、相関 ID がチャンネル名と一致するメッセージのみを転送します。

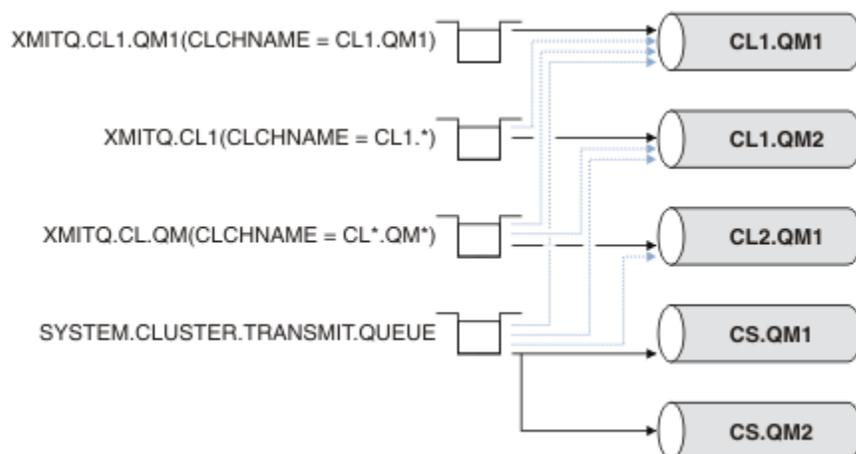


図 39. 複数のクラスター送信側チャンネル

CS.QM1 が停止すると、このクラスター送信側チャンネルの伝送キューに配置されたメッセージが検査されず。別のチャンネルで転送可能なメッセージは、ワークロード・バランシング・アルゴリズムによって再処

理されます。それらの関連 ID は、代替クラスター送信側チャンネル名に再設定されます。代替クラスター送信側チャンネルが CS.QM2 の場合、メッセージは SYSTEM.CLUSTER.TRANSMIT.QUEUE に残ります。代替チャンネルが CL1.QM1 の場合には、ワークロード・バランシング・アルゴリズムはメッセージを XMITQ.CL1.QM1 に転送します。クラスター送信側チャンネルが再始動すると、新しいメッセージおよび異なるクラスター送信側チャンネルのフラグが立てられなかったメッセージは、このチャンネルによって再び転送されます。

実行中のシステムで、伝送キューとクラスター送信側チャンネルの関連付けを変更することもできます。その場合、伝送キューの **CLCHNAME** パラメーターを変更するか、または **DEFCLXQ** キュー・マネージャー・パラメーターを変更します。変更によって影響を受けるチャンネルが再始動すると、そのチャンネルは伝送キュー切り替えプロセスを開始します (244 ページの『[クラスター送信側チャンネルを異なる伝送キューに切り替えるプロセスの仕組み](#)』を参照)。

チャンネルを再開すると、伝送キューを切り替えるプロセスが開始します。チャンネルが停止すると、ワークロードの再バランシング・プロセスが開始します。この 2 つのプロセスは、並列に実行できます。

単純なケースでは、クラスター送信側チャンネルが停止しても、キューのあらゆるメッセージを転送するクラスター送信側チャンネルを変更するための再バランシング・プロセスは発生しません。このケースは、メッセージを正しい宛先に転送できるクラスター送信側チャンネルが他にない場合です。メッセージをその宛先に転送する代替クラスター送信側チャンネルがないため、クラスター送信側チャンネルが停止した後も、メッセージには同じクラスター送信側チャンネルのフラグが立てられたままになります。切り替えが保留されている場合、チャンネルが開始すると、切り替えプロセスはメッセージを異なる宛先キューに移動します。その宛先キューで、メッセージは同じクラスター送信側チャンネルによって処理されます。

より複雑なケースでは、同じ宛先へのメッセージを複数のクラスター送信側チャンネルが処理できます。伝送キュー切り替えをトリガーするには、クラスター送信側チャンネルを停止してから再始動します。多くの場合、チャンネルを再始動するまでには、ワークロード・バランシング・アルゴリズムが、すでにメッセージを元の伝送キューから別のクラスター送信側チャンネルが処理する別の伝送キューに移動し終えています。別のクラスター送信側チャンネルで転送できないメッセージのみが、新しい伝送キューに転送されるために残ります。チャンネルが短時間で再始動された場合には、ワークロード・バランシング・アルゴリズムによって転送できなかったメッセージが残ることもあります。その場合、残されたメッセージのなかには、ワークロード・バランシング・プロセスによって切り替えられるものも、伝送キューを切り替えるプロセスによって切り替えられるものもあります。

## 関連概念

[535 ページの『ログのサイズの計算』](#)

キュー・マネージャーが必要とするログのサイズを見積もります。

## 関連タスク

[286 ページの『ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成』](#)

このタスクの手順に従って、ゲートウェイ・キュー・マネージャーを使用したオーバーラップするクラスターを構成します。このクラスターを、以降で説明するあるメッセージへのメッセージをクラスター内の他のアプリケーションへのメッセージから分離する例の開始点として使用します。

[262 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』](#)

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

[293 ページの『クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』](#)

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の 1 つのキュー・マネージャーに分離します。

[297 ページの『クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』](#)

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝

送キューやチャンネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

## 関連情報

[クラスター・チャンネル](#)

[クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離](#)

[クラスター化: クラスター伝送キューの構成方法の計画](#)

## 新規クラスターのセットアップ

クラスター例をセットアップするには、これらの手順に従ってください。クラスターを設定する方法は、TCP/IP、LU 6.2、および単一の伝送キューまたは複数の伝送キューを使用する場合で別々に説明されています。あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信することによって、クラスターの動作をテストします。

## 始める前に

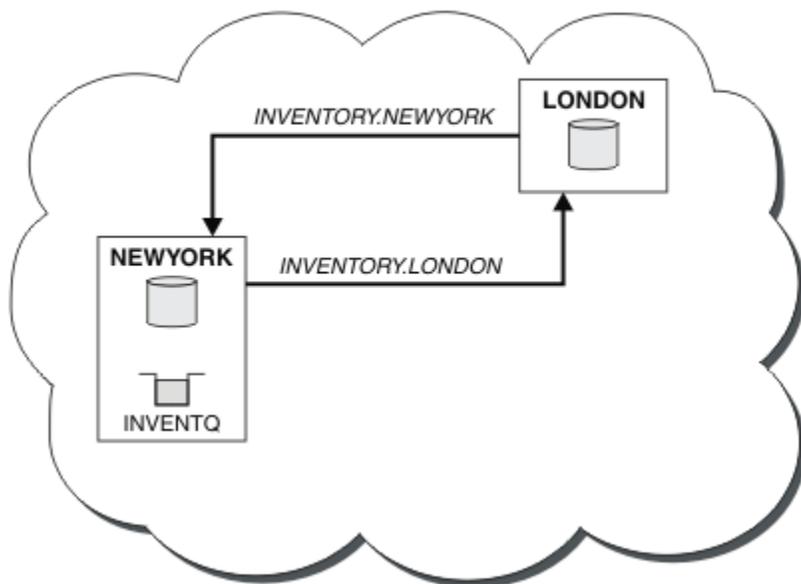
- 上記の手順に従う代わりに、IBM MQ Explorer に付属のウィザードの 1 つを使用して、このタスクで作成されるものと同様のクラスターを作成することができます。「キュー・マネージャー・クラスター」フォルダーを右クリックしてから、「新規」 > 「キュー・マネージャー・クラスター」をクリックして、ウィザードに示される指示に従います。
- クラスターのセットアップ手順を理解するために役立つ背景情報については、[237 ページの『クラスター・キューの定義』](#)、[クラスター・チャンネル](#)、および[リスナー](#)を参照してください。

## このタスクについて

これから、あるチェーン店で使用する新しい IBM MQ ネットワークをセットアップします。このチェーン店には、ロンドンとニューヨークにそれぞれ 1 つの支店があります。各支店のデータとアプリケーションは、各キュー・マネージャーを実行するシステムがそれぞれホストします。この 2 つのキュー・マネージャーの名前を LONDON および NEWYORK とします。在庫管理アプリケーションは、ニューヨーク支店のシステムで実行します。このシステムは、キュー・マネージャー NEWYORK に接続されます。このアプリケーションは、NEWYORK がホストしている INVENTQ というキューにメッセージが届くと起動されるようにします。キュー・マネージャー LONDON および NEWYORK を INVENTORY というクラスターにリンクさせて、これらのキュー・マネージャーが INVENTQ キューにメッセージを書き込むことができるようにします。

このクラスターは以下ようになります。

## INVENTORY



クラスター内の各キュー・マネージャーを、異なるクラスター伝送キューを使用してクラスター内の他のキュー・マネージャーにメッセージを送信するように構成できます。

クラスターをセットアップする手順は、トランスポート・プロトコル、伝送キューの数、またはプラットフォームによって少し異なります。3つの組み合わせから選択できます。どの組み合わせでも検査の手順は同じです。

INVENTORY は小さいクラスターです。クラスター概念を具体的に示すのに適しています。このクラスターについて理解する上で重要なのは、このクラスターには今後規模を拡大できる余地があるということです。

## 手順

- [249 ページの『キュー・マネージャーごとに1つの伝送キューを持ち、TCP/IP を使用するクラスターのセットアップ』](#)
- [252 ページの『キュー・マネージャーごとに複数の伝送キューを持ち TCP/IP を使用するクラスターのセットアップ』](#)
- [256 ページの『z/OS での LU 6.2 を使用するクラスターのセットアップ』](#)
- [258 ページの『クラスターの検査』](#)

## 関連タスク

[236 ページの『キュー・マネージャー・クラスターの構成』](#)

クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

## 関連情報

[クラスター](#)

[クラスター化と分散キューイングとの比較](#)

[クラスターのコンポーネント](#)

## キュー・マネージャーごとに1つの伝送キューを持ち、TCP/IP を使用するクラスターのセットアップ

これは、単純なクラスターのさまざまな構成について説明している3つのトピックのうちの1つです。

## 始める前に

作成されるクラスターの概要については、248 ページの『[新規クラスターのセットアップ](#)』を参照してください。

キュー・マネージャー属性 **DEFCLXQ** はデフォルト値 **SCTQ** のままにしておいてください。

## このタスクについて

マルチプラットフォームでトランスポート・プロトコル TCP/IP を使用するクラスターをセットアップするには、以下のステップに従います。 **z/OS** z/OS では、ステップ 251 ページの『4』でリスナーを定義するのではなく、750 ページの『[z/OS での TCP 接続の定義](#)』の説明に従って TCP/IP 接続をセットアップする必要があります。それ以外の場合、エラー・メッセージがキュー・マネージャーのエラー・ログではなくコンソールに書き込まれることを除いて、手順は z/OS と同じです。

## 手順

1. クラスターの編成およびクラスター名を決める。

この例では、LONDON、NEWYORK という 2 つのキュー・マネージャーをリンクさせてクラスターを構築します。2 つのキュー・マネージャーのみが含まれるクラスターは、分散キューイングを使用するネットワークに比べて利点はごくわずかです。しかし、スタートするには有効な方法であり、将来拡張できる余地も備えています。新しい支店をオープンするときに、新しいキュー・マネージャーをクラスターに簡単に追加できます。新しいキュー・マネージャーを追加しても、既存のネットワークが切断されるわけではありません (260 ページの『[クラスターにキュー・マネージャーを追加する](#)』を参照)。

当分の間、実行するアプリケーションは在庫管理アプリケーションのみとします。クラスター名は **INVENTORY** とします。

2. 完全リポジトリを保持するキュー・マネージャーを決定する。

クラスター内の 1 つ以上の (できれば 2 つの) キュー・マネージャーに完全リポジトリを格納するようにしてください。この例では、キュー・マネージャーは 2 つだけ (LONDON と NEWYORK) であるため、両方のキュー・マネージャーに完全リポジトリを格納します。

- a. 以降の手順については、任意の順序で実行できます。
- b. ステップを進めていくうちに、キュー・マネージャー・ログに警告メッセージが書き込まれることがあります。このメッセージは、まだ追加する必要がある定義が欠落していることの結果です。

Examples of the responses to the commands are shown in a box like this after each step in this task. These examples show the responses returned by IBM MQ for AIX. The responses vary on other platforms.

- c. 以降の手順に進む前に、上述のキュー・マネージャーが 2 つとも起動していることを確認してください。
3. キュー・マネージャー定義を変更してリポジトリ定義を追加する。

完全リポジトリを保持する各キュー・マネージャーで、次のように **ALTER QMGR** コマンドを使用して **REPOS** 属性を指定することにより、ローカル・キュー・マネージャー定義を変更します。

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

例えば、以下のように入力したとします。

```
a. runmqsc LONDON
```

## b. ALTER QMGR REPOS(INVENTORY)

LONDON が完全リポジトリに変更されます。

### 4. リスナーを定義します。

クラスター内のすべてのキュー・マネージャーに対して、他のキュー・マネージャーからのネットワーク要求を受け入れるリスナーを定義します。LONDON キュー・マネージャーに以下のコマンドを発行します。

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

CONTROL 属性は、キュー・マネージャーの開始および停止と同時に、このリスナーも開始および停止するようにします。

リスナーは定義と同時に開始されないため、最初は以下の MQSC コマンドを使用して手動で開始する必要があります。

```
START LISTENER(LONDON_LS)
```

クラスター内の他のすべてのキュー・マネージャーに対して同様のコマンドを発行します。ただし、各リスナーの名前は変更します。

リスナーで示すように、これらのリスナーを定義する方法はいくつかあります。

### 5. LONDON キュー・マネージャーの CLUSRCVR チャンネルを定義する。

クラスター内のすべてのキュー・マネージャーで、キュー・マネージャーがメッセージを受信できるクラスター受信側チャンネルを定義します。クラスター受信側チャンネル:CLUSRCVRを参照してください。CLUSRCVR チャンネルは、キュー・マネージャーの接続名を定義します。接続名はリポジトリに格納され、それを他のキュー・マネージャーが参照できます。CLUSTER キーワードは、キュー・マネージャーがクラスター内の他のキュー・マネージャーからメッセージを受信できることを示します。

この例では、チャンネル名を INVENTORY.LONDON とし、接続名 (CONNAME) は、キュー・マネージャーがあるマシンのネットワーク・アドレスである LONDON.CHSTORE.COM とします。ネットワーク・アドレスは、英数字の DNS ホスト名、または IPv4 小数点付き 10 進数形式の IP アドレスとして入力できます。例えば、192.0.2.0 です。あるいは IPv6 16 進数形式の場合は、例えば 2001:DB8:0204:acff:fe97:2c34:fde0:3485 です。ポート番号は指定されていないため、デフォルト・ポート (1414) が使用されます。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

### 6. NEWYORK キュー・マネージャーの CLUSRCVR チャンネルを定義する。

チャンネル・リスナーがデフォルト・ポート (通常は 1414) を使用し、クラスターに z/OS 上のキュー・マネージャーが含まれていない場合は、次のように CONNAME を省略できます。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

### 7. LONDON キュー・マネージャーに対して CLUSSDR チャンネルを定義します。

クラスターの各完全リポジトリ・キュー・マネージャーから、他の各完全リポジトリ・キュー・マネージャーへの CLUSSDR チャンネルを手動で定義します。クラスター送信側チャンネル:クラススドルを

参照します。この場合、キュー・マネージャーは2つだけであるため、両方のキュー・マネージャーに完全リポジトリを格納します。各キュー・マネージャーには、他方のキュー・マネージャーで定義された CLUSRCVR チャンネルを指す、手動で定義された CLUSSDR が必要です。CLUSSDR 定義で指定するチャンネル名は、対応する CLUSRCVR 定義でのチャンネル名と一致しなければなりません。キュー・マネージャーに、同じクラスターのクラスター受信側チャンネルとクラスター送信側チャンネル両方の定義が設定されると、クラスター送信側チャンネルが開始します。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

8. NEWYORK キュー・マネージャーに対して CLUSSDR チャンネルを定義します。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. クラスター・キュー INVENTQ を定義する。

キーワード CLUSTER を指定して、キュー・マネージャー NEWYORK に INVENTQ というキューを定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

CLUSTER というキーワードによって、クラスターはキューを認識できるようになります。キューは、定義されるとすぐに、クラスター内の他のキュー・マネージャーでも使用できるようになります。このキューにメッセージを送信する際、キュー・マネージャーは、リモート・キュー定義を作成する必要はありません。

すべての定義が完了しました。すべてのプラットフォームで、各キュー・マネージャー上でリスナー・プログラムを開始します。リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合にクラスター受信側チャンネルを始動します。

## 次のタスク

これで、[クラスターを検証する準備ができました](#)。

### 関連タスク

[252 ページの『キュー・マネージャーごとに複数の伝送キューを持ち TCP/IP を使用するクラスターのセットアップ』](#)

これは、単純なクラスターのさまざまな構成について説明している3つのトピックのうちの1つです。

[256 ページの『z/OSでのLU 6.2を使用するクラスターのセットアップ』](#)

これは、単純なクラスターのさまざまな構成を説明するツリー・トピックの1つです。

## キュー・マネージャーごとに複数の伝送キューを持ち TCP/IP を使用するクラスターのセットアップ

これは、単純なクラスターのさまざまな構成について説明している3つのトピックのうちの1つです。

## 始める前に

作成されるクラスターの概要については、248 ページの『[新規クラスターのセットアップ](#)』を参照してください。

## このタスクについて

マルチプラットフォームでトランスポート・プロトコル TCP/IP を使用するクラスターをセットアップするには、以下のステップに従います。リポジトリ・キュー・マネージャーは、リポジトリ・キュー・マネージャー同士およびクラスター内の他のキュー・マネージャーとの間でメッセージを送信する際に、別個のクラスター伝送キューを使用するように構成されています。別々の伝送キューを使用するキュー・マネージャーをクラスターに追加する場合は、262 ページの『[クラスターへのキュー・マネージャーの追加: 別個の伝送キュー](#)』に説明されている手順に従ってください。

## 手順

1. クラスターの編成およびクラスター名を決める。

この例では、LONDON、NEWYORK という 2 つのキュー・マネージャーをリンクさせてクラスターを構築します。2 つのキュー・マネージャーのみが含まれるクラスターは、分散キューイングを使用するネットワークに比べて利点はごくわずかです。しかし、スタートするには有効な方法であり、将来拡張できる余地も備えています。新しい支店をオープンするときに、新しいキュー・マネージャーをクラスターに簡単に追加できます。新しいキュー・マネージャーを追加しても、既存のネットワークが切断されるわけではありません (260 ページの『[クラスターにキュー・マネージャーを追加する](#)』を参照)。

当分の間、実行するアプリケーションは在庫管理アプリケーションのみとします。クラスター名は INVENTORY とします。

2. 完全リポジトリを保持するキュー・マネージャーを決定する。

クラスター内の 1 つ以上の (できれば 2 つの) キュー・マネージャーに完全リポジトリを格納するようにしてください。この例では、キュー・マネージャーは 2 つだけ (LONDON と NEWYORK) であるため、両方のキュー・マネージャーに完全リポジトリを格納します。

- a. 以降の手順については、任意の順序で実行できます。
- b. ステップを進めていくうちに、キュー・マネージャー・ログに警告メッセージが書き込まれることがあります。このメッセージは、まだ追加する必要がある定義が欠落していることの結果です。

Examples of the responses to the commands are shown in a box like this after each step in this task. These examples show the responses returned by IBM MQ for AIX. The responses vary on other platforms.

- c. 以降の手順に進む前に、上述のキュー・マネージャーが 2 つとも起動していることを確認してください。
3. キュー・マネージャー定義を変更してリポジトリ定義を追加する。

完全リポジトリを保持する各キュー・マネージャーで、次のように ALTER QMGR コマンドを使用して REPOS 属性を指定することにより、ローカル・キュー・マネージャー定義を変更します。

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

例えば、以下のように入力したとします。

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON が完全リポジトリに変更されます。

4. キュー・マネージャー定義を変更して、宛先ごとに別々のクラスター伝送キューを作成する。

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

クラスターに追加するキュー・マネージャーのそれぞれで、個別の伝送キューを使用するかどうかを決定します。260 ページの『[クラスターにキュー・マネージャーを追加する](#)』および 262 ページの『[クラスターへのキュー・マネージャーの追加: 別個の伝送キュー](#)』のトピックを参照してください。

5. リスナーを定義します。

クラスター内のすべてのキュー・マネージャーに対して、他のキュー・マネージャーからのネットワーク要求を受け入れるリスナーを定義します。LONDON キュー・マネージャーに以下のコマンドを発行します。

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

CONTROL 属性は、キュー・マネージャーの開始および停止と同時に、このリスナーも開始および停止するようにします。

リスナーは定義と同時に開始されないため、最初は以下の MQSC コマンドを使用して手動で開始する必要があります。

```
START LISTENER(LONDON_LS)
```

クラスター内の他のすべてのキュー・マネージャーに対して同様のコマンドを発行します。ただし、各リスナーの名前は変更します。

[リスナー](#)で示すように、これらのリスナーを定義する方法はいくつかあります。

6. LONDON キュー・マネージャーの CLUSRCVR チャンネルを定義する。

クラスター内のすべてのキュー・マネージャーで、キュー・マネージャーがメッセージを受信できるクラスター受信側チャンネルを定義します。[クラスター受信側チャンネル:CLUSRCVR](#) を参照してください。CLUSRCVR チャンネルは、キュー・マネージャーの接続名を定義します。接続名はリポジトリに格納され、それを他のキュー・マネージャーが参照できます。CLUSTER キーワードは、キュー・マネージャーがクラスター内の他のキュー・マネージャーからメッセージを受信できることを示します。

この例では、チャンネル名を INVENTORY.LONDON とし、接続名 (CONNAME) は、キュー・マネージャーがあるマシンのネットワーク・アドレスである LONDON.CHSTORE.COM とします。ネットワーク・アドレスは、英数字の DNS ホスト名、または IPv4 小数点付き 10 進数形式の IP アドレスとして入力できます。例えば、192.0.2.0 です。あるいは IPv6 16 進数形式の場合は、例えば 2001:DB8:0204:acff:fe97:2c34:fde0:3485 です。ポート番号は指定されていないため、デフォルト・ポート (1414) が使用されます。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

7. NEWYORK キュー・マネージャーの CLUSRCVR チャンネルを定義する。

チャンネル・リスナーがデフォルト・ポート (通常は 1414) を使用し、クラスターに z/OS 上のキュー・マネージャーが含まれていない場合は、次のように CONNAME を省略できます。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

## 8. LONDON キュー・マネージャーに対して CLUSSDR チャンネルを定義します。

クラスターの各完全リポジトリ・キュー・マネージャーから、他の各完全リポジトリ・キュー・マネージャーへの CLUSSDR チャンネルを手動で定義します。[クラスター送信側チャンネル:クラスドル](#)を参照します。この場合、キュー・マネージャーは2つだけであるため、両方のキュー・マネージャーに完全リポジトリを格納します。各キュー・マネージャーには、他方のキュー・マネージャーで定義された CLUSRCVR チャンネルを指す、手動で定義された CLUSSDR が必要です。CLUSSDR 定義で指定するチャンネル名は、対応する CLUSRCVR 定義でのチャンネル名と一致しなければなりません。キュー・マネージャーに、同じクラスターのクラスター受信側チャンネルとクラスター送信側チャンネル両方の定義が設定されると、クラスター送信側チャンネルが開始します。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

## 9. NEWYORK キュー・マネージャーに対して CLUSSDR チャンネルを定義します。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

## 10. クラスター・キュー INVENTQ を定義する。

キーワード CLUSTER を指定して、キュー・マネージャー NEWYORK に INVENTQ というキューを定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

CLUSTER というキーワードによって、クラスターはキューを認識できるようになります。キューは、定義されるとすぐに、クラスター内の他のキュー・マネージャーでも使用できるようになります。このキューにメッセージを送信する際、キュー・マネージャーは、リモート・キュー定義を作成する必要はありません。

すべての定義が完了しました。すべてのプラットフォームで、各キュー・マネージャー上でリスナー・プログラムを開始します。リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合にクラスター受信側チャンネルを始動します。

## 次のタスク

これで、[クラスターを検証する準備ができました](#)。

### 関連タスク

[249 ページの『キュー・マネージャーごとに1つの伝送キューを持ち、TCP/IP を使用するクラスターのセットアップ』](#)

これは、単純なクラスターのさまざまな構成について説明している3つのトピックのうちの1つです。

[256 ページの『z/OSでのLU 6.2を使用するクラスターのセットアップ』](#)

これは、単純なクラスターのさまざまな構成を説明するツリー・トピックの1つです。

## z/OS での LU 6.2 を使用するクラスターのセットアップ

これは、単純なクラスターのさまざまな構成を説明するツリー・トピックの 1 つです。

### 始める前に

作成されるクラスターの概要については、[248 ページの『新規クラスターのセットアップ』](#)を参照してください。

### 手順

1. クラスターの編成およびクラスター名を決める。

この例では、LONDON、NEWYORK という 2 つのキュー・マネージャーをリンクさせてクラスターを構築します。2 つのキュー・マネージャーのみが含まれるクラスターは、分散キューイングを使用するネットワークに比べて利点はごくわずかです。しかし、スタートするには有効な方法であり、将来拡張できる余地も備えています。新しい支店をオープンするときに、新しいキュー・マネージャーをクラスターに簡単に追加できます。新しいキュー・マネージャーを追加しても、既存のネットワークが切断されるわけではありません ([260 ページの『クラスターにキュー・マネージャーを追加する』](#)を参照)。

当分の間、実行するアプリケーションは在庫管理アプリケーションのみとします。クラスター名は INVENTORY とします。

2. 完全リポジトリを保持するキュー・マネージャーを決定する。

クラスター内の 1 つ以上の (できれば 2 つの) キュー・マネージャーに完全リポジトリを格納するようにしてください。この例では、キュー・マネージャーは 2 つだけ (LONDON と NEWYORK) であるため、両方のキュー・マネージャーに完全リポジトリを格納します。

- a. 以降の手順については、任意の順序で実行できます。
  - b. ステップを進めていくうちに、z/OS システム・コンソールに警告メッセージが書き込まれることがあります。このメッセージは、まだ追加する必要がある定義が欠落していることの結果です。
  - c. 以降の手順に進む前に、上述のキュー・マネージャーが 2 つとも起動していることを確認してください。
3. キュー・マネージャー定義を変更してリポジトリ定義を追加する。

完全リポジトリを保持する各キュー・マネージャーで、次のように ALTER QMGR コマンドを使用して REPOS 属性を指定することにより、ローカル・キュー・マネージャー定義を変更します。

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: IBM MQ queue manager changed.
```

例えば、以下のように入力したとします。

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON が完全リポジトリに変更されます。

4. リスナーを定義します。

 z/OS のチャンネル・イニシエーターおよび [753 ページの『LU 6.2 での受信』](#)を参照してください。

リスナーは定義と同時に開始されないため、最初は以下の MQSC コマンドを使用して手動で開始する必要があります。

```
START LISTENER(LONDON_LS)
```

クラスター内の他のすべてのキュー・マネージャーに対して同様のコマンドを発行します。ただし、各リスナーの名前は変更します。

#### 5. LONDON キュー・マネージャーの CLUSRCVR チャンネルを定義する。

クラスター内のすべてのキュー・マネージャーで、キュー・マネージャーがメッセージを受信できるクラスター受信側チャンネルを定義します。[クラスター受信側チャンネル:CLUSRCVR](#)を参照してください。CLUSRCVR チャンネルは、キュー・マネージャーの接続名を定義します。接続名はリポジトリに格納され、それを他のキュー・マネージャーが参照できます。CLUSTER キーワードは、キュー・マネージャーがクラスター内の他のキュー・マネージャーからメッセージを受信できることを示します。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
AMQ8014: WebSphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

#### 6. NEWYORK キュー・マネージャーの CLUSRCVR チャンネルを定義する。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager NEWYORK')
```

#### 7. LONDON キュー・マネージャーに対して CLUSSDR チャンネルを定義します。

クラスターの各完全リポジトリ・キュー・マネージャーから、他の各完全リポジトリ・キュー・マネージャーへの CLUSSDR チャンネルを手動で定義します。[クラスター送信側チャンネル:クラスドル](#)を参照します。この場合、キュー・マネージャーは2つだけであるため、両方のキュー・マネージャーに完全リポジトリを格納します。各キュー・マネージャーには、他方のキュー・マネージャーで定義された CLUSRCVR チャンネルを指す、手動で定義された CLUSSDR が必要です。CLUSSDR 定義で指定するチャンネル名は、対応する CLUSRCVR 定義でのチャンネル名と一致しなければなりません。キュー・マネージャーに、同じクラスターのクラスター受信側チャンネルとクラスター送信側チャンネル両方の定義が設定されると、クラスター送信側チャンネルが開始します。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(CPIC) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: WebSphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

#### 8. NEWYORK キュー・マネージャーに対して CLUSSDR チャンネルを定義します。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from NEWYORK to repository at LONDON')
```

#### 9. クラスター・キュー INVENTQ を定義する。

キーワード CLUSTER を指定して、キュー・マネージャー NEWYORK に INVENTQ というキューを定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: WebSphere MQ queue created.
```

CLUSTER というキーワードによって、クラスターはキューを認識できるようになります。キューは、定義されるとすぐに、クラスター内の他のキュー・マネージャーでも使用できるようになります。このキューにメッセージを送信する際、キュー・マネージャーは、リモート・キュー定義を作成する必要はありません。

すべての定義が完了しました。すべてのプラットフォームで、各キュー・マネージャー上でリスナー・プログラムを開始します。リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合にクラスター受信側チャンネルを始動します。

## 次のタスク

これで、[クラスターを検証する準備ができました](#)。

### 関連タスク

[249 ページの『キュー・マネージャーごとに1つの伝送キューを持ち、TCP/IPを使用するクラスターのセットアップ』](#)

これは、単純なクラスターのさまざまな構成について説明している3つのトピックのうちの1つです。

[252 ページの『キュー・マネージャーごとに複数の伝送キューを持ち TCP/IPを使用するクラスターのセットアップ』](#)

これは、単純なクラスターのさまざまな構成について説明している3つのトピックのうちの1つです。

## クラスターの検査

ピア・トピックでは、単純なクラスターの3つの異なる構成について説明します。このトピックでは、クラスターを検査する方法について説明します。

## 始める前に

このトピックでは、以下のいずれかのタスクを使用して作成したクラスターを検証することを想定しています。

- [249 ページの『キュー・マネージャーごとに1つの伝送キューを持ち、TCP/IPを使用するクラスターのセットアップ』](#).
- [252 ページの『キュー・マネージャーごとに複数の伝送キューを持ち TCP/IPを使用するクラスターのセットアップ』](#).
- [256 ページの『z/OSでの LU 6.2を使用するクラスターのセットアップ』](#).

作成されたクラスターの概要については、[248 ページの『新規クラスターのセットアップ』](#)を参照してください。

## このタスクについて

クラスターの検査は、以下の方法から1つ以上を使って行うことができます。

1. クラスター属性およびチャンネル属性を表示する管理コマンドを実行する。
2. クラスター・キューに対してメッセージを送受信するサンプル・プログラムを実行する。
3. クラスター・キューに要求メッセージを送信し、クラスター化されていない応答キューに応答メッセージで応答する独自のプログラムを作成する。

## 手順

DISPLAY **runmqsc** コマンドを実行して、クラスターを検査する。

表示される応答は、以下の手順の応答のようになります。

1. NEWYORK キュー・マネージャーから、**DISPLAY CLUSQMGR** コマンドを実行する。

```
dis clusqmgr(*)
```

```
1 : dis clusqmgr(*)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(NEWYORK)      CLUSTER(INVENTORY)
CHANNEL(INVENTORY.NEWYORK)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(LONDON)      CLUSTER(INVENTORY)
CHANNEL(INVENTORY.LONDON)
```

2. NEWYORK キュー・マネージャーから、**DISPLAY CHANNEL STATUS** コマンドを実行する。

```
dis chstatus(*)
```

```
1 : dis chstatus(*)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.NEWYORK) XMITQ( )
CONNAME(192.0.2.0)        CURRENT
CHLTYPE(CLUSRCVR)        STATUS(RUNNING)
RQMNAME(LONDON)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.LONDON) XMITQ(SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON)
CONNAME(192.0.2.1)        CURRENT
CHLTYPE(CLUSSDR)         STATUS(RUNNING)
RQMNAME(LONDON)
```

**amqsput** を使用して、2つのキュー・マネージャーの間でメッセージを送信します。

3. LONDON で、コマンド **amqsput INVENTQ LONDON** を実行します。

何らかのメッセージを入力して、ブランク行を 1 行追加します。

4. NEWYORK で、コマンド **amqsget INVENTQ NEWYORK** を実行します。

これで、LONDON で入力したメッセージが表示されるようになりました。プログラムは 15 秒後に終了します。

独自のプログラムを使用して、2つのキュー・マネージャーの間でメッセージを送信する。

次の手順では、キュー・マネージャー LONDON から NEWYORK の INVENTQ キューにメッセージを書き込み、そのメッセージに対する応答を LONDON\_reply というキューに受信します。

5. LONDON で、クラスター・キューにメッセージを書き込む。

- a) LONDON\_reply という名前のローカル・キューを定義します。
- b) MQOPEN オプションを MQOO\_OUTPUT に設定します。
- c) MQOPEN 呼び出しを発行して、キュー INVENTQ をオープンします。
- d) メッセージ記述子の ReplyToQ 名を LONDON\_reply に設定します。
- e) MQPUT 呼び出しを発行して、メッセージを書き込みます。
- f) メッセージをコミットします。

6. NEWYORK で、クラスター・キューにあるメッセージを受信して、応答キューに応答を書き込む。

- a) MQOPEN オプションを MQOO\_BROWSE に設定します。
- b) MQOPEN 呼び出しを発行して、キュー INVENTQ をオープンします。
- c) MQGET 呼び出しを発行して、INVENTQ からメッセージを取得します。

- d) メッセージ記述子から `ReplyToQ` 名を検索します。
  - e) `ReplyToQ` 名をオブジェクト記述子の `ObjectName` フィールドに入れます。
  - f) `MQOPEN` オプションを `MQOO_OUTPUT` に設定します。
  - g) `MQOPEN` 呼び出しを発行して、キュー・マネージャー `LONDON` で `LONDON_reply` を開きます。
  - h) `MQPUT` 呼び出しを発行して、メッセージを `LONDON_reply` に書き込みます。
7. `LONDON` で、応答を受け取ります。
- a) `MQOPEN` オプションを `MQOO_BROWSE` に設定します。
  - b) `MQOPEN` 呼び出しを発行して、キュー `LONDON_reply` をオープンします。
  - c) `MQGET` 呼び出しを発行して、`LONDON_reply` からメッセージを取得します。

## クラスターにキュー・マネージャーを追加する

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、単一のクラスター伝送キュー `SYSTEM.CLUSTER.TRANSMIT.QUEUE` を使用して転送されます。

### 始める前に

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも1つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

シナリオ

- 248 ページの『[新規クラスターのセットアップ](#)』での説明に従って、`INVENTORY` クラスターを設定しました。このクラスターは、`LONDON`、`NEWYORK` という2つのキュー・マネージャーで構成されています。そして、両方のキュー・マネージャーに完全リポジトリが格納されています。
- キュー・マネージャー `PARIS` は、最初のプライマリー・インストールが所有しています。所有関係が異なる場合は、`setmqenv` コマンドを実行して、`PARIS` が属するインストールのコマンド環境をセットアップしてください。
- TCP 接続は、3つすべてのシステム間で存在します。キュー・マネージャーは、キュー・マネージャーの制御下で開始される TCP リスナーを使用して構成されます。

### このタスクについて

1. その後、このチェーン店ではパリに新しい支店をオープンすることになったため、このクラスターに `PARIS` というキュー・マネージャーの追加が必要になりました。
2. キュー・マネージャー `PARIS` は、`INVENTQ` キューにメッセージを書き込むことにより、ニューヨーク支店のシステムで実行されているアプリケーションに新しい在庫情報を送信するものとします。

以下の手順に従って、クラスターにキュー・マネージャーを追加します。

### 手順

1. `PARIS` が最初に参照する完全リポジトリを決める。

クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリを参照しなければなりません。キュー・マネージャーは、クラスターに関する情報を完全リポジトリから収集し、自分用の部分リポジトリを作成します。いずれかのリポジトリを完全リポジトリとして選択してください。クラスターに新しいキュー・マネージャーを追加すると、そのキュー・マネージャーは他方のリポジトリもすぐに認識できるようになります。キュー・マネージャーに加えられた変更に関する情報は2つのリポジトリに直接送信されます。ここでは、(単に地理的な理由により) キュー・マネージャー `PARIS` をキュー・マネージャー `LONDON` にリンクします。

**注:** キュー・マネージャー `PARIS` の開始後は、残りの手順は任意の順序で実行できます。

## 2. キュー・マネージャー PARIS に CLUSRCVR チャンネルを定義する。

クラスター内の各キュー・マネージャーにクラスター受信側チャンネルを定義する必要があります。このチャンネルにより、キュー・マネージャーはメッセージを受信することができます。キュー・マネージャー PARIS には、以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

クラスター受信側チャンネルは、このキュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態になったことを通知します。他のキュー・マネージャーでは、クラスター受信側チャンネル INVENTORY.PARIS の送信側用の定義を作成しないでください。他の定義は、必要な場合に自動的に行われます。[クラスター・チャンネル](#)を参照します。

## 3. z/OS

IBM MQ for z/OS でチャンネル・イニシエーターを開始します。

## 4. キュー・マネージャー PARIS に CLUSSDR チャンネルを定義する。

完全リポジトリではないキュー・マネージャーをクラスターに追加する際は、完全リポジトリとの初期接続を行うためのクラスター送信側チャンネルを1つだけ定義します。[クラスター送信側チャンネル: クラスドル](#)を参照します。

PARIS に、LONDON.CHSTORE.COM というネットワーク・アドレスが割り当てられているキュー・マネージャー宛ての INVENTORY.LONDON という CLUSSDR チャンネルを以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

## 5. オプション: 以前にクラスターから除去したキュー・マネージャーを同じクラスターに追加する場合、今回クラスター・メンバーとして表示されることを確認します。表示されない場合は、以下の追加の手順を実行します。

### a) 追加するキュー・マネージャーで **REFRESH CLUSTER** コマンドを発行します。

このステップを行うと、クラスター・チャンネルが停止され、クラスター内の他のキュー・マネージャー内で確実に最新である新しい一連のシーケンス番号が、ローカル・クラスター・キャッシュに設定されます。

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

**注:** 大規模クラスターでは、処理中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、破壊的な影響を及ぼす恐れがあります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。[大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性がある](#)を参照してください。

### b) CLUSSDR チャンネルを再始動します。

(例えば、[START CHANNEL](#) コマンドを使用します)。

### c) CLUSRCVR チャンネルを再始動します。

## タスクの結果

以下の図は、このタスクによってセットアップされるクラスターを示します。

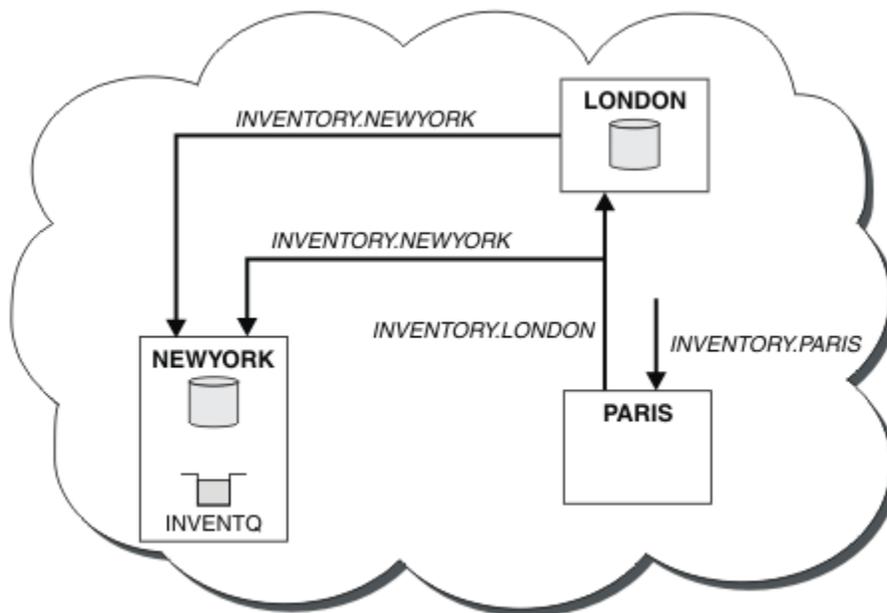


図 40. 3つのキュー・マネージャーで構成される INVENTORY クラスター

CLUSRCVR チャンネルと CLUSSDR チャンネルの 2 つを定義するだけで、キュー・マネージャー PARIS がクラスターに追加されます。

これで、キュー・マネージャー PARIS は、キュー・マネージャー LONDON の完全リポジトリを参照することにより、INVENTQ キューがキュー・マネージャー NEWYORK にホストされていることを認識できるようになります。パリ支店のシステムでホストされているアプリケーションが INVENTQ キューにメッセージを書き込む場合、キュー・マネージャー PARIS はクラスター受信側チャンネル INVENTORY.NEWYORK に接続するクラスター送信側チャンネルを自動的に定義します。このとき、そのアプリケーションのあるキュー・マネージャーの名前を指定して、応答先キューを定義すると、そのアプリケーションはメッセージに対する応答を受信することができます。

### クラスターへのキュー・マネージャーの追加: 別個の伝送キュー

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

#### 始める前に

- キュー・マネージャーはクラスターのメンバーではありません。
- クラスターは存在します。つまり、このキュー・マネージャーが直接接続できる完全リポジトリがあり、そのリポジトリが使用可能になっています。クラスターを作成するステップについては、[248 ページの『新規クラスターのセットアップ』](#)を参照してください。

#### このタスクについて

このタスクは [260 ページの『クラスターにキュー・マネージャーを追加する』](#) (クラスター・メッセージを単一の伝送キューに配置するクラスターにキュー・マネージャーを追加) の代替方法です。

このタスクでは、クライアント送信側チャンネルごとに個別のクライアント伝送キューを自動作成するクラスターにキュー・マネージャーを追加します。

キューの定義数を小さくするために、デフォルトでは単一の伝送キューが使用されます。個々の伝送キューを使用することが有利となるのは、複数の異なるキュー・マネージャーおよび異なるクラスターを宛先とするトラフィックをモニターする必要がある場合です。また、分離またはパフォーマンスの目標を達成するために、さまざまな宛先へのトラフィックを分離しなければならないこともあります。

## 手順

1. デフォルト・クラスター・チャンネル伝送キュー・タイプを変更します。

以下のようにして、キュー・マネージャー PARIS を変更します。

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

キュー・マネージャーは、メッセージをクラスター・マネージャーに送信するためのクラスター送信側チャンネルを作成するたびに、クラスター伝送キューを作成します。これは、このクラスター送信側チャンネル専用の伝送キューです。この伝送キューは、永続動的キューです。これは、モデル・キュー SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE から SYSTEM.CLUSTER.TRANSMIT.ChannelName という名前で作成されます。



**重要:** IBM WebSphere MQ 7.5 より前のバージョンの製品からアップグレードされたキュー・マネージャーで専用の SYSTEM.CLUSTER.TRANSMIT.QUEUES を使用する場合は、SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE の SHARE/NOSHARE オプションが **SHARE** に設定されている必要があります。

2. PARIS が最初に参照する完全リポジトリを決める。

クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリを参照しなければなりません。キュー・マネージャーは、クラスターに関する情報を完全リポジトリから収集し、自分用の部分リポジトリを作成します。いずれかのリポジトリを完全リポジトリとして選択してください。クラスターに新しいキュー・マネージャーを追加すると、そのキュー・マネージャーは他方のリポジトリもすぐに認識できるようになります。キュー・マネージャーに加えられた変更に関する情報は2つのリポジトリに直接送信されます。ここでは、(単に地理的な理由により) キュー・マネージャー PARIS をキュー・マネージャー LONDON にリンクします。

**注:** キュー・マネージャー PARIS の開始後は、残りの手順は任意の順序で実行できます。

3. キュー・マネージャー PARIS に CLUSRCVR チャンネルを定義する。

クラスター内の各キュー・マネージャーにクラスター受信側チャンネルを定義する必要があります。このチャンネルにより、キュー・マネージャーはメッセージを受信することができます。キュー・マネージャー PARIS には、以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

クラスター受信側チャンネルは、このキュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態になったことを通知します。他のキュー・マネージャーでは、クラスター受信側チャンネル INVENTORY.PARIS の送信側用の定義を作成しないでください。他の定義は、必要な場合に自動的に行われます。[クラスター・チャンネル](#)を参照します。

4. キュー・マネージャー PARIS に CLUSSDR チャンネルを定義する。

完全リポジトリではないキュー・マネージャーをクラスターに追加する際は、完全リポジトリとの初期接続を行うためのクラスター送信側チャンネルを1つだけ定義します。[クラスター送信側チャンネル: クラスドル](#)を参照します。

PARIS に、LONDON.CHSTORE.COM というネットワーク・アドレスが割り当てられているキュー・マネージャー宛ての INVENTORY.LONDON という CLUSSDR チャンネルを以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

キュー・マネージャーは、モデル・キュー SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE から自動的に永続動的クラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON を作成します。伝送キューの CLCHNAME 属性は、INVENTORY.LONDON に設定されます。

## タスクの結果

以下の図は、このタスクによってセットアップされるクラスターを示します。

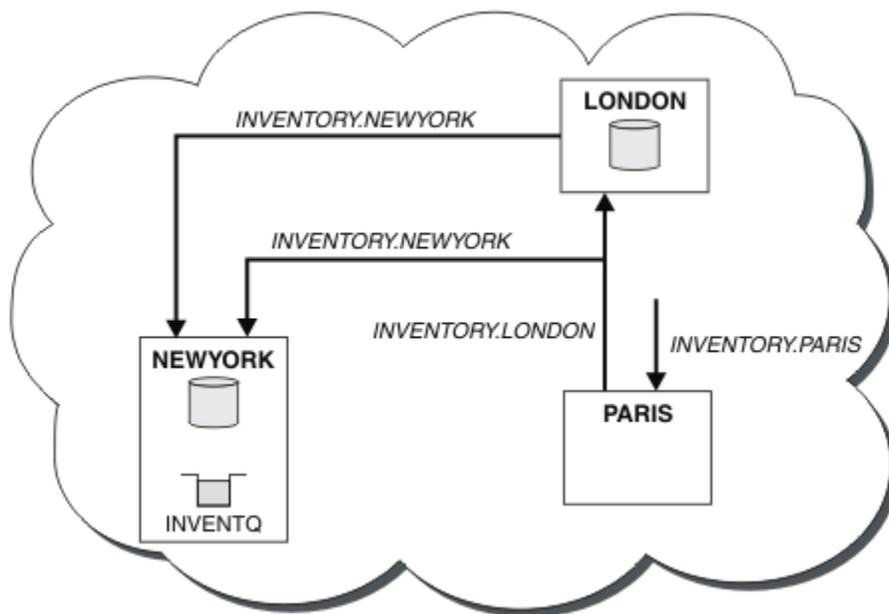


図 41. 3つのキュー・マネージャーで構成される INVENTORY クラスター

CLUSRCVR チャンネルと CLUSSDR チャンネルの2つを定義するだけで、キュー・マネージャー PARIS がクラスターに追加されます。

これで、キュー・マネージャー PARIS は、キュー・マネージャー LONDON の完全リポジトリを参照することにより、INVENTQ キューがキュー・マネージャー NEWYORK にホストされていることを認識できるようになります。パリ支店のシステムでホストされているアプリケーションが INVENTQ キューにメッセージを書き込む場合、キュー・マネージャー PARIS はクラスター受信側チャンネル INVENTORY.NEWYORK に接続するクラスター送信側チャンネルを自動的に定義します。このとき、そのアプリケーションのあるキュー・マネージャーの名前を指定して、応答先キューを定義すると、そのアプリケーションはメッセージに対する応答を受信することができます。

### 関連タスク

#### DHCP を使用したクラスターへのキュー・マネージャーの追加

DHCP を使用して、クラスターにキュー・マネージャーを追加します。この作業では、CLUSRCVR 定義で CONNAME 値を省略することについて説明します。

#### DHCP を使用したクラスターへのキュー・マネージャーの追加

DHCP を使用して、クラスターにキュー・マネージャーを追加します。この作業では、CLUSRCVR 定義で CONNAME 値を省略することについて説明します。

### 始める前に

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも1つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

この作業では、以下の2つの特殊フィーチャーについて説明します。

- CLUSRCVR 定義で CONNAME 値を省略する機能。
- CLUSSDR 定義で +QMNAME+ を使用する機能。

どちらのフィーチャーも、z/OS では提供されません。

シナリオ

- 248 ページの『[新規クラスターのセットアップ](#)』での説明に従って、INVENTORY クラスターを設定しました。このクラスターは、LONDON、NEWYORK という 2 つのキュー・マネージャーで構成されています。そして、両方のキュー・マネージャーに完全リポジトリが格納されています。
- その後、このチェーン店ではパリに新しい支店をオープンすることになったため、このクラスターに PARIS というキュー・マネージャーの追加が必要になりました。
- キュー・マネージャー PARIS は、INVENTQ キューにメッセージを書き込むことにより、ニューヨーク支店のシステムで実行されているアプリケーションに新しい在庫情報を送信するものとします。
- 各支店のシステム間にはネットワークの接続性があるものとします。
- ネットワーク・プロトコルは、TCP です。
- PARIS キュー・マネージャーシステムは DHCP を使用します。したがって、システム再始動時に IP アドレスが変更される可能性があります。
- PARIS および LONDON のシステム間のチャンネルは、定義された命名規則に従って命名されます。この規則には、LONDON にある完全リポジトリ・キュー・マネージャーのキュー・マネージャー名が使用されます。
- PARIS キュー・マネージャーの管理者は、LONDON リポジトリのキュー・マネージャーの名前に関する情報を知りません。LONDON リポジトリのキュー・マネージャーの名前は変わる可能性があります。

## このタスクについて

以下の手順に従って、DHCP を使用して、クラスターにキュー・マネージャーを追加します。

## 手順

1. PARIS が最初に参照する完全リポジトリを決める。

クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリを参照しなければなりません。キュー・マネージャーは、クラスターに関する情報を完全リポジトリから収集し、自分用の部分リポジトリを作成します。いずれかのリポジトリを完全リポジトリとして選択してください。クラスターに新しいキュー・マネージャーを追加すると、そのキュー・マネージャーは他方のリポジトリもすぐに認識できるようになります。キュー・マネージャーに加えられた変更に関する情報は 2 つのリポジトリに直接送信されます。ここでは、(単に地理的な理由により) キュー・マネージャー PARIS をキュー・マネージャー LONDON にリンクすることにします。

**注:** キュー・マネージャー PARIS の開始後は、残りの手順は任意の順序で実行できます。

2. キュー・マネージャー PARIS に CLUSRCVR チャンネルを定義する。

クラスター内の各キュー・マネージャーにクラスター受信側チャンネルを定義する必要があります。このチャンネルにより、キュー・マネージャーはメッセージを受信することができます。キュー・マネージャー PARIS には、以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR)
TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

クラスター受信側チャンネルは、このキュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態になったことを通知します。クラスター受信側チャンネルで CONNAME を指定する必要はありません。IBM MQ に対して、名前を省略するか、または CONNAME を指定することによって、システムからの接続名を検索するように要求することができます。IBM MQ は、システムの現在の IP アドレスを使用して、CONNAME を生成します。[CONNAME](#) を参照してください。このキュー・マネージャーのクラスター受信側チャンネル INVENTORY.PARIS に対応する送信側を、他のキュー・マネージャーに定義する必要はありません。他の定義は、必要な場合に自動的に行われます。

3. キュー・マネージャー PARIS に CLUSSDR チャンネルを定義する。

クラスター内の各キュー・マネージャーにクラスター送信側チャンネルを 1 つ定義する必要があります。このチャンネルにより、キュー・マネージャーは初期完全リポジトリにメッセージを送信することができ

きます。PARIS に、LONDON.CHSTORE.COM というネットワーク・アドレスが割り当てられているキュー・マネージャー宛ての INVENTORY.+QMNAME+ というチャンネルを以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.+QMNAME+) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. オプション: 以前にクラスターから除去したキュー・マネージャーを同じクラスターに追加する場合、今回クラスター・メンバーとして表示されることを確認します。表示されない場合は、以下の追加の手順を実行します。

- a) 追加するキュー・マネージャーで **REFRESH CLUSTER** コマンドを発行します。

このステップを行うと、クラスター・チャンネルが停止され、クラスター内の他のキュー・マネージャー内で確実に最新である新しい一連のシーケンス番号が、ローカル・クラスター・キャッシュに設定されます。

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

**注:** 大規模クラスターでは、処理中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、破壊的な影響を及ぼす恐れがあります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性があるを参照してください。

- b) CLUSSDR チャンネルを再始動します。  
(例えば、**START CHANNEL** コマンドを使用します)。  
c) CLUSRCVR チャンネルを再始動します。

## タスクの結果

この作業によってセットアップされるクラスターは [260 ページの『クラスターにキュー・マネージャーを追加する』](#) の場合と同じです。

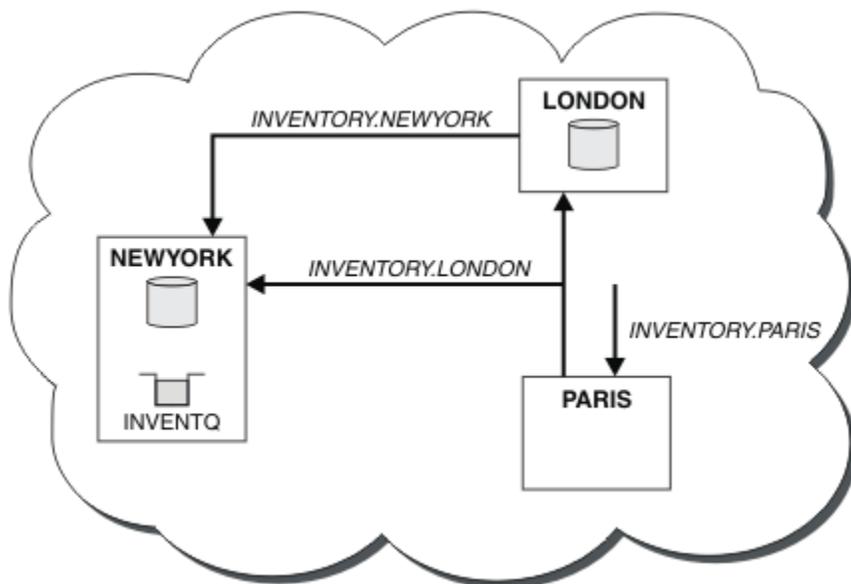


図 42. 3 つのキュー・マネージャーで構成される INVENTORY クラスター

CLUSRCVR チャンネルと CLUSSDR チャンネルの 2 つを定義するだけで、キュー・マネージャー PARIS がクラスターに追加されます。

PARIS キュー・マネージャーで、ストリング+QMNAME+を含む CLUSSDR が開始します。LONDON システム IBM MQ では、+QMNAME+をキュー・マネージャー名(LONDON)に解決します。IBM MQ は、INVENTORY.LONDON というチャンネルの定義を、対応するクラススイーヴル定義に一致させます。

IBM MQ は、解決されたチャンネル名を PARIS キュー・マネージャーに戻します。PARIS では、INVENTORY.+QMNAME+ というチャンネルの CLUSSDR チャンネル定義が、INVENTORY.LONDON に対して内部的に生成された CLUSSDR 定義によって置き換えられます。この定義には解決済みのチャンネル名が含まれていますが、その他はユーザーが作成した+QMNAME+ 定義と同じです。クラスター・リポジトリも、新たに解決されたチャンネル名を伴うチャンネル定義によって、最新の状態にされます。

注:

1. +QMNAME+ 名で作成されたチャンネルは即時に非アクティブになります。それを使用してデータを伝送することは決してありません。
2. チャンネル出口では、ある呼び出しと次の呼び出しの間にチャンネル名が変化した場合、それを確認することができます。

これで、キュー・マネージャー PARIS は、キュー・マネージャー LONDON のリポジトリを参照することにより、INVENTQ キューがキュー・マネージャー NEWYORK にホストされていることを認識できるようになります。システムによって、ホストされているアプリケーションが、メッセージを INVENTQ, PARIS に自動的に設定しようとする、クラスター送信側チャンネルは、クラスター受信側チャンネル INVENTORY.NEWYORK に接続するためのクラスター送信側チャンネルを自動的に定義します。このとき、そのアプリケーションのあるキュー・マネージャーの名前を指定して、応答先キューを定義すると、そのアプリケーションはメッセージに対する応答を受信することができます。

## 関連タスク

[クラスターへのキュー・マネージャーの追加: 別個の伝送キュー](#)

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

## 関連情報

[DEFINE CHANNEL](#)

## キューをホストするキュー・マネージャーの追加

別の INVENTQ キューをホストするために、別のキュー・マネージャーをクラスターに追加します。各キュー・マネージャー上のキューに要求が交互に送信されます。既存の INVENTQ ホストを変更する必要はありません。

## 始める前に

注: クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

シナリオ

- 260 ページの『[クラスターにキュー・マネージャーを追加する](#)』での説明に従って、INVENTORY クラスターを設定しました。このクラスターには、3 つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK の 2 つは完全リポジトリを保有し、PARIS は部分リポジトリを保有しています。在庫管理アプリケーションは、ニューヨークのシステムで動作します。このシステムは、キュー・マネージャー NEWYORK に接続されています。このアプリケーションは、INVENTQ キューのメッセージが到着すると実行されます。
- トロントでは、新しい店舗の準備が進められています。処理能力を増強するため、ニューヨークのシステムと同様、トロントのシステムでも在庫管理アプリケーションを実行する意向があります。
- ネットワークは、4 つのシステム間すべてに接続されています。
- ネットワーク・プロトコルは、TCP です。

注: キュー・マネージャー TORONTO には、部分リポジトリしか入っていません。完全リポジトリ・キュー・マネージャーをクラスターに追加する場合は、[272 ページの『完全リポジトリの別のキュー・マネージャーへの移動』](#)を参照してください。

## このタスクについて

以下の手順に従って、キューをホストするキュー・マネージャーを追加します。

## 手順

1. TORONTO が最初に参照する完全リポジトリを決める。

クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリを参照しなければなりません。キュー・マネージャーは、クラスターに関する情報を完全リポジトリから収集し、自分用の部分リポジトリを作成します。どのリポジトリが選択されるかは、あまり重要ではありません。この例では、NEWYORK を選択します。新しいキュー・マネージャーをクラスターに追加すると、このキュー・マネージャーは両方のリポジトリと通信します。

2. CLUSRCVR チャンネルを定義する。

クラスター内の各キュー・マネージャーにクラスター受信側チャンネルを定義する必要があります。このチャンネルにより、キュー・マネージャーはメッセージを受信することができます。TORONTO では、CLUSRCVR チャンネルを次のように定義します。

```
DEFINE CHANNEL(INVENTORY.TORONTO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(TORONTO.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for TORONTO')
```

TORONTO キュー・マネージャーは、クラスター受信側チャンネルを使用して INVENTORY クラスター内の他のキュー・マネージャーからメッセージを受信できる状態であることを通知します。

3. キュー・マネージャー TORONTO に CLUSSDR チャンネルを定義する。

クラスター内の各キュー・マネージャーは、その最初の完全リポジトリへのメッセージの送信に使用できるクラスター送信側チャンネルを 1 つ定義する必要があります。この例では、NEWYORK を選択します。TORONTO には次の定義が必要です。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from TORONTO to repository at NEWYORK')
```

4. オプション: 以前にクラスターから除去したキュー・マネージャーを同じクラスターに追加する場合、今回クラスター・メンバーとして表示されることを確認します。表示されない場合は、以下の追加の手順を実行します。

- a) 追加するキュー・マネージャーで **REFRESH CLUSTER** コマンドを発行します。

このステップを行うと、クラスター・チャンネルが停止され、クラスター内の他のキュー・マネージャー内で確実に最新である新しい一連のシーケンス番号が、ローカル・クラスター・キャッシュに設定されます。

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

注: 大規模クラスターでは、処理中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、破壊的な影響を及ぼす恐れがあります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性があるを参照してください。

- b) CLUSSDR チャンネルを再始動します。  
(例えば、[START CHANNEL](#) コマンドを使用します)。
- c) CLUSRCVR チャンネルを再始動します。

5. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。

これを調べる前に、在庫管理アプリケーションがメッセージの処理順序に依存しないことを確認して、トロントのシステムにアプリケーションをインストールしてください。

6. クラスター・キュー INVENTQ を定義する。

INVENTQ キューは、キュー・マネージャー NEWYORK に既にホストされていますが、TORONTO によってもホストされます。キュー・マネージャー TORONTO に次のように定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

## タスクの結果

269 ページの図 43 は、このタスクによってセットアップされる INVENTORY クラスターを示しています。

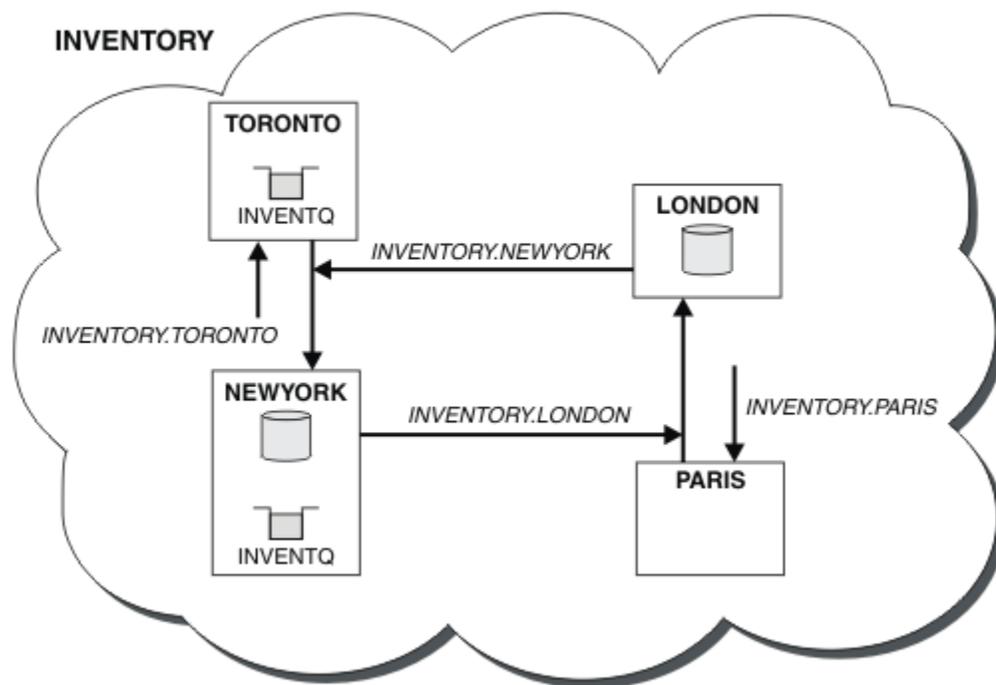


図 43. 4 つのキュー・マネージャーが格納されている INVENTORY クラスター

現在、INVENTQ キューと在庫管理アプリケーションは、このクラスター内の 2 つのキュー・マネージャーによってホストされています。これにより、これらの可用性が増大してメッセージのスループットが向上し、ワークロードが 2 つのキュー・マネージャーに分散されるようになります。TORONTO または NEWYORK によって INVENTQ に書き込まれたメッセージは、可能なときは必ずローカル・キュー・マネージャーのインスタンスによって処理されます。LONDON または PARIS によって書き込まれたメッセージは、TORONTO または NEWYORK に交互に経路指定されるため、ワークロードのバランスを取ることができます。

クラスターをこのように変更するとき、NEWYORK、LONDON、および PARIS の各キュー・マネージャーの定義を変更する必要はありませんでした。これらのキュー・マネージャーの完全リポジトリは、TORONTO の INVENTQ にメッセージを送信できるようにするために必要な情報によって、自動的に更新されます。在庫管理アプリケーションは、NEWYORK または TORONTO のうちの 1 つのキュー・マネージャーが使用不可の状態になり、それ自身に十分な処理能力がある場合、動作を継続できます。在庫管理アプリケーションが両方の場所でホストされている場合、正しく動作するはずですが。

この作業結果から分かるように、複数のキュー・マネージャーで同じアプリケーションを実行することが可能です。ワークロードを均等に分散するためにクラスタリングが可能です。

アプリケーションは両方の場所でレコードを処理できない可能性があります。例えば、カスタマー・アカウント照会を追加し、LONDON および NEWYORK で実行されているアプリケーションを更新することを決めたと仮定します。アカウント・レコードは 1 カ所でのみ保有できます。データ区分化の手法を用いて、要求の分散方法を制御することを選択できます。レコードの配布を分割できます。レコードの半分を整理して、例えばアカウント番号 00000 から 49999 までを LONDON で保有するようにすることができます。残りの半分である 50000 から 99999 までは、NEWYORK で保有します。その後、すべてのメッセージのアカウント・フィールドを調べ、該当するキュー・マネージャーにメッセージを送信するクラスター・ワークロード出口プログラムを作成することができます。

## 次のタスク

これですべての定義が完了しました。IBM MQ for z/OS でまだチャンネル・イニシエーターを開始していない場合は開始します。すべてのプラットフォームで、キュー・マネージャー TORONTO でリスナー・プログラムを始動します。リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合にクラスター受信側チャンネルを始動します。

### **z/OS** 既存のクラスターへのキュー共有グループの追加

z/OS のキュー共有グループを既存のクラスターに追加します。

## 始める前に

注:

1. クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。
2. キュー共有グループは、IBM MQ for z/OS でのみサポートされています。この作業は、他のプラットフォームには該当しません。

シナリオ

- 248 ページの『[新規クラスターのセットアップ](#)』での説明に従って、INVENTORY クラスターを設定しました。これは、LONDON および NEWYORK という 2 つのキュー・マネージャーで構成されています。
- このクラスターにキュー共有グループを追加することにします。グループ QSGP は、3 つのキュー・マネージャー P1、P2、および P3 で構成されます。これらは、P1 によって定義される INVENTQ キューのインスタンスを共有します。

## このタスクについて

以下の手順に従って、共有キューをホストする新しいキュー・マネージャーを追加します。

## 手順

1. キュー・マネージャーが最初に参照する完全リポジトリを決める

クラスター内の各キュー・マネージャーは、いずれかの完全リポジトリを参照しなければなりません。キュー・マネージャーは、クラスターに関する情報を完全リポジトリから収集し、自分用の部分リポジトリを作成します。どの完全リポジトリが選択されるかは、あまり重要ではありません。この例では、NEWYORK を選択します。キュー共有グループをクラスターに追加すると、それは両方の完全リポジトリと通信します。

2. CLUSRCVR チャンネルを定義する。

クラスター内の各キュー・マネージャーにクラスター受信側チャンネルを定義する必要があります。このチャンネルにより、キュー・マネージャーはメッセージを受信することができます。P1、P2、および P3 で、以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.Pn) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
```

```
CONNNAME(Pn.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for sharing queue manager')
```

クラスター受信側チャンネルは、各キュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態であることを通知します。

### 3. キュー共有グループの CLUSSDR チャンネルを定義する

クラスター内の各メンバーは、そのプログラムの最初の完全リポジトリにメッセージを送信できるクラスター送信側チャンネルを1つ定義する必要があります。この例では、NEWYORK を選択しました。キュー共有グループ内のキュー・マネージャーの1つには、以下のようなグループ定義が必要です。この定義により、すべてのキュー・マネージャーが必ずクラスター送信側チャンネル定義を持つようになります。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) QSGDISP(GROUP)
DESCR('Cluster-sender channel to repository at NEWYORK')
```

### 4. 共有キューを定義する

次のように P1 にキュー INVENTQ を定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

新規キュー・マネージャーで、チャンネル・イニシエーターおよびリスナー・プログラムを始動します。リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャンネルを起動します。

## タスクの結果

271 ページの図 44 は、このタスクによってセットアップされるクラスターを示します。

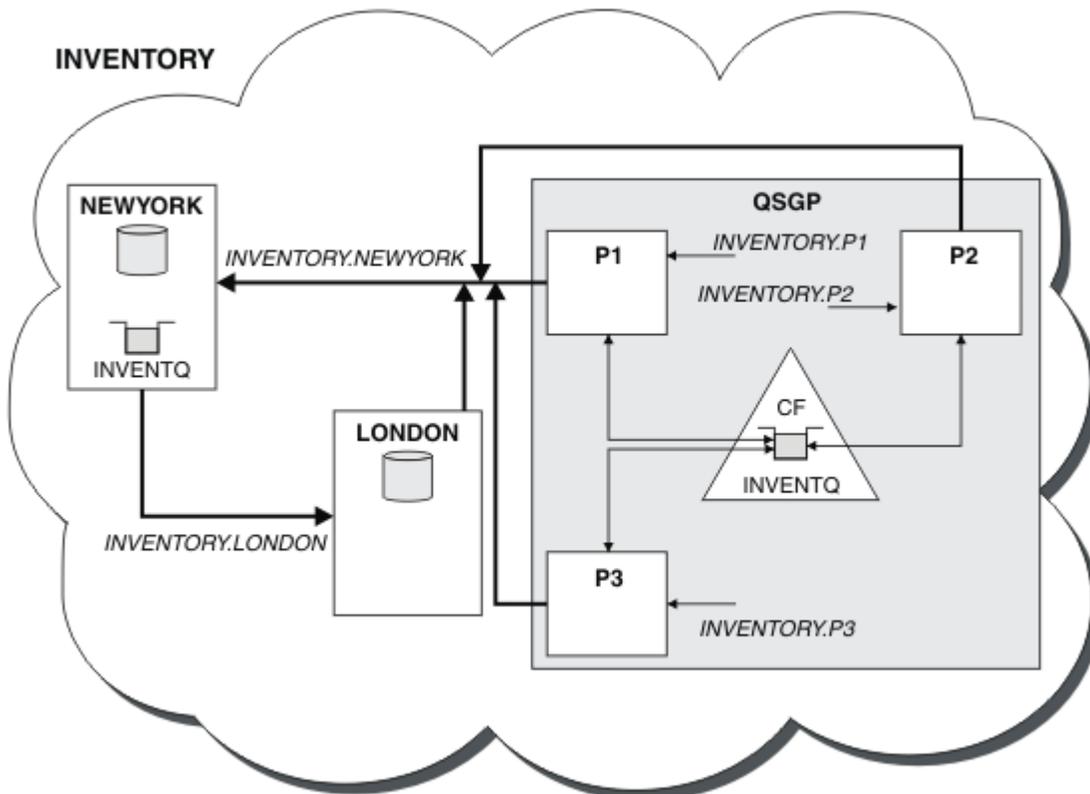


図 44. クラスターとキュー共有グループ

これで、LONDON によって INVENTQ キューに入れられるメッセージは、キューをホストすると公示されている 4 つのキュー・マネージャーの間で経路指定されます。

## 次のタスク

キュー共有グループのメンバーがクラスター・キューをホストする利点は、グループの他のメンバーが要求に対して応答できるということです。この例では、P1 が、共有キューでメッセージを受信した後で利用不可になったとします。キュー共有グループの他のメンバーが代わりに応答できます。

## 完全リポジトリの別のキュー・マネージャーへの移動

完全リポジトリを 1 つのキュー・マネージャーから別のキュー・マネージャーへ移動し、2 番目のリポジトリで保有された情報から新しいリポジトリを作成します。

## 始める前に

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

シナリオ

- 260 ページの『[クラスターにキュー・マネージャーを追加する](#)』での説明に従って、INVENTORY クラスターを設定しました。
- 業務上の理由により、キュー・マネージャー LONDON から完全リポジトリを削除して、これをキュー・マネージャー PARIS の完全リポジトリで置きかえる必要が生じました。キュー・マネージャー NEWYORK は、引き続き完全リポジトリを保有します。

## このタスクについて

以下の手順に従って、完全リポジトリを別のキュー・マネージャーへ移動します。

## 手順

1. PARIS を完全リポジトリ・キュー・マネージャーに変更する

PARIS で、以下のコマンドを発行します。

```
ALTER QMGR REPOS(INVENTORY)
```

2. PARIS で CLUSSDR チャンネルを追加する

PARIS には、現在宛先が LONDON になっているクラスター送信側チャンネルがあります。LONDON はクラスターの完全リポジトリを保有しなくなりました。PARIS には、NEWYORK を指す新しいクラスター送信側チャンネルが必要です。このチャンネルでは、もう一方のフル・リポジトリが保持されています。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-sender channel from PARIS to repository at NEWYORK')
```

3. 宛先が PARIS の CLUSSDR チャンネルを NEWYORK に定義する

現在、NEWYORK には、宛先が LONDON になっているクラスター送信側チャンネルがあります。他の完全リポジトリが PARIS に移動したので、宛先が PARIS の新しいクラスター送信側チャンネルを NEWYORK に追加する必要があります。

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-sender channel from NEWYORK to repository at PARIS')
```

PARIS にクラスター送信側チャンネルを追加すると、PARIS は NEWYORK からクラスターに関する情報を取得します。そして、NEWYORK からの情報を利用して、固有の完全リポジトリを作成します。

#### 4. キュー・マネージャー PARIS に現在完全リポジトリが存在することを確認する

キュー・マネージャー PARIS が、キュー・マネージャー NEWYORK の完全リポジトリから、固有の完全リポジトリを作成したことを確認します。次のコマンドを発行します。

```
DIS QCLUSTER(*) CLUSTER (INVENTORY)
DIS CLUSQMR(*) CLUSTER (INVENTORY)
```

これらのコマンドによって、このクラスター内の NEWYORK にあるものと同じリソースの詳細が表示されることを確認します。

**注:** キュー・マネージャー NEWYORK が使用できない場合、この情報の作成を完了することはできません。この作業が完了するまで、次の手順に進まないでください。

#### 5. LONDON のキュー・マネージャー定義を変更する

最後に LONDON のキュー・マネージャーを変更して、このキュー・マネージャーがクラスターの完全リポジトリを保有しないようにします。LONDON で、次のコマンドを発行します。

```
ALTER QMGR REPOS(' ')
```

キュー・マネージャーは、クラスター情報を受け取ることがなくなりました。30 日後、その完全リポジトリに格納されている情報は、期限が切れます。このため、キュー・マネージャー LONDON は、固有の部分リポジトリを作成します。

#### 6. 未解決の定義を削除または変更する

クラスターの新しい配置が予期されるとおりに機能していることを確認したら、正確ではなくなった、手動で定義された CLUSSDR 定義を削除または変更します。

- キュー・マネージャー PARIS で LONDON へのクラスター送信側チャンネルを停止して削除する必要があります。次に、開始チャンネル・コマンドを発行して、クラスターが自動チャンネルを再度使用することができるようにします。

```
STOP CHANNEL(INVENTORY.LONDON)
DELETE CHANNEL(INVENTORY.LONDON)
START CHANNEL(INVENTORY.LONDON)
```

- キュー・マネージャー NEWYORK で LONDON へのクラスター送信側チャンネルを停止して削除してください。その後、開始チャンネル・コマンドを発行するとクラスターは自動チャンネルを再度使用することができます。

```
STOP CHANNEL(INVENTORY.LONDON)
DELETE CHANNEL(INVENTORY.LONDON)
START CHANNEL(INVENTORY.LONDON)
```

- クラスター内のすべてのキュー・マネージャーにおいて、宛先が LONDON の、手動で定義された他のすべてのクラスター送信側チャンネルを、宛先が NEWYORK または PARIS のチャンネルに置き換えます。チャンネルを削除した後は、必ず **start channel** コマンドを発行して、クラスターで自動チャンネルを再び使用できるようにしてください。この小規模な例では、他のチャンネルはありません。他に忘れていたチャンネルがないか確認するには、各キュー・マネージャーから、TYPE(CLUSSDR) を指定して DISPLAY CHANNEL コマンドを発行します。以下に例を示します。

```
DISPLAY CHANNEL(*) TYPE(CLUSSDR)
```

完全リポジトリを LONDON から PARIS に移動した後、できるだけ早くこの作業を実行することが重要です。この作業を実行するまでの間に、INVENTORY.LONDON という CLUSSDR チャンネルを手動で定義したキュー・マネージャーが、このチャンネルを使用する情報の要求を送信する可能性があります。

LONDONでは、完全リポジトリではなくなった後にそのような要求を受け取った場合、キュー・マネージャーのエラー・ログにエラー・メッセージを書き込みます。以下の例は、LONDONで表示される可能性のあるエラー・メッセージを示します。

- AMQ9428: 予期しないクラスター・キュー・オブジェクトのパブリッシュを受け取りました
- AMQ9432: 照会が非リポジトリ・キュー・マネージャーで受信されました

キュー・マネージャー LONDON は完全リポジトリではないので、情報の要求に回答しません。LONDONから情報を要求しているキュー・マネージャーは、手動で定義された CLUSSDR 定義が PARIS を宛先とするよう修正されるまで NEWYORK を使用してクラスター情報を得る必要があります。この状態を、有効な構成として長い期間容認するべきではありません。

## タスクの結果

274 ページの図 45 は、このタスクによってセットアップされるクラスターを示します。

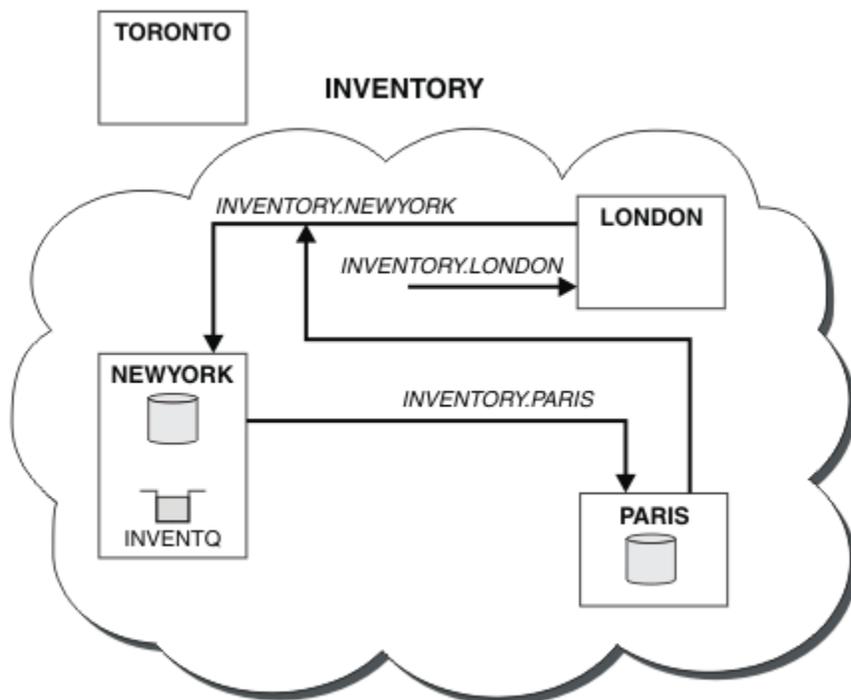


図 45. 完全リポジトリが PARIS に移動した INVENTORY クラスター

## クラスター内での通信の確立

送達するメッセージがある場合に、通信チャンネルを開始するには、チャンネル開始プログラムが必要です。チャンネル・リスナーは、チャンネルの相手側が開始されてメッセージが受信されるのを待ちます。

### 始める前に

クラスター内にあるキュー・マネージャー間の通信を確立するには、サポートされている通信プロトコルのいずれかを使用してリンクを構成します。すべてのプラットフォームでサポートされているプロトコルとしては、TCP と LU 6.2 があります。また、Windows システムでは NetBIOS または SPX がサポートされています。分散キューイングの場合と同様に、この設定処理でもチャンネル開始プログラムおよびチャンネル・リスナーを設定する必要があります。

### このタスクについて

チャンネル開始プログラムは、すべてのクラスター・キュー・マネージャーに設定する必要があります。チャンネル開始プログラムは、システムによって定義された SYSTEM.CHANNEL.INITQ という開始キューをモ

ニターします。SYSTEM.CHANNEL.INITQ は、クラスター伝送キューを含むすべての伝送キューの開始キューとなります。

各キュー・マネージャーには、チャンネル・リスナーが必要です。チャンネル・リスナー・プログラムは、着信ネットワーク要求を待機し、必要な場合は、該当する受信側チャンネルを起動します。チャンネル・リスナーの実装方法は、プラットフォームによって異なりますが、一部の機能は共通しています。すべての IBM MQ プラットフォームで、リスナーは START LISTENER コマンドを使用して起動可能です。IBM MQ for IBM i、Windows、および UNIX and Linux システムでは、キュー・マネージャーと同時にリスナーを自動的に開始できます。リスナーを自動的に起動するには、LISTENER オブジェクトの CONTROL 属性を QMGR または STARTONLY に設定してください。

**z/OS** 非共有リスナー・ポート (INDISP(QMGR)) は、z/OS の CLUSRCVR チャンネルおよび CLUSSDR チャンネルの場合は z/OS に使用する必要があります。

## 手順

1. チャンネル・イニシエーターを開始します。

• **z/OS**

### IBM MQ for z/OS

各キュー・マネージャーに 1 つのチャンネル開始プログラムが設定されています。これらのチャンネル開始プログラムはそれぞれ別個のアドレス・スペースとして動作します。MQSC START CHINIT コマンドを使用してチャンネル開始プログラムを開始しますが、このコマンドは、キュー・マネージャーの始動時に発行します。

• **ULW**

### IBM MQ for UNIX, Linux, and Windows

キュー・マネージャー属性 SCHINIT が QMGR に設定されている場合は、キュー・マネージャーを始動すると、チャンネル開始プログラムが自動的に開始されます。それ以外の場合は、runmqsc START CHINIT コマンドまたは runmqchi 制御コマンドを使用することで開始できます。

• **IBM i**

### IBM MQ for IBM i

キュー・マネージャー属性 SCHINIT が QMGR に設定されている場合は、キュー・マネージャーを始動すると、チャンネル開始プログラムが自動的に開始されます。それ以外の場合は、runmqsc START CHINIT コマンドまたは runmqchi 制御コマンドを使用することで開始できます。

2. チャンネル・リスナーを開始します。

• **z/OS**

### IBM MQ for z/OS

IBM MQ 提供のチャンネル・リスナー・プログラムを使用します。IBM MQ チャンネル・リスナーを起動するには、MQSC コマンドの START LISTENER を使用しますが、このコマンドは、チャンネル開始プログラムの始動時に発行します。以下に例を示します。

```
START LISTENER PORT(1414) TRPTYPE(TCP)
```

または:

```
START LISTENER LUNAME(LONDON.LUNAME) TRPTYPE(LU62)
```

キュー共有グループのメンバーは、キュー・マネージャーごとにリスナーを使用する代わりに、共有リスナーを使用できます。共有リスナーをクラスターと併用しないでください。具体的に言うと、CLUSRCVR チャンネルの CONNAME をキュー共有グループの共有リスナーのアドレスに使用しないでください。使用すると、キュー・マネージャーは、定義していないキューに関するメッセージを受信する可能性があります。

## IBM i

### IBM MQ for IBM i

IBM MQ 提供のチャンネル・リスナー・プログラムを使用します。IBM MQ チャンネル・リスナーを起動するには、**CL** コマンドの **STRMQLSR** を使用します。以下に例を示します。

```
STRMQLSR MQMNAME(QM1) PORT(1414)
```

## Windows

### IBM MQ for Windows

IBM MQ 提供のチャンネル・リスナー・プログラムまたはオペレーティング・システム提供の機能を使用します。

IBM MQ 提供のチャンネル・リスナーを起動するときには **RUNMQLSR** コマンドを使用します。以下に例を示します。

```
RUNMQLSR -t tcp -p 1414 -m QM1
```

## Linux

## UNIX

### IBM MQ on UNIX and Linux

IBM MQ 提供のチャンネル・リスナー・プログラムか、オペレーティング・システム提供の機能 (例えば、TCP 通信の場合には **inetd**) のどちらかを使用します。

IBM MQ 提供のチャンネル・リスナーを起動するときには **runmqlsr** コマンドを使用します。以下に例を示します。

```
runmqlsr -t tcp -p 1414 -m QM1
```

**inetd** によってチャンネルを起動する場合には、次の 2 つのファイルを構成してください。

- /etc/services** というファイルを編集する。スーパーユーザーまたはルート・ユーザーとしてログインする必要があります。このファイルに次の行がない場合は、それを以下のように追加します。

```
MQSeries      1414/tcp      # WebSphere MQ channel listener
```

1414 は、IBM MQ に必要なポート番号を示します。このポート番号は変更できますが、送信側で指定したポート番号と一致しなければなりません。

- /etc/inetd.conf** というファイルを編集する。以下に示す行がこのファイルにない場合には、次のとおりに入力してファイルに追加してください。

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
-m queue.manager.name
```

**MQ\_INSTALLATION\_PATH** は、IBM MQ のインストール先の上位ディレクトリーに置き換えられます。

**inetd** により構成ファイルが再度読み取られると、新しい設定が有効になります。次に、ルート・ユーザー ID により、以下に示すコマンドを実行します。

AIX の場合:

```
refresh -s inetd
```

HP-UX の場合:

```
inetd -c
```

Solaris または Linux の場合:

- a. 以下のコマンドにより、**inetd** のプロセス ID を調べる

```
ps -ef | grep inetd
```

- b. 以下の該当するコマンドを実行する:

- Solaris 9 と Linux の場合:

```
kill -1 inetd processid
```

- Solaris 10 以降のバージョンの場合:

```
inetconv
```

## 既存のネットワークのクラスターへの変換

既存の分散キューイング・ネットワークをクラスターに変換し、キュー・マネージャーを追加して容量を増やします。

### 始める前に

272 ページの『[完全リポジトリの別のキュー・マネージャーへの移動](#)』から 248 ページの『[新規クラスターのセットアップ](#)』では、新規クラスターを作成して拡張しました。次の 2 つの作業では、キュー・マネージャーの既存のネットワークをクラスターに変換するためのさまざまな手法について調べます。

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

シナリオ

- IBM MQ ネットワークは既に構築されており、これによりチェーン店の支店が全国規模で接続されています。ネットワークの構造はハブ・スポーク構造であり、すべてのキュー・マネージャーは中心にある 1 つのキュー・マネージャーに接続されています。中心にあるキュー・マネージャーは、在庫管理アプリケーションが動作するシステム上にあります。アプリケーションは、INVENTQ キューにメッセージが到着すると実行されます。このキューの各キュー・マネージャーには、リモート・キュー定義があります。

このネットワークを [278 ページの図 46](#) に示します。

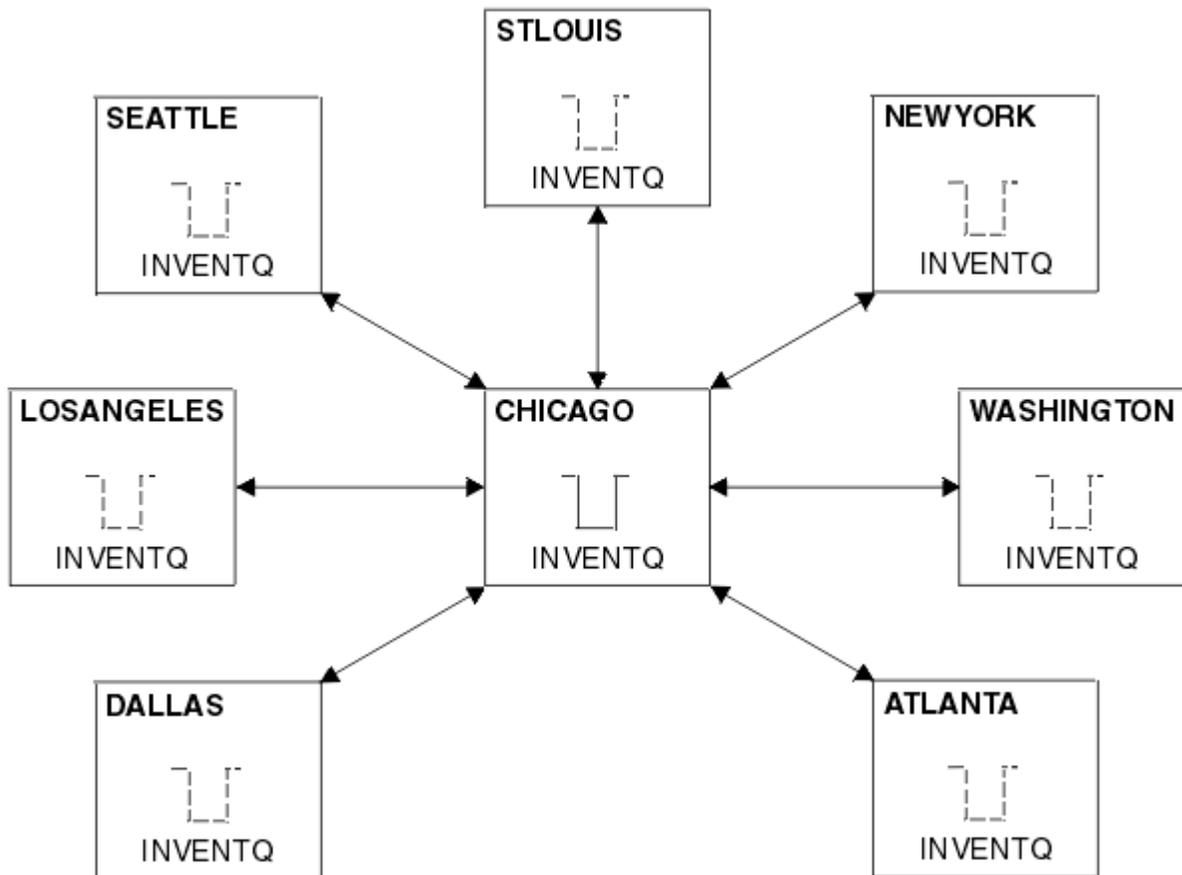


図 46. ハブ・スポーク・ネットワーク

- 管理を容易にするため、このネットワークをクラスターに変換して、中心の場所に別のキュー・マネージャーを作成して、ワークロードを共有する予定です。

クラスター名は CHNSTORE とします。

注：20 文字の最大長を超えない、形式が `cluster_name.queue_manager_name` の名前 (例えば、CHNSTORE.WASHINGTON) を使用してクラスター受信側チャンネル名を作成できるように、クラスター名 CHNSTORE が選択されています。

- 中心にある 2 つのキュー・マネージャーは、完全リポジトリをホストし、在庫管理アプリケーションにアクセス可能です。
- 在庫管理アプリケーションは、中心にある 2 つのキュー・マネージャーのいずれかによってホストされた INVENTQ キューのメッセージが到着すると実行されます。
- 在庫管理アプリケーションは、同時に実行される唯一のアプリケーションとなり、複数のキュー・マネージャーによってアクセスが可能です。他のすべてのアプリケーションは、以前と同様の動作を継続します。
- すべての支店は、中心にある 2 つのキュー・マネージャーにネットワークで接続されます。
- ネットワーク・プロトコルは、TCP です。

## このタスクについて

以下の手順に従って、既存のネットワークをクラスターに変換します。

### 手順

1. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。

先に進む前に、アプリケーションがメッセージの類縁性を処理できることを確認してください。メッセージの類縁性とは、2 つのアプリケーション間で交換される会話型メッセージ間の関係で、それらのメ

メッセージは特定のキュー・マネージャーによって、または特定の順序で処理されなければなりません。メッセージの類縁性について詳しくは、[355 ページの『メッセージの類縁性の処理』](#)を参照してください。

2. 中心にある 2 つのキュー・マネージャーを完全リポジトリ・キュー・マネージャーに変更する。

2 つのキュー・マネージャー CHICAGO および CHICAGO2 が、このネットワークのハブにあります。CHAINSTORE クラスターに対応する活動をこれら 2 つのキュー・マネージャーに集約することに決めました。在庫管理アプリケーションおよび INVENTQ キューの定義以外に、クラスターの 2 つの完全リポジトリをこれらのキュー・マネージャーにホストさせることが必要です。2 つのキュー・マネージャーで、それぞれ次のコマンドを出します。

```
ALTER QMGR REPOS(CHNSTORE)
```

3. 各キュー・マネージャーで CLUSRCVR チャンネルを定義する。

クラスター内の各キュー・マネージャーで、クラスター受信側チャンネルとクラスター送信側チャンネルを定義してください。どちらのチャンネルを最初に定義しても構いません。

CLUSRCVR 定義を作成して、各キュー・マネージャー、そのネットワーク・アドレス、および他の情報をクラスターに通知します。例えば、キュー・マネージャー ATLANTA では、次のように定義します。

```
DEFINE CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNAME(ATLANTA.CHSTORE.COM) CLUSTER(CHNSTORE)  
DESCR('Cluster-receiver channel')
```

4. 各キュー・マネージャーで CLUSSDR チャンネルを定義する。

各キュー・マネージャーで CLUSSDR を定義して、そのキュー・マネージャーをいずれかの完全リポジトリ・キュー・マネージャーにリンクします。例えば、ATLANTA を CHICAGO2 にリンクします。

```
DEFINE CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNAME(CHICAGO2.CHSTORE.COM) CLUSTER(CHNSTORE)  
DESCR('Cluster-sender channel to repository queue manager')
```

5. 在庫管理アプリケーションを CHICAGO2 にインストールする。

キュー・マネージャー CHICAGO には、既に在庫管理アプリケーションがあります。したがって、キュー・マネージャー CHICAGO2 にこのアプリケーションをコピーする必要があります。

6. 中心にあるキュー・マネージャーに INVENTQ キューを定義する。

CHICAGO で、INVENTQ キューのローカル・キュー定義を変更して、そのキューをそのクラスターで使用できるようにします。次のコマンドを実行します。

```
ALTER QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

CHICAGO2 で、同じキューの定義を作成します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

z/OS では、**CSQUTIL** の COMMAND 関数の MAKEDEF オプションを使用して、CHICAGO の INVENTQ を CHICAGO2 に正確にコピーすることができます。

これらを定義すると、CHICAGO と CHICAGO2 の完全リポジトリにメッセージが送信され、それらの中の情報が更新されます。キュー・マネージャーは、メッセージを INVENTQ に書き込む際に、メッセージの宛先を選択できることを完全リポジトリから判断します。

7. クラスターの変更が伝搬されたことを確認する。

前の手順で作成した定義がクラスター全体に伝搬されたことを確認します。完全リポジトリ・キュー・マネージャーで以下のコマンドを発行します。

## 相互接続された新しいクラスターの追加

既存のクラスターと一部のキュー・マネージャーを共有する新しいクラスターを追加します。

### 始める前に

注:

1. クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。
2. この作業を開始する前に、キュー名が衝突していないかどうかを確認し、衝突による影響を把握します。作業を進める前に、キューの名前を変更するか、キューの別名を設定する必要があることがあります。

シナリオ

- 277 ページの『既存のネットワークのクラスターへの変換』での説明に従って、IBM MQ クラスターを設定しました。
- MAILORDER と呼ばれる新しいクラスターが実行される予定です。このクラスターには、CHNSTORE クラスターにあるキュー・マネージャーのうちの 4 つ (CHICAGO、CHICAGO2、SEATTLE、ATLANTA) が格納され、さらに 2 つのキュー・マネージャー (HARTFORD および OMAHA) が追加されます。MAILORDER アプリケーションは、オマハにあるシステムで実行され、キュー・マネージャー OMAHA に接続します。このアプリケーションは、クラスター内にある他のキュー・マネージャーのうち、MORDERQ キューにメッセージを書き込むキュー・マネージャーによって実行されます。
- MAILORDER クラスターの完全リポジトリは、2 つのキュー・マネージャー CHICAGO および CHICAGO2 によって保守されます。
- ネットワーク・プロトコルは、TCP です。

### このタスクについて

以下の手順に従って、新規の内部接続されたクラスターを追加します。

### 手順

1. クラスター名の名前リストを作成する

CHICAGO と CHICAGO2 の完全リポジトリ・キュー・マネージャーは、CHNSTORE と MAILORDER の両方のクラスターの完全リポジトリを保有する予定です。まず、クラスターの名前を含む名前リストを作成します。CHICAGO と CHICAGO2 の名前リストを次のように定義します。

```
DEFINE NAMELIST(CHAINMAIL)
DESCR('List of cluster names')
NAMES(CHNSTORE, MAILORDER)
```

2. 2 つのキュー・マネージャー定義を変更する

この時点で、CHICAGO および CHICAGO2 の 2 つのキュー・マネージャー定義を変更します。現在、これらの定義では、これらのキュー・マネージャーが CHNSTORE クラスターの完全リポジトリを保有することになっています。これらのキュー・マネージャーが、CHAINMAIL 名前リストに登録されているすべてのクラスターの完全リポジトリを保有するように、この定義を変更してください。次のように、CHICAGO および CHICAGO2 キュー・マネージャー定義を変更します。

```
ALTER QMGR REPOS(' ') REPOSNL(CHAINMAIL)
```

3. CHICAGO および CHICAGO2 で CLUSRCVR チャンネルを変更する

CHICAGO および CHICAGO2 の CLUSRCVR チャンネル定義では、チャンネルは、CHNSTORE クラスターで使用可能であることになっています。クラスター受信側定義を変更して、CHAINMAIL 名前リストに含まれるすべてのクラスターに対してチャンネルが使用可能であることを示す必要があります。次のようにして、CHICAGO でクラスター受信側定義を変更します。

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

CHICAGO2 で、次のコマンドを入力します。

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

#### 4. CHICAGO および CHICAGO2 で CLUSSDR チャンネルを変更する。

2 つの CLUSSDR チャンネル定義を変更して、名前リストを追加します。CHICAGO で、次のコマンドを入力します。

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

CHICAGO2 で、次のコマンドを入力します。

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

#### 5. SEATTLE および ATLANTA で名前リストを作成する。

SEATTLE および ATLANTA は複数のクラスターのメンバーになるので、複数のクラスターの名前が入っている名前リストを作成する必要があります。SEATTLE および ATLANTA で、次のように名前リストを定義します。

```
DEFINE NAMELIST(CHAINMAIL)
DESCR('List of cluster names')
NAMES(CHNSTORE, MAILORDER)
```

#### 6. SEATTLE および ATLANTA で CLUSRCVR チャンネルを変更する。

SEATTLE および ATLANTA の CLUSRCVR チャンネル定義では、チャンネルは、CHNSTORE クラスターで使用可能であることになっています。クラスター受信側チャンネル定義を変更して CHAINMAIL 名前リストに含まれるすべてのクラスターに対してチャンネルが使用可能であることを示す必要があります。SEATTLE で、次のコマンドを入力します。

```
ALTER CHANNEL(CHNSTORE.SEATTLE) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

ATLANTA で、次のコマンドを入力します。

```
ALTER CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

#### 7. SEATTLE および ATLANTA で CLUSSDR チャンネルを変更する。

2 つの CLUSSDR チャンネル定義を変更して、名前リストを追加します。SEATTLE で、次のコマンドを入力します。

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

ATLANTA で、次のコマンドを入力します。

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

#### 8. CLUSRCVR および CLUSSDR チャンネルを HARTFORD および OMAHA で定義する。

2つの新しいキュー・マネージャー HARTFORD および OMAHA で、クラスター受信側チャンネルとクラスター送信側チャンネルを定義してください。どちらを先に定義しても構いません。HARTFORD で、次のコマンドを入力します。

```
DEFINE CHANNEL(MAILORDER.HARTFORD) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(HARTFORD.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for HARTFORD')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from HARTFORD to repository at CHICAGO')
```

OMAHA で、次のコマンドを入力します。

```
DEFINE CHANNEL(MAILORDER.OMAHA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(OMAHA.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for OMAHA')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from OMAHA to repository at CHICAGO')
```

#### 9. OMAHA で MORDERQ キューを定義する。

この作業の最終段階では、MORDERQ キューをキュー・マネージャー OMAHA で定義します。OMAHA で、次のコマンドを入力します。

```
DEFINE QLOCAL(MORDERQ) CLUSTER(MAILORDER)
```

#### 10. クラスターの変更が伝搬されたことを確認する。

前の手順で作成した定義がクラスター全体に伝搬されたことを確認します。完全リポジトリ・キュー・マネージャーで、次のコマンドを発行します。

```
DIS QCLUSTER (MORDERQ)
DIS CLUSQMGR
```

#### 11.

## タスクの結果

上述の作業によって設定されるクラスターは、[283 ページの図 47](#) で示されています。

現在の設定では、クラスターが2つ重なり合っています。2つのクラスターの完全リポジトリは、CHICAGO および CHICAGO2 に保有されています。OMAHA で実行される MAILORDER アプリケーションは、CHICAGO で実行される在庫管理アプリケーションとは独立しています。しかし、一部のキュー・マネージャーは、CHNSTORE クラスターと MAILORDER クラスターの両方に属しているため、それらのキュー・マネージャーは、どちらのアプリケーションにもメッセージを送信できます。この作業を実行して2つのクラスターを重ね合わせるときは、キュー名が衝突しないよう注意してください。

CHNSTORE クラスターの NEWYORK と MAILORDER クラスターの OMAHA に、ACCOUNTQ と呼ばれるキューがあると仮定します。クラスターをオーバーラップさせた後で、SEATTLE にあるアプリケーションが ACCOUNTQ キューにメッセージを書き込んだ場合、メッセージは ACCOUNTQ のどちらのインスタンスにも送信される可能性があります。

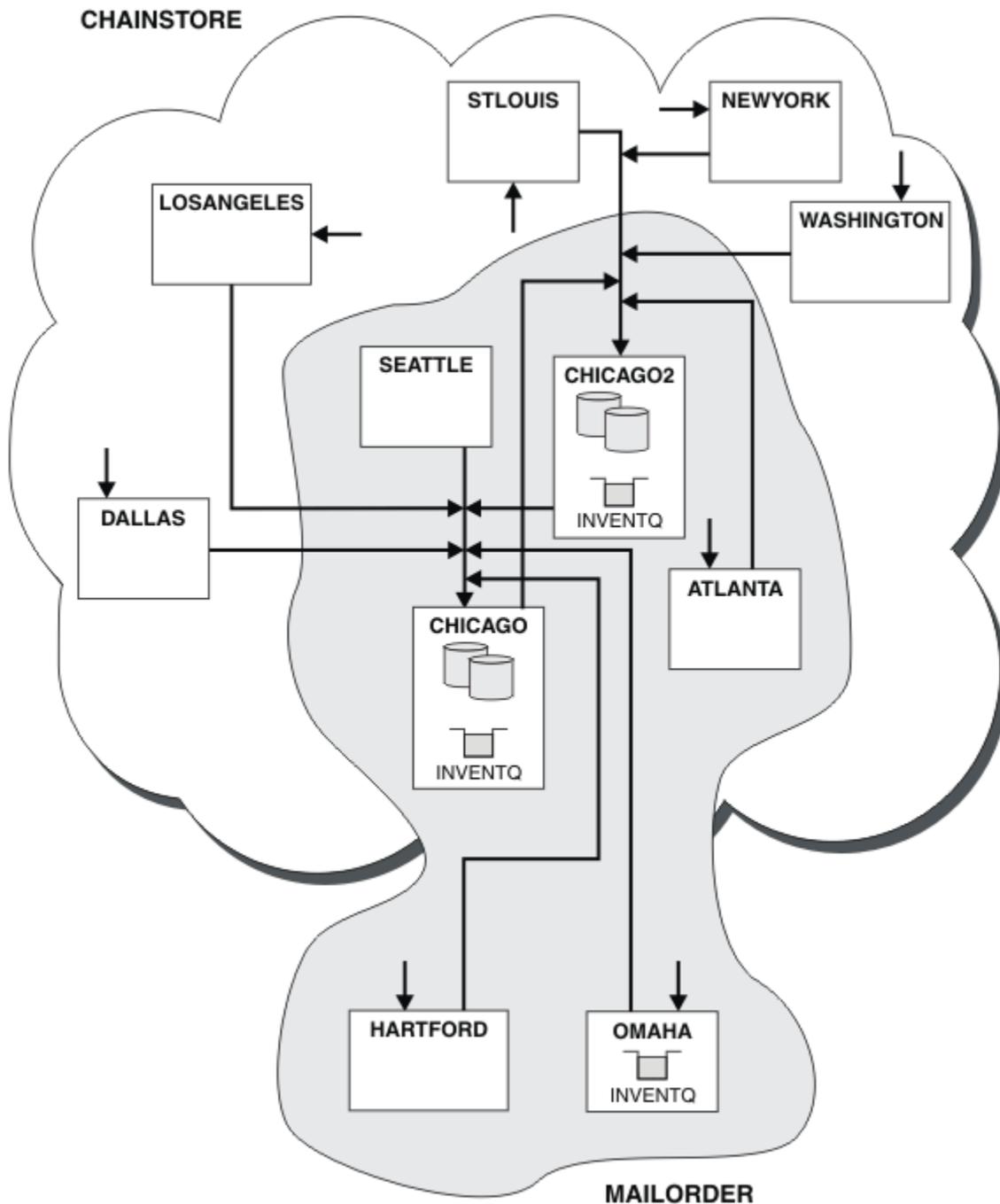


図 47. 相互接続されたクラスター

### 次のタスク

MAILORDER クラスターと CHNSTORE クラスターをマージして、CHNSTORE と呼ばれる 1 つの大型クラスターを形成することにしたと想定します。

CHNSTORE クラスターと MAILORDER クラスター、つまり、CHICAGO と CHICAGO2 が完全リポジトリを保有するようなクラスターをマージするには、以下のようにします。

- CHICAGO と CHICAGO2 のキュー・マネージャー定義を変更し、名前リスト (CHAINMAIL) を指定している REPOSNL 属性を削除して、この属性をクラスター名 (CHNSTORE) を指定している REPOS 属性と置き換えます。以下に例を示します。

```
ALTER QMGR(CHICAGO) REPOSNL(' ') REPOS(CHNSTORE)
```

- MAILORDER クラスターの各キュー・マネージャーで、すべてのチャンネル定義およびキュー定義を変更して、CLUSTER 属性の値を MAILORDER から CHNSTORE に変更します。例えば、HARTFORD で、次のように入力します。

```
ALTER CHANNEL(MAILORDER.HARTFORD) CLUSTER(CHNSTORE)
```

OMAHA で、次のコマンドを入力します。

```
ALTER QLOCAL(MORDERQ) CLUSTER(CHNSTORE)
```

- CHAINMAIL というクラスター名前リストを指定するすべての定義 (つまり CHICAGO、CHICAGO2、SEATTLE、ATLANTA での CLUSRCVR チャンネル定義および CLUSSDR チャンネル) を変更して、その代わりに CHNSTORE クラスターを指定します。

この例から、名前リストを使用する利点が理解できます。CHICAGO および CHICAGO2 のキュー・マネージャー定義を変更する代わりに、CHAINMAIL 名前リストの値を変更できます。同様に CLUSRCVR および CLUSSDR チャンネル定義を CHICAGO、CHICAGO2、SEATTLE、ATLANTA で変更する代わりに、名前リストを変更することによって必要な結果を得ることができます。

## 関連タスク

クラスター・ネットワークの削除

ネットワークからクラスターを削除し、分散キューイング構成を復元します。

## クラスター・ネットワークの削除

ネットワークからクラスターを削除し、分散キューイング構成を復元します。

## 始める前に

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリーが常に使用可能でなければなりません。この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 277 ページの『[既存のネットワークのクラスターへの変換](#)』での説明に従って、IBM MQ クラスターを設定しました。
- 今度は、このクラスターをシステムから除去します。キュー・マネージャーのネットワークは、クラスターが設定される前と同じように引き続き機能します。

## このタスクについて

以下の手順に従って、クラスター・ネットワークを除去します。

## 手順

1. CHNSTORE クラスターからクラスター・キューを削除します。

CHICAGO と CHICAGO2 の両方で、キュー INVENTQ のローカル・キュー定義を変更して、クラスターからキューを削除します。次のコマンドを実行します。

```
ALTER QLOCAL(INVENTQ) CLUSTER(' ')
```

キューを変更すると、完全リポジトリ内の情報が更新され、クラスター全体に伝搬されます。DEFBIND(NOTFIXED)によってキューが定義されている、MQOO\_BIND\_NOT\_FIXEDを使用するアクティブ・アプリケーション、およびMQOO\_BIND\_AS\_Q\_DEFを使用するアプリケーションは、次にMQPUT呼び出しまたはMQPUT1呼び出しの試行に失敗します。理由コードMQRC\_UNKNOWN\_OBJECT\_NAMEが戻されます。

ステップ1を最初に行う必要はありませんが、最初に行わない場合はステップ4の後で行ってください。

## 2. クラスター・キューにアクセスするすべてのアプリケーションを停止する

クラスター・キューにアクセスするすべてのアプリケーションを停止してください。そうしないと、ステップ5でクラスターをリフレッシュすると、一部のクラスター情報がローカル・キュー・マネージャー上に残ることがあります。この情報は、すべてのアプリケーションが停止し、クラスター・チャンネルが切断されたときに除去されます。

## 3. 完全リポジトリ・キュー・マネージャーからリポジトリ属性を削除する

CHICAGOとCHICAGO2の両方で、キュー・マネージャーの定義を変更してリポジトリ属性を削除します。これを実行するには、次のコマンドを入力します。

```
ALTER QMGR REPOS(' ')
```

キュー・マネージャーは、完全リポジトリを保有しなくなったことをクラスター内の他のキュー・マネージャーに通知します。他のキュー・マネージャーがこの情報を受け取ると、完全リポジトリが終了したことを示すメッセージが表示されます。さらに、クラスターCHNSTOREに関して利用できないリポジトリがなくなったことを示す1つまたは複数のメッセージが表示されます。

## 4. クラスター・チャンネルを削除する

CHICAGOでクラスター・チャンネルを削除するために、次のコマンドを発行します。

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) CLUSTER(' ')  
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR) CLUSTER(' ')
```

**注:**最初にCLUSSDRコマンド、次にCLUSRCVRコマンドの順序で発行することが重要です。最初にCLUSRCVRコマンド、次にCLUSSDRコマンドの順序で発行しないでください。これを行うと、STOPPED状況の未確定チャンネルが作成されます。この場合は、START CHANNELコマンドを発行して、停止したチャンネルをリカバリーする必要があります(例えば、START CHANNEL(CHNSTORE.CHICAGO))。

クラスターCHNSTOREに関するリポジトリがないことを示すメッセージが表示されます。

手順1で説明されているようにクラスター・キューを削除していない場合は、ここで削除してください。

## 5. クラスター・チャンネルを停止する

CHICAGOでクラスター・チャンネルを停止するために、次のコマンドを発行します。

```
STOP CHANNEL(CHNSTORE.CHICAGO2)  
STOP CHANNEL(CHNSTORE.CHICAGO)
```

## 6. クラスター内のキュー・マネージャーごとに手順4と5を繰り返す

## 7. クラスター・チャンネルを停止し、クラスター・チャンネルおよびクラスター・キューに関するすべての定義をそれぞれのキュー・マネージャーから除去します。

## 8. オプション: キュー・マネージャーによって保持されるクラスター情報のキャッシュの消去

キュー・マネージャーがクラスターのメンバーでなくなっても、各キュー・マネージャーにはクラスターに関する情報のキャッシュ・コピーが保存されています。このデータを削除する場合は、[314 ページの『キュー・マネージャーのクラスター化以前の状態への復元』](#)に説明されている作業を参照してください。

## 9. INVENTQのリモート・キュー定義を置き換える

ネットワークが引き続き機能するように、キュー・マネージャーごとに INVENTQ のリモート・キュー定義を置き換えてください。

#### 10. クラスタを整理する

必要のなくなったキューやチャンネルの定義をすべて削除してください。

#### 関連タスク

##### 相互接続された新しいクラスタの追加

既存のクラスタと一部のキュー・マネージャーを共有する新しいクラスタを追加します。

## ゲートウェイ・キュー・マネージャーを使用した2つのオーバーラップするクラスタの作成

このタスクの手順に従って、ゲートウェイ・キュー・マネージャーを使用したオーバーラップするクラスタを構成します。このクラスタを、以降で説明するあるメッセージへのメッセージをクラスタ内の他のアプリケーションへのメッセージから分離する例の開始点として使用します。

### このタスクについて

286 ページの図 48 に、クラスタ・メッセージ・トラフィックの分離方法を説明するために使用するクラスタ構成例を示します。この例は、クラスタ化: 複数のクラスタ伝送キューの使用によるアプリケーションの分離で説明されています。

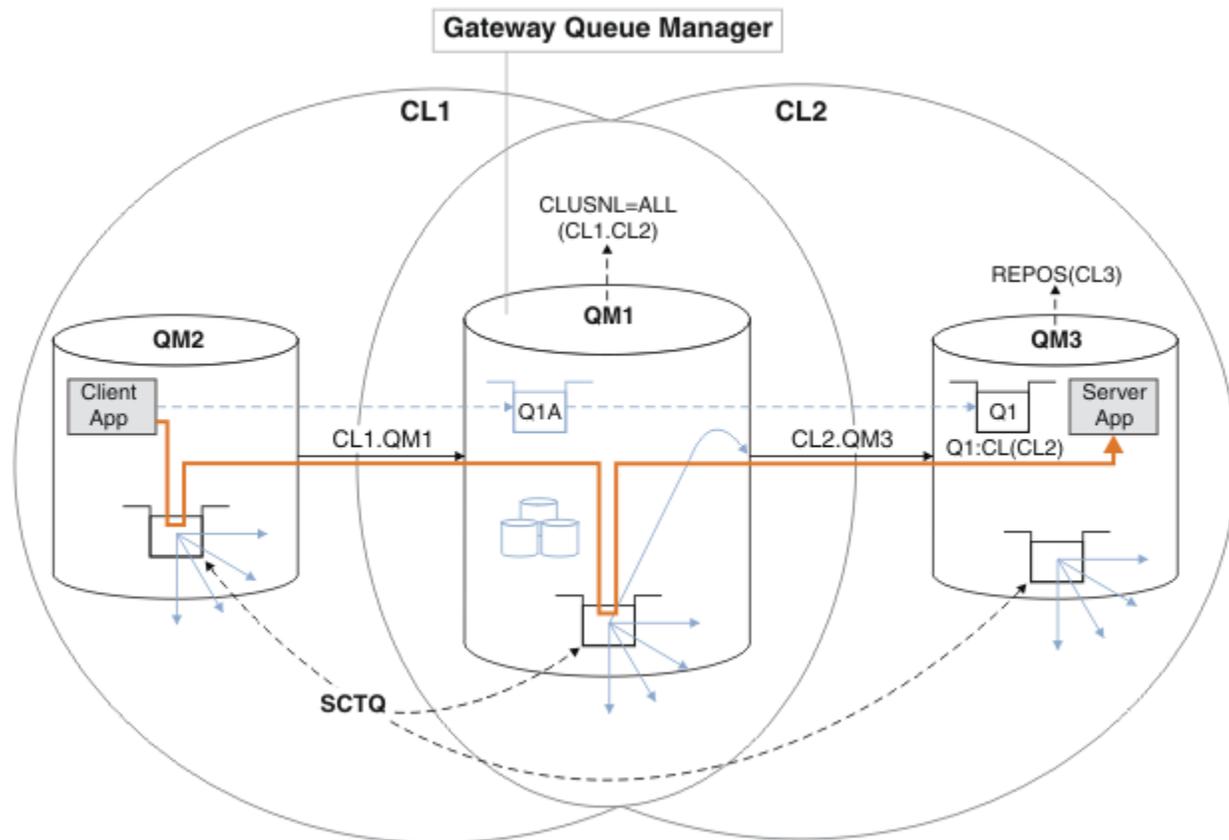


図 48. ハブ・スポーク・アーキテクチャにデプロイされた、IBM MQ クラスタを使用するクライアント/サーバー・アプリケーション

例を構築するためのステップの数を可能な限り少なくするため、構成は、現実的というよりも、むしろ簡素なものにとどめられています。この例は、2つの別々の組織で作成された2つのクラスタの統合を表しています。より現実的なシナリオについては、クラスタ化: クラスタ伝送キューの構成方法の計画を参照してください。

手順に従って、クラスターを構成します。これらのクラスターが、以降で説明するクライアント・アプリケーションからサーバー・アプリケーションへのメッセージ・トラフィックを分離する例で使用されます。

手順では、各クラスターが2つのリポジトリを使用できるように、いくつかのキュー・マネージャーを追加します。ゲートウェイ・キュー・マネージャーは、パフォーマンス上の理由から、リポジトリとして使用されません。

## 手順

1. QM1、QM2、QM3、QM4、QM5 の各キュー・マネージャーを作成して始動します。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QM n
strmqm QmgrName
```

注：QM4 および QM5 は、クラスターのバックアップ完全リポジトリです。

2. キュー・マネージャーごとにリスナーを定義して開始します。

```
*... On QM n
DEFINE LISTENER(TCP141 n) TRPTYPE(TCP) IPADDR(hostname) PORT(141 n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141 n)
```

3. すべてのクラスターのクラスター名前リストを作成します。

```
*... On QM1
DEFINE NAMELIST(ALL) NAMES(CL1, CL2) REPLACE
```

4. QM2 と QM4 を CL1 の完全リポジトリにし、QM3 と QM5 を CL2 の完全リポジトリにします。

- a) CL1 の場合:

```
*... On QM2 and QM4
ALTER QMGR REPOS(CL1) DEFCLXQ(SCTQ)
```

- b) CL2 の場合:

```
*... On QM3 and QM5
ALTER QMGR REPOS(CL2) DEFCLXQ(SCTQ)
```

5. キュー・マネージャーとクラスターのそれぞれに、クラスター送信側チャンネルとクラスター受信側チャンネルを追加します。

QM2、QM3、QM4、および QM5 で以下のコマンドを実行します。ここで、*c*、*n*、および *m* は、[287 ページの表 24](#) に示された各キュー・マネージャーに対応する値を取ります。

表 24. クラスター 1 および 2 を作成する際のパラメーター値

キュー・マネージャー	クラスター <i>c</i>	その他のリポジトリ <i>n</i>	このリポジトリ <i>m</i>
QM2	1	4	2
QM4	1	2	4
QM3	2	5	3
QM5	2	3	5

```
*... On QM m
DEFINE CHANNEL(CL c.QM n) CHLTYPE(CLUSSDR) CONNAME('localhost(141 n)') CLUSTER(CL c) REPLACE
DEFINE CHANNEL(CL c.QM m) CHLTYPE(CLUSRCVR) CONNAME('localhost(141 m)') CLUSTER(CL c) REPLACE
```

6. 各クラスターに、ゲートウェイ・キュー・マネージャー QM1 を追加します。

```
*... On QM1
DEFINE CHANNEL(CL1.QM2) CHLTYPE(CLUSSDR) CONNAME('localhost(1412)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL1.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL2.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL2) REPLACE
DEFINE CHANNEL(CL2.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL2) REPLACE
```

7. クラスター CL2 内のキュー・マネージャー QM3 に、ローカル・キュー Q1 を追加します。

```
*... On QM3
DEFINE QLOCAL(Q1) CLUSTER(CL2) REPLACE
```

8. ゲートウェイ・キュー・マネージャーに、クラスター化されたキュー・マネージャー別名 Q1A を追加します。

```
*... On QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(Queue) DEFBIND(NOTFIXED) REPLACE
```

**注:** QM1 以外のすべてのキュー・マネージャーでキュー・マネージャー別名を使用するアプリケーションは、別名キューをオープンするときに DEFBIND(NOTFIXED) を指定する必要があります。DEFBIND は、キューがアプリケーションによってオープンされるたびに、メッセージ・ヘッダー内のルーティング情報を固定するかどうかを指定します。これがデフォルト値の OPEN に設定されている場合、メッセージは Q1@QM1 にルーティングされます。Q1@QM1 は存在しないため、他のキュー・マネージャーからのメッセージは、送達不能キューに入れられることとなります。このキュー属性を DEFBIND(NOTFIXED) に設定することで、キュー設定をデフォルトで DEFBIND に設定する **amqsput** などのアプリケーションが正しく動作します。

9. ゲートウェイ・キュー・マネージャー QM1 に、すべてのクラスター化されたキュー・マネージャーのクラスター・キュー・マネージャー別名定義を追加します。

```
*... On QM1
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSNL(ALL) REPLACE
```

**ヒント:** ゲートウェイ・キュー・マネージャーのキュー・マネージャー別名定義は、別のクラスター内のキュー・マネージャーを参照するメッセージを転送します ([キュー・マネージャー別名とクラスターを参照](#))。

## 次のタスク

1. キュー別名定義をテストするために、キュー別名定義 Q1A を使用して、メッセージを QM2 から QM3 上の Q1 に送信します。

- a. QM2 でサンプル・プログラム **amqsput** を実行して、メッセージを書き込みます。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

- b. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
```

no more messages  
Sample AMQSGETO end

2. キュー・マネージャー別名定義をテストするために、要求メッセージを送信し、一時動的応答キューで応答メッセージを受信します。

以下の図に、一時動的キュー RQ に返される応答メッセージが辿るパスを示します。サーバー・アプリケーションは、QM3 に接続され、キュー・マネージャー名 QM2 を使用して、応答キューをオープンします。キュー・マネージャー名 QM2 は、QM1 上でクラスター・キュー・マネージャー別名として定義されます。QM3 は、応答メッセージを QM1 にルーティングします。QM1 は、そのメッセージを QM2 にルーティングします。

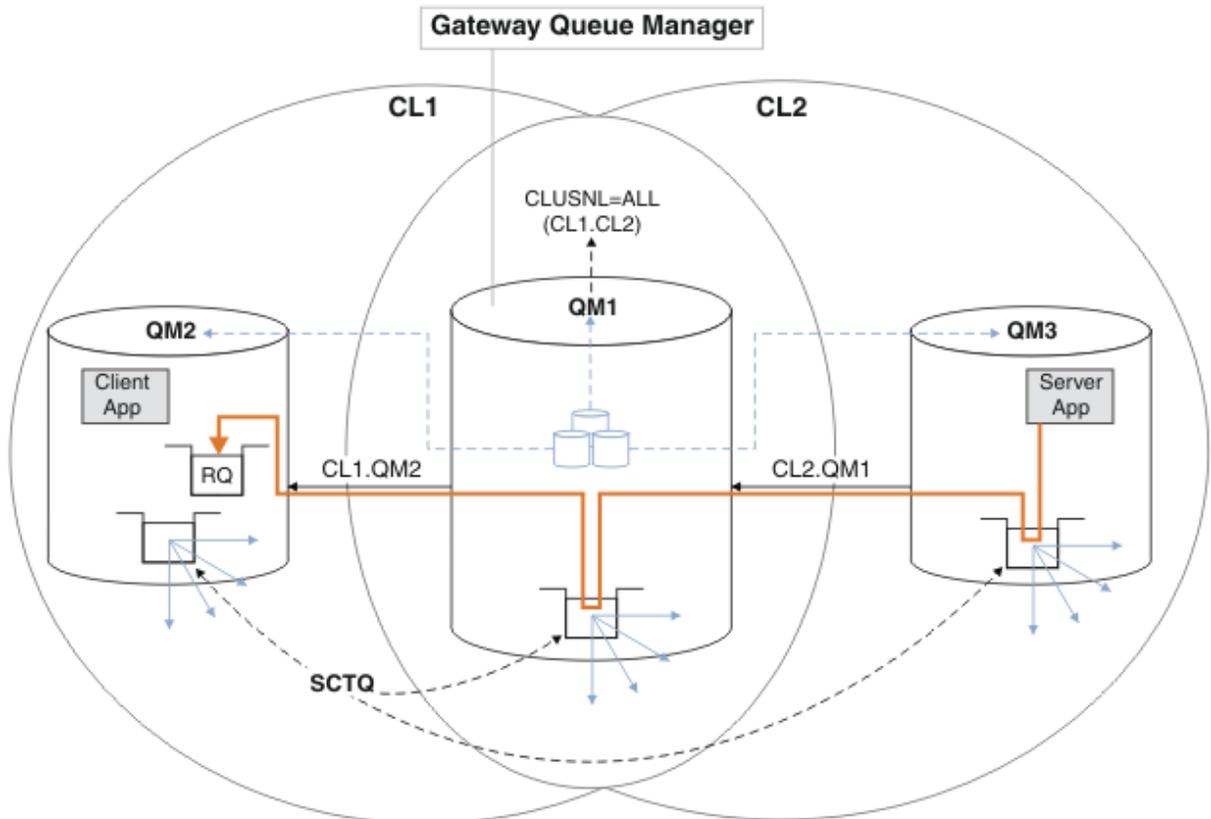


図 49. 応答メッセージを異なるクラスターに返すためのキュー・マネージャー別名の使用

このルーティングは、次のようにして行われます。QM1 には、各クラスター内のすべてのキュー・マネージャーのキュー・マネージャー別名定義があります。これらの別名は、すべてのクラスターでクラスター化されます。それぞれの別名からキュー・マネージャーへと向かうグレーの破線矢印は、各キュー・マネージャー別名が、クラスターのうちの少なくとも 1 つにある実際のキュー・マネージャーに解決されることを示します。この例では、QM2 別名が CL1 と CL2 の両方のクラスターでクラスター化されています。この別名は、CL1 内にある実際のキュー・マネージャー QM2 に解決されます。サーバー・アプリケーションは、キュー名 RQ への応答、およびキュー・マネージャー名 QM2 への応答を使用して応答メッセージを作成します。このメッセージは、QM1 にルーティングされます。これは、キュー・マネージャー別名定義 QM2 はクラスター CL2 内の QM1 に定義されており、キュー・マネージャー QM2 はクラスター CL2 のメンバーではないためです。メッセージをターゲット・キュー・マネージャーに送信できないことから、この別名定義を持つキュー・マネージャーにメッセージが送信されます。

QM1 は、QM2 に転送するために、メッセージを QM1 上のクラスター伝送キューに配置します。QM1 での QM2 のキュー・マネージャー別名定義は、QM2 を実際のターゲット・キューとして定義しているため、QM1 はメッセージを QM2 にルーティングします。別名定義が参照できるのは、実際の定義だけで、自身を参照することはできないため、この定義は循環しません。QM1 と QM2 は両方とも同じクラスター CL1 内にあることから、実際の定義は QM1 によって解決されます。QM1 は、CL1 のリポジトリか

ら QM2 の接続情報を見つけ出し、メッセージを QM2 にルーティングします。メッセージを QM1 によって転送するためには、サーバー・アプリケーションが、オプション DEFBIND を MQBND\_BIND\_NOT\_FIXED に設定して応答キューをオープンしていなければなりません。サーバー・アプリケーションがオプション MQBND\_BIND\_ON\_OPEN を設定して応答キューをオープンしている場合、メッセージは転送されず、送達不能キューに入れられます。

- a. QM3 に、トリガーを設定したクラスター化された要求キューを作成します。

```
*... On QM3
DEFINE QLOCAL(QR) CLUSTER(CL2) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
```

- b. ゲートウェイ・キュー・マネージャー QM1 に、QR のクラスター化されたキュー別名定義を作成します。

```
*... On QM1
DEFINE QALIAS(QRA) CLUSNL(ALL) TARGET(QR) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

- c. QM3 に、サンプル・エコー出力プログラム **amqsech** を開始するためのプロセス定義を作成します。

```
*... On QM3
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

- d. QM2 に、サンプル・プログラム **amqsreq** のモデル・キューを作成し、一時動的応答キューを作成します。

```
*... On QM2
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

- e. キュー・マネージャー別名定義をテストするために、キュー別名定義 QRA を使用して、要求を QM2 から QM3 上の QR に送信します。

- i) QM3 でトリガー・モニター・プログラムを実行します。

```
runmqtrm -m QM3
```

出力は次のとおりです。

```
C:\IBM\MQ>runmqtrm -m QM3
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.
01/02/2012 16:17:15: IBM MQ trigger monitor started.
```

```
-----
01/02/2012 16:17:15: Waiting for a trigger message
```

- ii) QM2 でサンプル・プログラム **amqsreq** を実行して要求を書き込み、応答を待機します。

```
C:\IBM\MQ>amqsreq QRA QM2
Sample AMQSREQ0 start
server queue is QRA
replies to 4F2961C802290020
A request message from QM2 to QR on QM3
```

```
response <A request message from QM2 to QR on QM3>
no more replies
Sample AMQSREQ0 end
```

## 関連タスク

262 ページの『[クラスターへのキュー・マネージャーの追加: 別個の伝送キュー](#)』

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

## 関連情報

[アクセス制御と複数のクラスター伝送キュー](#)

[クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離](#)

[クラスター化: クラスター伝送キューの構成方法の計画](#)

## リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。このソリューションは、クラスター化されたキュー・リモート定義と、別個の送信側チャンネルおよび伝送キューを使用します。

## 始める前に

286 ページの『[ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成](#)』の [ハブ・スポーク・アーキテクチャーにデプロイされた IBM MQ クラスターを使用するクライアント/サーバー・アプリケーション](#) に示されているオーバーラップするクラスターを、タスクで説明している手順に従って作成します。

## このタスクについて

このソリューションは、分散キューイングを使用して、Server App アプリケーションのメッセージをゲートウェイ・キュー・マネージャーの他のメッセージ・トラフィックから分離します。メッセージを他別の伝送キューおよび他のチャンネルに方向転換するには、QM1 にクラスター化されたリモート・キュー定義を定義しなければなりません。リモート・キュー定義には、QM3 上の Q1 へのメッセージのみを保管する、特定の伝送キューへの参照を組み込む必要があります。292 ページの [図 50](#) では、クラスター・キュー別名 Q1A がリモート・キュー定義 Q1R で補足され、伝送キューと送信側チャンネルが追加されています。

このソリューションでは、すべての応答メッセージが共通 SYSTEM.CLUSTER.TRANSMIT.QUEUE を使用して返されます。

このソリューションには、同じクラスター内の同じキュー・マネージャー上にある複数の宛先キューへのトラフィックを簡単に分離できるという利点があります。このソリューションの欠点は、それぞれに異なるキュー・マネージャー上にある Q1 の複数のコピーの間でクラスター・ワークロード・バランシングを使用できないことです。この欠点を克服するには、293 ページの『[クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する](#)』を参照してください。また、ある伝送キューから別の伝送キューへの切り替えも管理する必要があります。

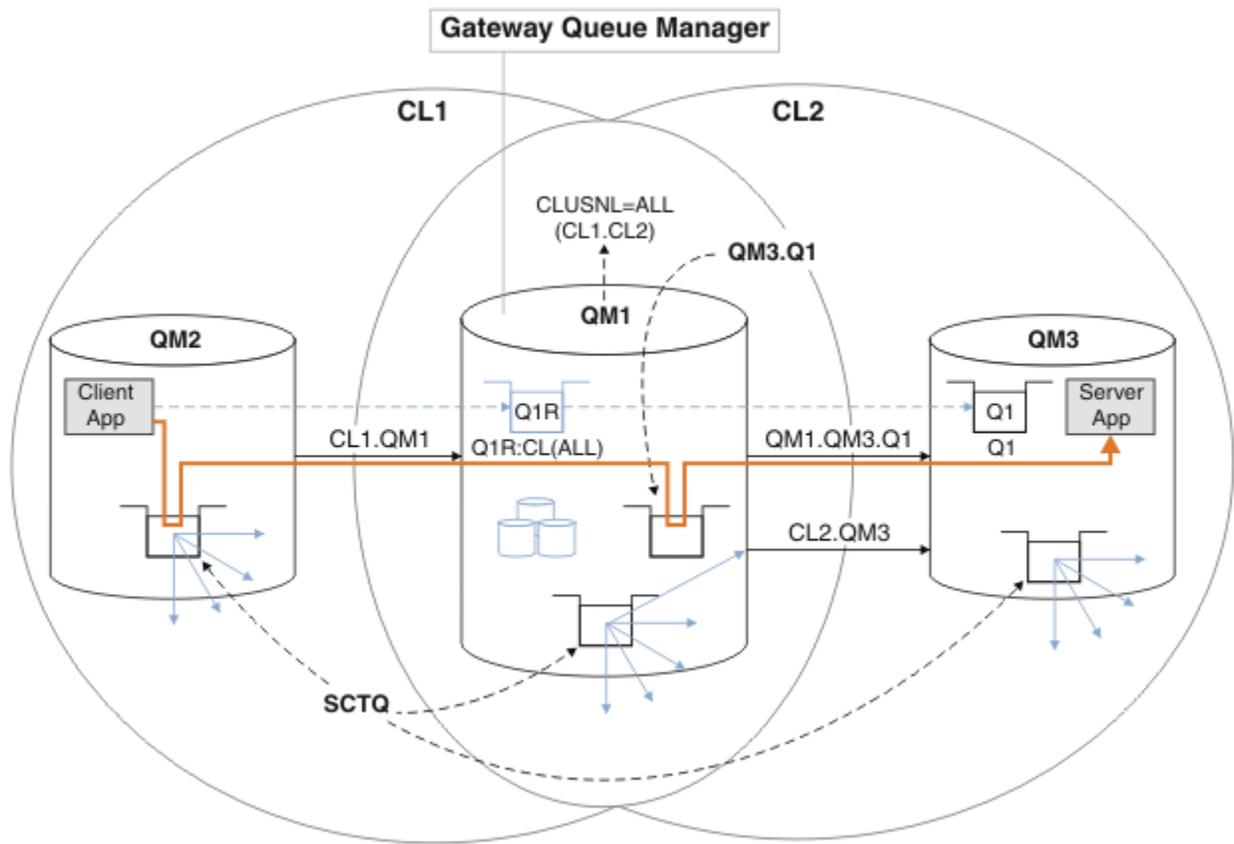


図 50. リモート・キュー定義を使用してハブ・スポーク・クラスター・アーキテクチャーにデプロイされたクライアント/サーバー・アプリケーション

## 手順

1. Q1 へのメッセージ・トラフィックをゲートウェイ・キュー・マネージャーから分離するためのチャンネルを作成します。
  - a) ゲートウェイ・キュー・マネージャー QM1 に、ターゲット・キュー・マネージャー QM3 への送信側チャンネルを作成します。

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(SDR) CONNAME(QM3HostName(1413)) XMITQ(QM3.Q1) REPLACE
```

- b) ターゲット・キュー・マネージャー QM3 に、受信側チャンネルを作成します。

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(RCVR) REPLACE
```

2. ゲートウェイ・キュー・マネージャーに、Q1 へのメッセージ・トラフィック用の伝送キューを作成します。

```
DEFINE QLOCAL(QM3.Q1) USAGE(XMITQ) REPLACE
START CHANNEL(QM1.QM3.Q1)
```

伝送キューに関連付けるチャンネルを開始すると、伝送キューがそのチャンネルに関連付けられます。伝送キューがチャンネルに関連付けられると、チャンネルが自動的に開始します。

3. ゲートウェイ・キュー・マネージャー上の Q1 のクラスター化されたキュー別名定義を、クラスター化されたリモート・キュー定義で補足します。

```
DEFINE QREMOTE CLUSNL(ALL) RNAME(Q1) RQMNAME(QM3) XMITQ(QM3.Q1) REPLACE
```

## 次のタスク

構成をテストするために、ゲートウェイ・キュー・マネージャー QM1 上のクラスター化されたキュー・リモート定義 Q1R を使用して、メッセージを QM2 から QM3 上の Q1 にメッセージを送信します。

1. QM2 でサンプル・プログラム **amqsput** を実行して、メッセージを書き込みます。

```
C:>IBM>MQ>amqsput Q1R QM2
Sample AMQSPUT0 start
target queue is Q1R
Sample request message from QM2 to Q1 using Q1R
```

```
Sample AMQSPUT0 end
```

2. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

```
C:>IBM>MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1R>
no more messages
Sample AMQSGET0 end
```

## 関連タスク

[クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する](#)

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の 1 つのキュー・マネージャーに分離します。

[クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する](#)

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

[クラスター伝送キューを区別するようにデフォルトを変更して、メッセージ・トラフィックを分離する](#)

キュー・マネージャーがクラスター・キューまたはトピックへのメッセージを伝送キューに保管するデフォルトの方法を変更できます。デフォルトを変更することは、ゲートウェイ・キュー・マネージャーでクラスター・メッセージを分離する手段となります。

262 ページの『[クラスターへのキュー・マネージャーの追加: 別個の伝送キュー](#)』

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

## 関連情報

[クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離](#)

[クラスター化: クラスター伝送キューの構成方法の計画](#)

[アクセス制御と複数のクラスター伝送キュー](#)

## クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の 1 つのキュー・マネージャーに分離します。

## 始める前に

1. ゲートウェイ・キュー・マネージャーが IBM WebSphere MQ 7.5 以降にある必要があります。
2. 286 ページの『[ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成](#)』のハブ・スポーク・アーキテクチャーにデプロイされた IBM MQ クラスターを使用するクライアント/サーバー・アプリケーションに示されているオーバーラップするクラスターを、タスクで説明している手順に従って作成します。

## このタスクについて

ゲートウェイ・キュー・マネージャー QM1 で、伝送キューを追加して、そのキュー属性 CLCHNAME を設定します。CLCHNAME は、QM3 上のクラスター受信側チャンネルの名前に設定します (295 ページの図 51 を参照)。

このソリューションには、291 ページの『[リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する](#)』で説明したソリューションに勝る数々の利点があります。

- 追加しなければならない定義の数が少なくなります。
- 同じクラスター CL2 内の異なるキュー・マネージャー上にあるターゲット・キュー Q1 の複数のコピーの間でのワークロード・บาลancingをサポートします。
- チャンネルが再始動すると、ゲートウェイ・キュー・マネージャーが自動的に新しい構成に切り替え、メッセージが失われることはありません。
- ゲートウェイ・キュー・マネージャーは、引き続きメッセージを受信した順に転送します。これは、QM3 のキュー Q1 へのメッセージがまだ SYSTEM.CLUSTER.TRANSMIT.QUEUE にあるときに切り替えが発生した場合にも該当します。

295 ページの図 51 でクラスター・メッセージ・トラフィックを分離するための構成では、291 ページの『[リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する](#)』で紹介されているリモート・キューを使用した構成ほど効果的にトラフィックを分離することはできません。クラスター CL2 内のキュー・マネージャー QM3 が多数の異なるクラスター・キューとサーバー・アプリケーションをホストしている場合、これらのキューのすべてが、QM1 を QM3 に接続するクラスター・チャンネル CL2.QM3 を共有します。その他のフローは、295 ページの図 51 のグレイの矢印で示されています。この矢印は、SYSTEM.CLUSTER.TRANSMIT.QUEUE からクラスター送信側チャンネル CL2.QM3 への潜在的なクラスター・メッセージ・トラフィックを表しています。

解決策は、キュー・マネージャーが特定のクラスター内で 1 つのクラスター・キューだけをホストするように制限することです。キュー・マネージャーがすでに多数のクラスター・キューをホストしている場合には、この制限に対応するために、別のキュー・マネージャーを作成するか、または別のクラスターを作成する必要があります (297 ページの『[クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する](#)』を参照)。

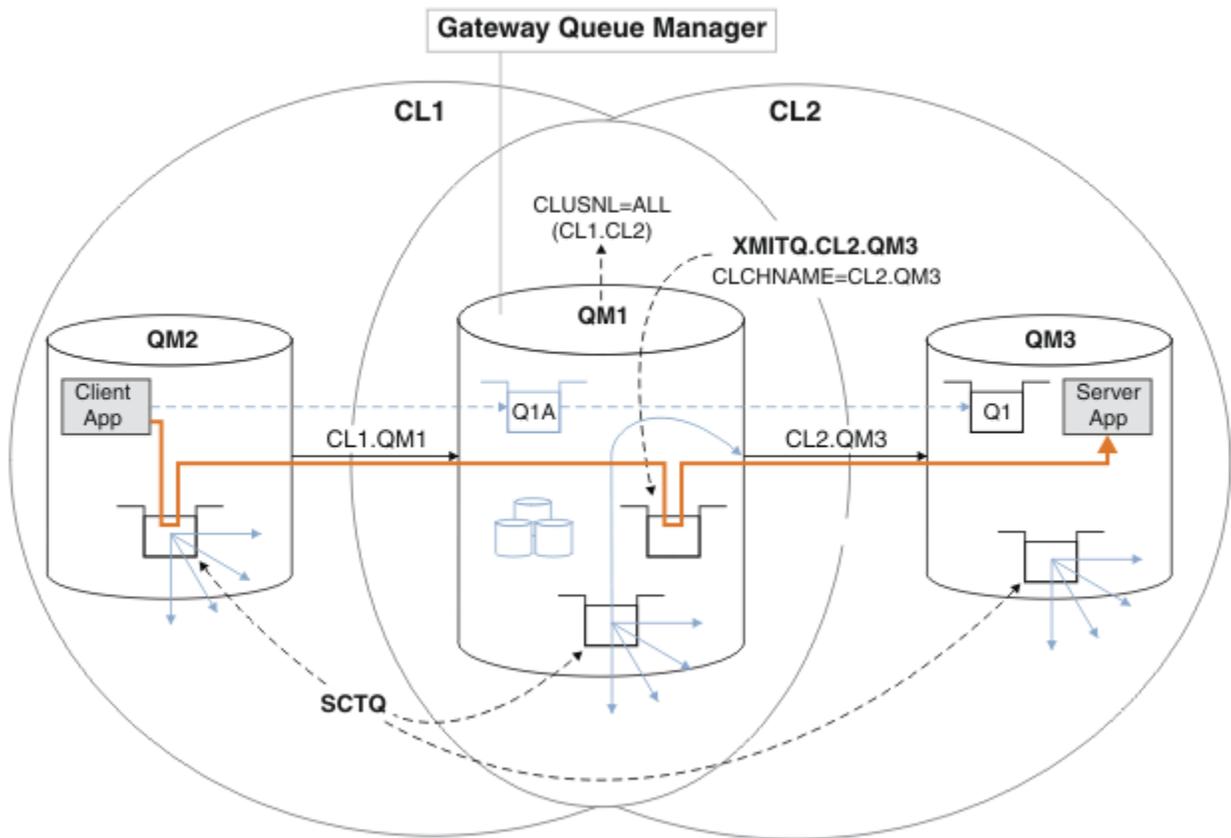


図 51. 追加のクラスター伝送キューを使用してハブ・スポーク・アーキテクチャーにデプロイされたクライアント/サーバー・アプリケーション

## 手順

1. ゲートウェイ・キュー・マネージャー QM1 で、クラスター送信側チャンネル CL2.QM3 用の追加のクラスター伝送キューを作成します。

```
*... on QM1
DEFINE QLOCAL(XMITQ.CL2.QM3) USAGE(XMITQ) CLCHNAME(CL2.QM3)
```

2. 伝送キュー XMITQ.CL2.QM3 を使用するように切り替えます。

- a) クラスター送信側チャンネル CL2.QM3 を停止します。

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

コマンドが受け入れられたという応答が返されます。

AMQ8019: Stop IBM MQ channel accepted.

- b) チャンネル CL2.QM3 が停止していることを確認します。

チャンネルが停止しない場合は、FORCE オプションを指定した **STOP CHANNEL** コマンドを再度実行できます。FORCE オプションを設定する一例は、チャンネルが停止しないため、そのチャンネルを同期化するもう一方のキュー・マネージャーを再始動できない場合です。

```
*... On QM1
start
```

応答は、チャンネル状況の要約です。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)             STATUS(STOPPED)
サブステート(MQGET) XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

c) チャンネル CL2.QM3 を開始します。

```
*... On QM1
START CHANNEL(CL2.QM3)
```

コマンドが受け入れられたという応答が返されます。

```
AMQ8018: Start IBM MQ channel accepted.
```

d) チャンネルが開始したことを確認します。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

応答は、チャンネル状況の要約です。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)             STATUS(RUNNING)
SUBSTATE(MQGET)          XMITQ(XMITQ.CL2.QM3)
```

e) 伝送キューが切り替えられたことを確認します。

メッセージ " アンカル 7341 チャンネル CL2.QM3 の伝送キューは XMITQ です。 " CL2"第 3 四半期"のゲートウェイ・キュー・マネージャーのエラー・ログをモニターしてください。

## 次のタスク

単独の伝送キューをテストするために、キュー別名定義 Q1A を使用して、メッセージを QM2 から QM3 上の Q1 に送信します。

1. QM2 でサンプル・プログラム **amqsput** を実行して、メッセージを書き込みます。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

2. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
```

```
no more messages
Sample AMQSGETO end
```

## 関連概念

### [241 ページの『クラスター伝送キューとクラスター送信側チャンネルの操作』](#)

クラスター・キュー・マネージャー間のメッセージは、クラスター伝送キューに保管されてから、クラスター送信側チャンネルによって転送されます。どの時点でも、クラスター送信側チャンネルが関連付けられる伝送キューは1つです。チャンネルの構成を変更すると、そのチャンネルは次回開始したときに、別の伝送キューに切り替わる可能性があります。この切り替え処理は自動化されていて、トランザクションで実行されます。

## 関連タスク

[リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する](#)

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。このソリューションは、クラスター化されたキュー・リモート定義と、別個の送信側チャンネルおよび伝送キューを使用します。

[クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する](#)

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

[クラスター伝送キューを区別するようにデフォルトを変更して、メッセージ・トラフィックを分離する](#)  
キュー・マネージャーがクラスター・キューまたはトピックへのメッセージを伝送キューに保管するデフォルトの方法を変更できます。デフォルトを変更することは、ゲートウェイ・キュー・マネージャーでクラスター・メッセージを分離する手段となります。

### [262 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』](#)

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

## 関連情報

[アクセス制御と複数のクラスター伝送キュー](#)

[クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離](#)

[クラスター化: クラスター伝送キューの構成方法の計画](#)

## クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

## 始める前に

このタスクで説明する手順は、[295 ページの図 51](#) に示されている構成を変更するために作成されています。

1. [ゲートウェイ・キュー・マネージャーが IBM WebSphere MQ 7.5 以降にある必要があります。](#)
2. [286 ページの『ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成』](#) のハブ・スポーク・アーキテクチャーにデプロイされた [IBM MQ クラスターを使用するクライアント/サーバー・アプリケーション](#) に示されているオーバーラップするクラスターを、タスクで説明している手順に従って作成します。
3. [293 ページの『クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する』](#) の [295 ページの図 51](#) に示されている手順に従

って、クラスターが追加されていないソリューションを作成します。これを、このタスクの手順のベースとして使用します。

## このタスクについて

293 ページの『[クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する](#)』で説明した、単一アプリケーションへのメッセージ・トラフィックを分離するためのソリューションが有効に働くのは、ターゲット・クラスター・キューがキュー・マネージャー上の唯一のクラスター・キューである場合です。そうでない場合には、2つの選択肢があります。1つはキューを別のキュー・マネージャーに移動すること、もう1つはキューをキュー・マネージャー上の他のクラスター・キューと分離するクラスターを作成することです。

このタスクでは、クラスターを追加してターゲット・キューを分離する手順を説明します。クラスターは、その目的のためだけに追加します。実際には、クラスターおよびクラスターの命名体系を設定するプロセスの中で、体系的に、特定のアプリケーションを分離するタスクに取り組んでください。キューの分離が必要になるたびにクラスターを追加すると、最終的に多数のクラスターを管理しなければならなくなります。このタスクでは、293 ページの『[クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する](#)』の構成を変更して、QM3 上の Q1 を分離するためのクラスター CL3 を追加します。アプリケーションは、この変更作業の初めから終わりまで実行し続けます。

新しい定義と変更された定義は、299 ページの図 52 に強調表示されています。変更内容を要約すると、まず、クラスターを作成します。これは、クラスターの新しい完全リポジトリも作成しなければならないことを意味します。この例では、QM3 が CL3 の完全リポジトリの1つにされます。新しいクラスターにゲートウェイ・キュー・マネージャーを追加するために、QM1 のクラスター送信側チャンネルとクラスター受信側チャンネルを作成します。Q1 の定義を変更して、CL3 に切り替えます。ゲートウェイ・キュー・マネージャーのクラスター名前リストを変更し、新規クラスター・チャンネルを使用するクラスター伝送キューを追加します。最後に、キュー別名 Q1A を新規クラスター名前リストに切り替えます。

IBM MQ は、293 ページの『[クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する](#)』で追加した伝送キュー XMITQ.CL2.QM3 から新規伝送キュー XMITQ.CL3.QM3 に自動的にメッセージを転送することはできません。自動的にメッセージを転送できるのは、両方の伝送キューが同じクラスター送信側チャンネルによって処理される場合のみです。代わりに、このタスクでは、適切だと考えられる手動による切り替え方法の1つを説明します。転送が完了した時点で、QM3 上の他の CL2 クラスター・キューのデフォルト・クラスター伝送キューを使用するように戻すことを選択できます。あるいは、引き続き XMITQ.CL2.QM3 を使用することもできます。デフォルト・クラスター伝送キューに戻すことにした場合、ゲートウェイ・キュー・マネージャーが自動的に切り替えを管理します。

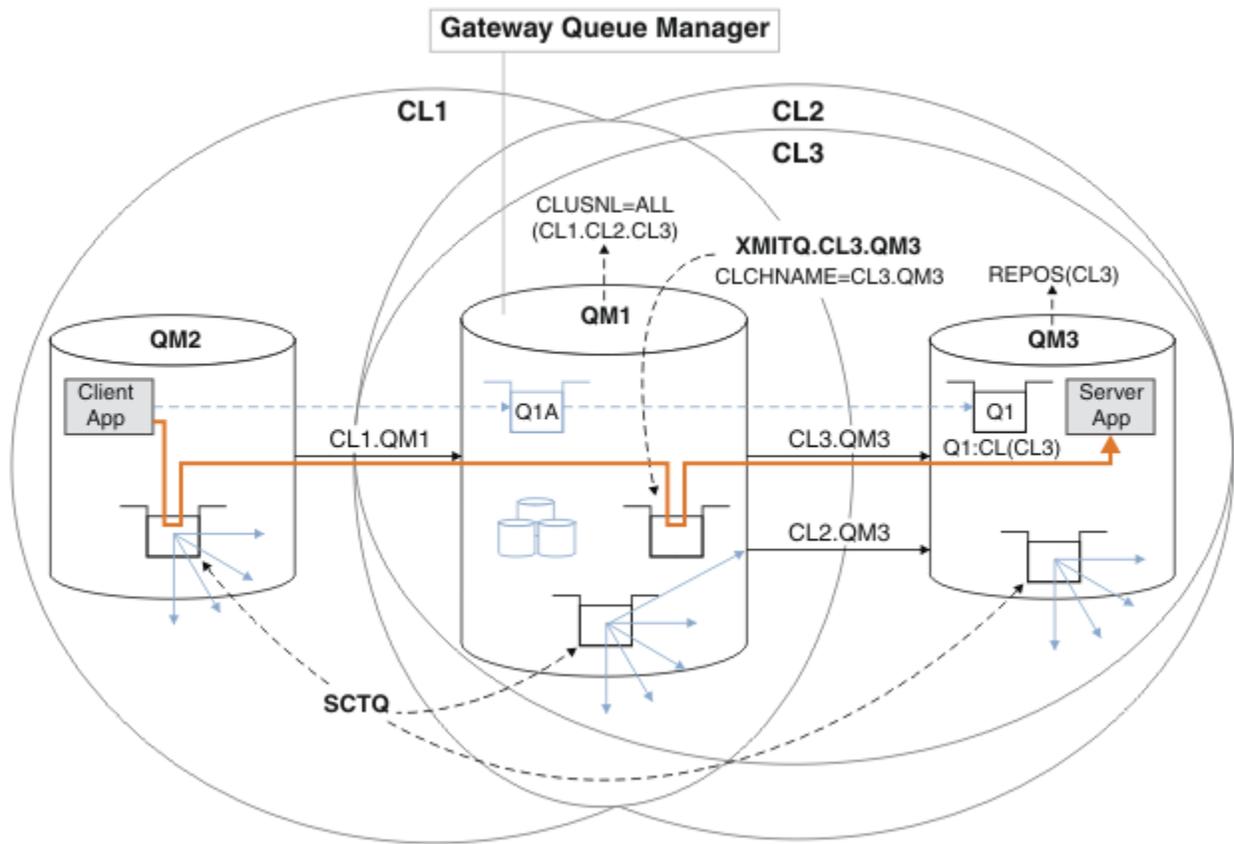


図 52. 追加のクラスターを使用して、同じキュー・マネージャー上の多数のクラスター・キューのうちの 1 つに向かうメッセージ・トラフィックをゲートウェイ・キュー・マネージャー内で分離する

## 手順

1. キュー・マネージャー QM3 および QM5 を変更して、CL2 と CL3 両方のリポジトリにします。

キュー・マネージャーを複数のクラスターのメンバーにするには、キュー・マネージャーがクラスター名前リストを使用して、それがメンバーとなっているクラスターを識別する必要があります。

```
*... On QM3 and QM5
DEFINE NAMELIST(CL23) NAMES(CL2, CL3) REPLACE
ALTER QMGR REPOS(' ') REPOSNL(CL23)
```

2. CL3 のキュー・マネージャー QM3 と QM5 との間のチャンネルを定義します。

```
*... On QM3
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSRCVR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE

*... On QM5
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
```

3. ゲートウェイ・キュー・マネージャーを CL3 に追加します。

ゲートウェイ・キュー・マネージャーを追加するために、QM1 を部分リポジトリとして CL3 に追加します。部分リポジトリを作成するには、クラスター送信側チャンネルとクラスター受信側チャンネルを QM1 に追加します。

さらに、ゲートウェイ・キュー・マネージャーに接続されたすべてのクラスターの名前リストに CL3 を追加します。

```
*... On QM1
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
```

```
DEFINE CHANNEL(CL3.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL3) REPLACE
ALTER NAMELIST(ALL) NAMES(CL1, CL2, CL3)
```

- ゲートウェイ・キュー・マネージャー QM1 に、QM3 の CL3 に向かうメッセージ用のクラスター伝送キューを追加します。

最初に、伝送キューを切り替える準備ができるまで、伝送キューからメッセージを転送するクラスター送信側チャンネルを停止します。

```
*... On QM1
DEFINE QLOCAL(XMITQ.CL3.QM3) USAGE(XMITQ) CLCHNAME(CL3.QM3) GET(DISABLED) REPLACE
```

- 既存のクラスター伝送キュー XMITQ.CL2.QM3 からメッセージを排出します。

以下のサブ手順は、メッセージがゲートウェイ・キュー・マネージャーに到着した順序と一致するように、Q1 内でメッセージの順序を保持することを目的としています。クラスターでは、メッセージの順序は完全に保証されませんが、おそらく順序は保持されます。メッセージの順序を保証することが必要な場合には、アプリケーションでメッセージの順序を定義する必要があります ([メッセージがキューから取り出される順序を参照](#))。

- a) QM3 上のターゲット・キュー Q1 を CL2 から CL3 に変更します。

```
*... On QM3
ALTER QLOCAL(Q1) CLUSTER(CL3)
```

- b) メッセージの配信が開始されるまで、XMITQ.CL3.QM3 をモニターします。

Q1 から CL3 への切り替えがゲートウェイ・キュー・マネージャーに伝搬されると、XMITQ.CL3.QM3 へのメッセージ配信が開始されます。

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL3.QM3) CURDEPTH
```

- c) QM3 上の Q1 への配信を待機中のメッセージがなくなるまで、XMITQ.CL2.QM3 をモニターします。

注: XMITQ.CL2.QM3 は、CL2 のメンバーとなっている他の QM3 上のキューへのメッセージを保管している可能性があります。その場合、キューの項目数はゼロになりません。

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL2.QM3) CURDEPTH
```

- d) 新規クラスター伝送キュー XMITQ.CL3.QM3 からの取得を有効にします。

```
*... On QM1
ALTER QLOCAL(XMITQ.CL3.QM3) GET(ENABLED)
```

6. 前のクラスター伝送キュー XMITQ.CL2.QM3 が必要ない場合には、これを削除します。

QM3 上の CL2 内にあるクラスター・キューへのメッセージは、ゲートウェイ・キュー・マネージャー QM1 上のデフォルト・クラスター伝送キューを使用する状態に戻ります。デフォルト・クラスター伝送キューは、SYSTEM.CLUSTER.TRANSMIT.QUEUE または SYSTEM.CLUSTER.TRANSMIT.CL2.QM3 のいずれかです。このどちらであるかは、QM1 のキュー・マネージャー属性 **DEFCLXQ** の値が、SCTQ または CHANNEL のどちらであるかによって決まります。クラスター送信側チャンネル CL2.QM3 が次に開始した時点から、キュー・マネージャーは自動的に XMITQ.CL2.QM3 からメッセージを転送します。

- a) 伝送キュー XMITQ.CL2.QM3 を、クラスター伝送キューから一般の伝送キューに変更します。

これにより、伝送キューとすべてのクラスター送信側チャンネルとの関連付けが解除されます。これに応じて、クラスター送信側チャンネルが次に開始した時点から、IBM MQ は自動的にメッセージを XMITQ.CL2.QM3 からデフォルト・クラスター伝送キューに転送します。それまでは、QM3 上の CL2 へのメッセージは、引き続き XMITQ.CL2.QM3 に配置されます。

```
*... On QM1
ALTER QLOCAL(XMITQ.CL2.QM3) CLCHNAME(' ')
```

- b) クラスター送信側チャンネル CL2.QM3 を停止します。

クラスター送信側チャンネルを停止してから再始動すると、XMITQ.CL2.QM3 からデフォルト・クラスター伝送キューへのメッセージの転送が開始します。通常は、手動でチャンネルを手動で停止して開始することによって、転送を開始します。チャンネルの切断間隔が満了してチャンネルがシャットダウンした後、再始動する場合には、自動的に転送が開始します。

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

コマンドが受け入れられたという応答が返されます。

AMQ8019: Stop IBM MQ channel accepted.

- c) チャンネル CL2.QM3 が停止していることを確認します。

チャンネルが停止しない場合は、FORCE オプションを指定した **STOP CHANNEL** コマンドを再度実行できます。FORCE オプションを設定する一例は、チャンネルが停止しないため、そのチャンネルを同期化するもう一方のキュー・マネージャーを再始動できない場合です。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

応答は、チャンネル状況の要約です。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413))      CURRENT
RQMNAME(QM3)                   STATUS(STOPPED)
SUBSTATE(MQGET)                XMITQ(XMITQ.CL2.QM3)
```

- d) チャンネル CL2.QM3 を開始します。

```
*... On QM1
START CHANNEL(CL2.QM3)
```

コマンドが受け入れられたという応答が返されます。

AMQ8018: Start IBM MQ channel accepted.

- e) チャンネルが開始したことを確認します。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

応答は、チャンネル状況の要約です。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413))      CURRENT
RQMNAME(QM3)                   STATUS(RUNNING)
SUBSTATE(MQGET)                XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE/CL2.QM3)
```

- f) メッセージのゲートウェイ・キュー・マネージャーのエラー・ログをモニターする"AMQ734 チャンネル CL2.QM3 の伝送キューは SYSTEM.CLUSTER.TRANSMIT です。 QUEUE/CL2.QM3 "。

g) クラスター伝送キュー XMITQ.CL2.QM3 を削除します。

```
*... On QM1
DELETE QLOCAL(XMITQ.CL2.QM3)
```

## 次のタスク

クラスター・キューを単独でテストするために、QM2 からキュー別名定義 Q1A を使用して、メッセージを QM3 上の Q1 に送信します。

1. QM2 でサンプル・プログラム **amqspust** を実行して、メッセージを書き込みます。

```
C:\IBM\MQ>amqspust Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

2. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

## 関連概念

241 ページの『クラスター伝送キューとクラスター送信側チャネルの操作』

クラスター・キュー・マネージャー間のメッセージは、クラスター伝送キューに保管されてから、クラスター送信側チャネルによって転送されます。どの時点でも、クラスター送信側チャネルが関連付けられる伝送キューは1つです。チャネルの構成を変更すると、そのチャネルは次回開始したときに、別の伝送キューに切り替わる可能性があります。この切り替え処理は自動化されていて、トランザクションで実行されます。

## 関連タスク

リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。このソリューションは、クラスター化されたキュー・リモート定義と、別個の送信側チャネルおよび伝送キューを使用します。

クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の1つのキュー・マネージャーに分離します。

クラスター伝送キューを区別するようにデフォルトを変更して、メッセージ・トラフィックを分離する  
キュー・マネージャーがクラスター・キューまたはトピックへのメッセージを伝送キューに保管するデフォルトの方法を変更できます。デフォルトを変更することは、ゲートウェイ・キュー・マネージャーでクラスター・メッセージを分離する手段となります。

262 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

## 関連情報

[アクセス制御と複数のクラスター伝送キュー](#)

[クラスター化: 複数のクラスター伝送キューの使用によるアプリケーションの分離](#)

[クラスター化: クラスター伝送キューの構成方法の計画](#)

## クラスター伝送キューを区別するようにデフォルトを変更して、メッセージ・トラフィックを分離する

キュー・マネージャーがクラスター・キューまたはトピックへのメッセージを伝送キューに保管するデフォルトの方法を変更できます。デフォルトを変更することは、ゲートウェイ・キュー・マネージャーでクラスター・メッセージを分離する手段となります。

## 始める前に

1. ゲートウェイ・キュー・マネージャーが IBM WebSphere MQ 7.5 以降にある必要があります。
2. [286 ページの『ゲートウェイ・キュー・マネージャーを使用した 2 つのオーバーラップするクラスターの作成』](#)のハブ・スポーク・アーキテクチャーにデプロイされた IBM MQ クラスターを使用するクライアント/サーバー・アプリケーションに示されているオーバーラップするクラスターを、タスクで説明している手順に従って作成します。

## このタスクについて

複数のクラスター・キューを備えたアーキテクチャーを実装するには、ゲートウェイ・キュー・マネージャーが IBM WebSphere MQ 7.5 以降に配置されている必要があります。複数のクラスター伝送キューを使用するために必要な作業は、ゲートウェイ・キュー・マネージャーでデフォルト・クラスター伝送キュー・タイプを変更することだけです。QM1 で、キュー・マネージャー属性 **DEFCLXQ** の値を、SCTQ から CHANNEL に変更します。304 ページの図 53 を参照してください。この図には、1 つのメッセージ・フローが示されています。他のキュー・マネージャーや他のクラスターへのフローには、キュー・マネージャーが追加の永続動的クラスター伝送キューを作成します。各クラスター送信側チャンネルは、それぞれに異なるクラスター伝送キューからメッセージを転送します。

ゲートウェイ・キュー・マネージャーをクラスターに初めて接続しているのでない限り、変更は即時に適用されません。このタスクには、既存の構成に対する変更を管理する際の標準的な手順が含まれます。キュー・マネージャーが初めてクラスターに参加するときに、個々のクラスター伝送キューを使用するようにセットアップするには、[262 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』](#)を参照してください。

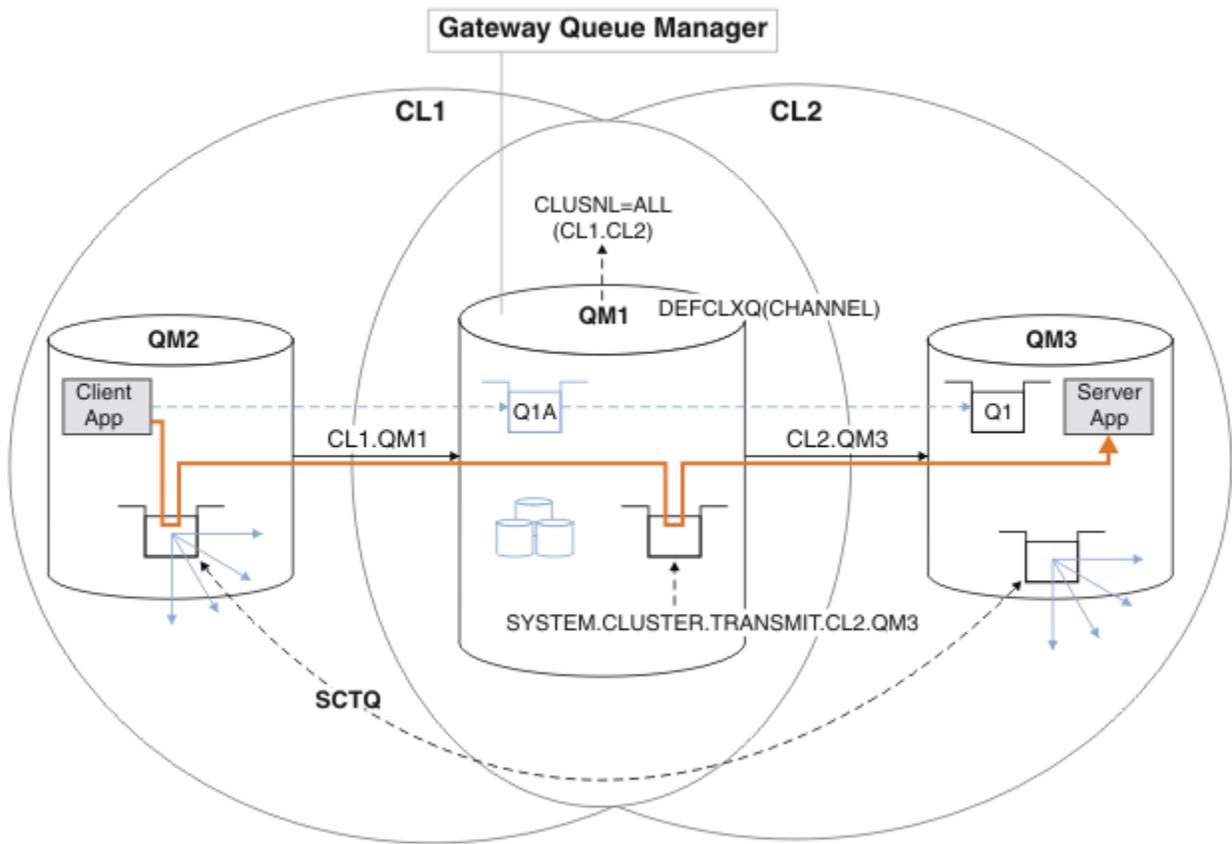


図 53. ゲートウェイ・キュー・マネージャーで個々のクラスター伝送キューを使用するハブ・スポーク・アーキテクチャーにデプロイされたクライアント/サーバー・アプリケーション

## 手順

1. 個々のクラスター伝送キューを使用するようにゲートウェイ・キュー・マネージャーを変更します。

```
*... On QM1
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. 個々のクラスター伝送キューに切り替えます。

実行中でないクラスター送信側チャンネルは、いずれも次に開始するときに、個々の伝送キューを使用するように切り替わります。

実行中のチャンネルを切り替えるには、キュー・マネージャーを再始動するか、または以下のステップに従います。

- a) `SYSTEM.CLUSTER.TRANSMIT.QUEUE` を使用して実行中のクラスター送信側チャンネルをリストします。

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

応答は、チャンネル状況レポートのリストです。

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)          CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412)) CURRENT
RQMNAME(QM2)             STATUS(RUNNING)
SUBSTATE(MQGET)          XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

```

AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)              STATUS(RUNNING)
SUBSTATE(MQGET)           XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1415)) CURRENT
RQMNAME(QM5)              STATUS(RUNNING)
SUBSTATE(MQGET)           XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1414)) CURRENT
RQMNAME(QM4)              STATUS(RUNNING)
SUBSTATE(MQGET)           XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)

```

b) 実行中のチャンネルを停止します。

リストに含まれるチャンネルごとに、以下のコマンドを実行します。

```

*... On QM1
STOP CHANNEL(ChannelName)

```

*ChannelName* は CL1.QM2CL1.QM4、CL1.QM3、CL1.QM5 のそれぞれです。

コマンドが受け入れられたという応答が返されます。

```
AMQ8019: Stop IBM MQ channel accepted.
```

c) 停止されたチャンネルをモニターします。

```

*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')

```

応答は、まだ実行中のチャンネルと停止されたチャンネルのリストです。

```

AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1412)) CURRENT
RQMNAME(QM2)              STATUS(STOPPED)
SUBSTATE( )               XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)              STATUS(STOPPED)
SUBSTATE( )               XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM5)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1415)) CURRENT
RQMNAME(QM5)              STATUS(STOPPED)
SUBSTATE( )               XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM4)           CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1414)) CURRENT
RQMNAME(QM4)              STATUS(STOPPED)
SUBSTATE( )               XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)

```

d) 停止されたチャンネルのそれぞれを開始します。

このステップは、実行中であったすべてのチャンネルに対して行ってください。チャンネルが停止しない場合は、FORCE オプションを指定した **STOP CHANNEL** コマンドを再度実行できます。FORCE オプションを設定する一例は、チャンネルが停止しないため、そのチャンネルを同期化するもう一方のキュー・マネージャーを再始動できない場合です。

```
*... On QM1
START CHANNEL(CL2.QM5)
```

コマンドが受け入れられたという応答が返されます。

AMQ8018: Start IBM MQ channel accepted.

e) 切り替えられる伝送キューをモニターします。

メッセージのゲートウェイ・キュー・マネージャーのエラー・ログをモニターする。" AMQ7341 チャンネル CL2.QM3 の伝送キューは SYSTEM.CLUSTER.TRANSMIT です。 *QUEUE|CL2.QM3* "。

f) SYSTEM.CLUSTER.TRANSMIT.QUEUE が使用されていないことを確認します。

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
DISPLAY QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE) CURDEPTH
```

応答として、以下のような、チャンネル状況レポートのリストおよび SYSTEM.CLUSTER.TRANSMIT.QUEUE の深さが返されます。

AMQ8420: チャンネル状況が見つかりませんでした。

AMQ8409: Display Queue details.

```
QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)      TYPE(QLOCAL)
CURDEPTH(0)
```

g) 開始されたチャンネルをモニターします。

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

応答として、チャンネルのリストが返されます。この場合は、新しいデフォルト・クラスター伝送キューを使用して既に実行されているチャンネルであり、以下のようになります。

AMQ8417: Display Channel Status details.

```
CHANNEL(CL1.QM2)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1412)) CURRENT
RQMNAME(QM2)              STATUS(RUNNING)
SUBSTATE(MQGET)
```

XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM2)

AMQ8417: Display Channel Status details.

```
CHANNEL(CL2.QM3)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)              STATUS(RUNNING)
SUBSTATE(MQGET)
```

XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM3)

AMQ8417: Display Channel Status details.

```
CHANNEL(CL2.QM5)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1415)) CURRENT
RQMNAME(QM5)              STATUS(RUNNING)
SUBSTATE(MQGET)
```

XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)

AMQ8417: Display Channel Status details.

```
CHANNEL (CL1.QM4)          CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1414)) CURRENT
RQMNAME (QM4)             STATUS (RUNNING)
SUBSTATE (MQGET)
XMITQ (SYSTEM.CLUSTER.TRANSMIT.CL1.QM4)
```

## 次のタスク

1. 自動的に定義されたクラスター伝送キューをテストするために、メッセージを QM2 から QM3 上の Q1 に送信し、キュー別名定義 Q1A を使用してキュー名を解決します。

- a. QM2 でサンプル・プログラム **amqspout** を実行して、メッセージを書き込みます。

```
C:\IBM\MQ>amqspout Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

- b. サンプル・プログラム **amqsget** を実行して、QM3 上の Q1 からメッセージを取得します。

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. クラスター・キューに対するメッセージが発生したキュー・マネージャーでクラスター・キューのセキュリティを構成して、セキュリティを再構成するかどうかを検討します。

## 関連タスク

リモート・キュー定義を追加して、ゲートウェイ・キュー・マネージャーから送信されたメッセージを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。このソリューションは、クラスター化されたキュー・リモート定義と、別個の送信側チャンネルおよび伝送キューを使用します。

クラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後このソリューションは、追加のクラスター伝送キューを使用して、メッセージ・トラフィックをクラスター内の1つのキュー・マネージャーに分離します。

クラスターおよびクラスター伝送キューを追加して、ゲートウェイ・キュー・マネージャーから送信されたクラスター・メッセージ・トラフィックを分離する

ゲートウェイ・キュー・マネージャーを使用するオーバーラップするクラスターの構成を変更します。変更後、メッセージは、ゲートウェイ・キュー・マネージャーから、他のクラスター・メッセージと同じ伝送キューやチャンネルを使用せずにアプリケーションに転送されます。からアプリケーションに転送された後、これは、追加のクラスターを使用して、特定のクラスター・キューにメッセージを分離します。

262 ページの『クラスターへのキュー・マネージャーの追加: 別個の伝送キュー』

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、複数のクラスター伝送キューを使用して転送されます。

## 関連情報

アクセス制御と複数のクラスター伝送キュー

## キュー・マネージャーからのクラスター・キューの削除

トロントで INVENTQ キューを無効にします。すべての在庫管理メッセージをニューヨークに送信し、トロントの INVENTQ キューが空になったらそれを削除します。

### 始める前に

注: クラスターへの変更をクラスター全体に伝搬させるには、少なくとも1つの完全リポジトリーが常に使用可能でなければなりません。この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

シナリオ

- 267 ページの『[キューをホストするキュー・マネージャーの追加](#)』での説明に従って、INVENTORY クラスターを設定しました。このクラスターには、4つのキュー・マネージャーが格納されています。LONDON と NEWYORK はどちらも完全リポジトリーを保有しています。PARIS と TORONTO は、部分リポジトリーを保有しています。在庫管理アプリケーションはニューヨークおよびトロントのシステムで動作し、INVENTQ キューのメッセージが到着すると実行されます。
- ワークロードが軽減されたので、在庫管理アプリケーションをトロントで実行する必要はなくなりました。キュー・マネージャー TORONTO によってホストされている INVENTQ キューを無効にし、さらに TORONTO によって NEWYORK の INVENTQ キューにメッセージを送信することが必要です。
- ネットワークは、4つのシステム間すべてに接続されています。
- ネットワーク・プロトコルは、TCP です。

### このタスクについて

以下の手順に従って、クラスター・キューを除去します。

### 手順

#### 1. キューが使用できないことを示す

キューをクラスターから削除するには、クラスター名をローカル・キュー定義から削除します。INVENTQ の TORONTO を他のクラスターからアクセスできないように変更します:

```
ALTER QLOCAL(INVENTQ) CLUSTER('')
```

#### 2. キューが使用できないことを確認する

LONDON または NEWYORK のいずれかの完全リポジトリー・キュー・マネージャーで、次のコマンドを発行して、キューがキュー・マネージャー TORONTO にホストされていないことを確認します。

```
DIS QCLUSTER (INVENTQ)
```

ALTER コマンドが正常に実行されると、TORONTO は結果にリストされません。

#### 3. キューを無効にする

次のように TORONTO で INVENTQ キューを無効にして、このキューにこれ以上メッセージを書き込むことができないようにします。

```
ALTER QLOCAL(INVENTQ) PUT(DISABLED)
```

この結果、MQ00\_BIND\_ON\_OPEN によってこのキューに転送中のメッセージは、送達不能キューに転送されます。すべてのアプリケーションを停止して、このキュー・マネージャーのキューにメッセージが明示的に書き込まれることのないようにする必要があります。

#### 4. キューが空になるまでキューをモニターする

DISPLAY QUEUE コマンドを使用し、IPPROCS、OPPROCS、CURDEPTH の各属性を指定して、キューをモニターします (IBM i では **WRKMQMSTS** コマンドを使用します)。入出力処理の数および現在のキューのサイズがすべてゼロのとき、キューは空です。

#### 5. チャンネルをモニターして、未確定メッセージがないことを確認する

INVENTORY.TORONTO チャンネルに未確定メッセージがないことを確認するには、INVENTORY.TORONTO と呼ばれるクラスター送信側チャンネルを他のキュー・マネージャーのそれぞれについてモニターします。次のように、各キュー・マネージャーから INDOUBT パラメーターを指定して DISPLAY CHSTATUS コマンドを発行します。

```
DISPLAY CHSTATUS(INVENTORY.TORONTO) INDOUBT
```

未確定メッセージがある場合は、必ずそれを解決してから作業を進めてください。例えば、RESOLVE チャンネル・コマンドの発行や、チャンネルの停止および再起動などが考えられます。

#### 6. ローカル・キューを削除する

TORONTO の在庫管理アプリケーションに転送するメッセージがなくなったことを確認したら、次のようにキューを削除します。

```
DELETE QLOCAL(INVENTQ)
```

#### 7. これで、在庫管理アプリケーションをトロントのシステムから削除できます。

アプリケーションを削除することによって、重複が発生しないようになり、システムのスペースが節約されます。

## タスクの結果

この作業によって設定されるクラスターは、前の作業によって設定されたクラスターと類似しています。異なる点は、INVENTQ キューがキュー・マネージャー TORONTO で使用できなくなっていることです。

手順 1 でキューを処理対象から外すと、TORONTO キュー・マネージャーは 2 つの完全リポジトリ・キュー・マネージャーにメッセージを送信します。キュー・マネージャーは、完全リポジトリ・キュー・マネージャーに状況の変化を通知します。完全リポジトリ・キュー・マネージャーは、この情報をクラスター内の他のキュー・マネージャーのうち、INVENTQ に関する情報の更新を要求していたキュー・マネージャーに渡します。

キュー・マネージャーが INVENTQ キューにメッセージを書き込むと、更新された部分リポジトリにより、INVENTQ キューが NEWYORK キュー・マネージャーでのみ使用可能であることが示されます。メッセージは NEWYORK キュー・マネージャーに送信されます。

## 次のタスク

この作業では、削除するキューは 1 つだけであり、そのキューの削除元のクラスターも 1 つだけです。

クラスター名が多数登録されている名前リストを参照しているキューが多数あるとします。例えば、キュー・マネージャー TORONTO は、INVENTQ 以外に PAYROLLQ、SALESQ、PURCHASESQ をホストすることもあります。TORONTO は、これらのキューをそれぞれ該当するすべてのクラスター INVENTORY、PAYROLL、SALES、PURCHASES で使用可能にします。TORONTO キュー・マネージャー上でクラスター名の名前リストを次のように定義します。

```
DEFINE NAMELIST(TOROLIST)  
DESCR('List of clusters TORONTO is in')  
NAMES(INVENTORY, PAYROLL, SALES, PURCHASES)
```

各キュー定義に名前リストを追加します。

```
DEFINE QLOCAL(INVENTQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PAYROLLQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(SALESQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PURCHASESQ) CLUSNL(TOROLIST)
```

ここで、SALES 演算は PURCHASES 演算に引き継がれるため、これらのすべてのキューを SALES クラスターから削除する必要があると仮定します。これを実行するには、TOROLIST 名前リストを変更して、SALES クラスターの名前をこの名前リストから削除するだけで済みます。

名前リストに含まれるいずれかのクラスターから 1 つのキューを削除する場合は、それ以外のクラスター名のリストを登録した名前リストを作成します。次に、この新しい名前リストを使用するようキュー定義を変更します。PAYROLLQ を INVENTORY クラスターから削除するには、以下の手順を実行します。

1. 次のようにして、名前リストを作成します。

```
DEFINE NAMELIST(TOROSHORTLIST)
DESCR('List of clusters TORONTO is in other than INVENTORY')
NAMES(PAYROLL, SALES, PURCHASES)
```

2. 次のようにして、PAYROLLQ キュー定義を変更します。

```
ALTER QLOCAL(PAYROLLQ) CLUSNL(TOROSHORTLIST)
```

## クラスターからのキュー・マネージャーの削除: ベスト・プラクティス

キュー・マネージャーがクラスター内の少なくとも 1 つの完全リポジトリと正常に通信できるシナリオで、クラスターから 1 つのキュー・マネージャーを除去します。

### 始める前に

この方法は、少なくとも 1 つの完全リポジトリが使用可能で、かつ、除去されるキュー・マネージャーから接続できるシナリオでのベスト・プラクティスです。この方法では、手操作による介入を最小限に抑え、キュー・マネージャーがクラスターからの制御された離脱をネゴシエーションすることができます。除去されるキュー・マネージャーが完全リポジトリと接続できない場合、[312 ページの『クラスターからのキュー・マネージャーの削除: 代替方式』](#)を参照してください。

### このタスクについて

以下のタスク例では、キュー・マネージャー LONDON を INVENTORY クラスターから除去します。INVENTORY クラスターは [260 ページの『クラスターにキュー・マネージャーを追加する』](#)の説明に従ってセットアップされ、[308 ページの『キュー・マネージャーからのクラスター・キューの削除』](#)の説明に従って変更されます。

キュー・マネージャーをクラスターから除去する処理は、キュー・マネージャーを追加する処理よりも複雑です。

キュー・マネージャーがクラスターに加わる時、クラスターの既存メンバーには新規キュー・マネージャーについての情報はなく、したがって、そのことについての相互作用はありません。加えられるキュー・マネージャーに送信側チャンネルと受信側チャンネルを新規に作成し、そのキュー・マネージャーが完全リポジトリに接続できるようにする必要があります。

キュー・マネージャーがクラスターから削除される時に、そのキュー・マネージャーに接続しているアプリケーションが、クラスター内のどこかでホストされているキューなどのオブジェクトを使用している可能性があります。また、クラスター内の他のキュー・マネージャーに接続しているアプリケーションが、ターゲット・キュー・マネージャーでホストされているオブジェクトを使用している場合もあります。このようなアプリケーションがある結果、現行のキュー・マネージャーは、クラスターに加わっていたフル・リポジトリ以外のクラスター・メンバーと通信を確立するために、追加の送信側チャンネルを作成する場合があります。そのクラスター内のすべてのキュー・マネージャーには、他のクラスター・メンバーにつ

いて記述するデータのキャッシュ・コピーがあります。これには、除去されるメンバーが含まれる可能性があります。

## 手順

1. キュー・マネージャーをクラスターから除去する前に、クラスターで必要なリソースをキュー・マネージャーがホストしなくなったことを確認します。

- キュー・マネージャーが完全リポジトリをホストする場合、[272 ページの『完全リポジトリの別のキュー・マネージャーへの移動』](#)の1から6の手順を実行します。除去するキュー・マネージャーの完全リポジトリ機能が別のキュー・マネージャーに移動しない場合は、5と6の手順のみを実行します。
- キュー・マネージャーがクラスター・キューをホストする場合、[308 ページの『キュー・マネージャーからのクラスター・キューの削除』](#)の1から7の手順を実行します。
- キュー・マネージャーがクラスター・トピックをホストする場合、トピックを削除するか (例えば、[DELETE TOPIC](#) コマンドを使用する)、または [366 ページの『別のキュー・マネージャーへのクラスター・トピック定義の移動』](#)の説明に従って他のホストに移動します。

注: キュー・マネージャーをクラスターから除去する際にキュー・マネージャーがまだクラスター・トピックをホストしている場合、そのキュー・マネージャーは、トピックが削除されるまで、クラスターに残っているキュー・マネージャーにパブリケーションの送信を引き続き試行する可能性があります。

2. キュー・マネージャー LONDON で、手動で定義したクラスター受信側チャンネルを変更して、クラスターから削除します。

```
ALTER CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) CLUSTER('')
```

3. キュー・マネージャー LONDON で、手動で定義したクラスター送信側チャンネルを変更して、クラスターから削除します。

```
ALTER CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) CLUSTER('')
```

クラスター内の他のキュー・マネージャーは、このキュー・マネージャーとそのクラスター・リソースが、そのクラスターの一部でなくなったことを知ります。

4. クラスター内で完全リポジトリへのフローを待機しているメッセージがなくなるまで、キュー・マネージャー LONDON のクラスター伝送キューをモニターします。

```
DISPLAY CHSTATUS(INVENTORY.PARIS) XQMSGSA
```

メッセージが送信キューに残っている場合は、先へ進む前に、それらのメッセージが完全リポジトリ PARIS と NEWYORK に送られていない理由を特定する必要があります。

## タスクの結果

キュー・マネージャー LONDON は、クラスターの一部ではなくなりました。しかし、依然として独立したキュー・マネージャーとして機能することができます。

## 次のタスク

クラスターの残りのメンバー上で以下のコマンドを発行して、これらの変更の結果を確認できます。

```
DISPLAY CLUSQMGR(LONDON)
```

キュー・マネージャーは、その自動定義クラスター送信側チャンネルが停止するまで表示され続けます。停止するまで待つこともできますし、次のコマンドを発行して、アクティブ・インスタンスがないかどうかモニターすることもできます。

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

このキュー・マネージャーにこれ以上メッセージが送られないことが確実な場合には、クラスターの残りのメンバーに対して次のコマンドを発行して、LONDON へのクラスター送信側チャンネルを停止することもできます。

```
STOP CHANNEL(INVENTORY.LONDON) STATUS(INACTIVE)
```

変更がクラスター全体に伝搬して、このキュー・マネージャーに送達されるメッセージがなくなった後、LONDON の CLUSRCVR チャンネルを停止して削除します。

```
STOP CHANNEL(INVENTORY.LONDON)  
DELETE CHANNEL(INVENTORY.LONDON)
```

除去されたキュー・マネージャーは、[260 ページの『クラスターにキュー・マネージャーを追加する』](#)の説明に従って、後でクラスターに再び追加することができます。除去されたキュー・マネージャーは、クラスターの残りのメンバーの情報を引き続き最大 90 日間キャッシュに入れます。このキャッシュの有効期限が切れるまでは待たないことにした場合、[314 ページの『キュー・マネージャーのクラスター化以前の状態への復元』](#)の説明に従って強制的に除去できます。

## 関連タスク

[クラスターからのキュー・マネージャーの除去 \(IBM MQ Explorer を使用\)](#)

## 関連資料

[ALTER CHANNEL \(チャンネル設定の変更\)](#)

[DISPLAY CHANNEL \(チャンネル定義の表示\)](#)

[DISPLAY CHSTATUS \(チャンネル状況の表示\)](#)

[DISPLAY CLUSQMGR \(クラスター・キュー・マネージャーのチャンネル情報の表示\)](#)

[STOP CHANNEL \(チャンネルの停止\)](#)

## クラスターからのキュー・マネージャーの削除: 代替方式

重大なシステムまたは構成上の問題によりキュー・マネージャーがクラスター内のどの完全リポジトリとも通信できないシナリオで、クラスターから 1 つのキュー・マネージャーを除去します。

## 始める前に

クラスターからキュー・マネージャーを除去するために使用するこの代替方式により、除去されるキュー・マネージャーをクラスターにリンクしているすべてのクラスター・チャンネルを手動で停止して削除し、キュー・マネージャーをクラスターから強制的に除去します。この方法は、除去されるキュー・マネージャーがどの完全リポジトリとも通信できないシナリオで使用されます。このシナリオの原因の例として、キュー・マネージャーが動作を停止していることや、キュー・マネージャーとクラスターの間で発生した通信障害が長引いていることなどが考えられます。そうでなければ、最も一般的な方法 ([310 ページの『クラスターからのキュー・マネージャーの削除: ベスト・プラクティス』](#)) を使用してください。

## このタスクについて

以下のタスク例では、キュー・マネージャー LONDON を INVENTORY クラスターから除去します。INVENTORY クラスターは [260 ページの『クラスターにキュー・マネージャーを追加する』](#)の説明に従ってセットアップされ、[308 ページの『キュー・マネージャーからのクラスター・キューの削除』](#)の説明に従って変更されます。

キュー・マネージャーをクラスターから除去する処理は、キュー・マネージャーを追加する処理よりも複雑です。

キュー・マネージャーがクラスターに加わるとき、クラスターの既存メンバーには新規キュー・マネージャーについての情報はなく、したがって、そのことについての相互作用はありません。加えられるキュー・マネージャーに送信側チャンネルと受信側チャンネルを新規に作成し、そのキュー・マネージャーが完全リポジトリに接続できるようにする必要があります。

キュー・マネージャーがクラスターから削除されるときに、そのキュー・マネージャーに接続しているアプリケーションが、クラスター内のどこかでホストされているキューなどのオブジェクトを使用している可能性があります。また、クラスター内の他のキュー・マネージャーに接続しているアプリケーションが、ターゲット・キュー・マネージャーでホストされているオブジェクトを使用している場合もあります。このようなアプリケーションがある結果、現行のキュー・マネージャーは、クラスターに加わっていたフル・リポジトリ以外のクラスター・メンバーと通信を確立するために、追加の送信側チャンネルを作成する場合があります。そのクラスター内のすべてのキュー・マネージャーには、他のクラスター・メンバーについて記述するデータのキャッシュ・コピーがあります。これには、除去されるメンバーが含まれる可能性があります。

この手順が適していると考えられるのは緊急時です。緊急時にはキュー・マネージャーがクラスターから正規の手順で除外されるのを待つことはできません。

## 手順

1. キュー・マネージャーをクラスターから除去する前に、クラスターで必要なリソースをキュー・マネージャーがホストしなくなったことを確認します。
  - キュー・マネージャーが完全リポジトリをホストする場合、[272 ページの『完全リポジトリの別のキュー・マネージャーへの移動』](#)の 1 から 6 の手順を実行します。除去するキュー・マネージャーの完全リポジトリ機能が別のキュー・マネージャーに移動しない場合は、5 と 6 の手順のみを実行します。
  - キュー・マネージャーがクラスター・キューをホストする場合、[308 ページの『キュー・マネージャーからのクラスター・キューの削除』](#)の 1 から 7 の手順を実行します。
  - キュー・マネージャーがクラスター・トピックをホストする場合、トピックを削除するか (例えば、[DELETE TOPIC](#) コマンドを使用する)、または [366 ページの『別のキュー・マネージャーへのクラスター・トピック定義の移動』](#)の説明に従って他のホストに移動します。

**注:** キュー・マネージャーをクラスターから除去する際にキュー・マネージャーがまだクラスター・トピックをホストしている場合、そのキュー・マネージャーは、トピックが削除されるまで、クラスターに残っているキュー・マネージャーにパブリケーションの送信を引き続き試行する可能性があります。

2. クラスター内の他のキュー・マネージャーと通信するために使用されているすべてのチャンネルを停止します。キュー・マネージャー LONDON 上の CLUSRCVR チャンネルを停止するには、MODE(FORCE) を使用します。そうでない場合は、送信側キュー・マネージャーがチャンネルを停止するまで待たなければならないことがあります。

```
STOP CHANNEL(INVENTORY.LONDON) MODE(FORCE)
STOP CHANNEL(INVENTORY.TORONTO)
STOP CHANNEL(INVENTORY.PARIS)
STOP CHANNEL(INVENTORY.NEWYORK)
```

3. キュー・マネージャー LONDON 上で、チャンネルが停止するまでチャンネルの状態をモニターします。

```
DISPLAY CHSTATUS(INVENTORY.LONDON)
DISPLAY CHSTATUS(INVENTORY.TORONTO)
DISPLAY CHSTATUS(INVENTORY.PARIS)
DISPLAY CHSTATUS(INVENTORY.NEWYORK)
```

チャンネルが停止した後は、クラスター内の他のキュー・マネージャーがそれ以上アプリケーション・メッセージを送受信することはなくなります。

4. キュー・マネージャー LONDON で、手動で定義したクラスター・チャンネルを削除します。

```
DELETE CHANNEL(INVENTORY.NEWYORK)
DELETE CHANNEL(INVENTORY.TORONTO)
```

5. クラスター内の残りのキュー・マネージャーは、削除されるキュー・マネージャーの情報を保持し、引き続きメッセージをそこに送信する場合があります。残りのキュー・マネージャーからこの情報を消去するには、いずれかの完全リポジトリ上のクラスターから、除去されるキュー・マネージャーを以下のようにリセットします。

```
RESET CLUSTER(INVENTORY) ACTION(FORCEREMOVE) QMNAME(LONDON) QUEUES(YES)
```

削除されるキュー・マネージャーと同じ名前の別のキュー・マネージャーがクラスター内に存在する可能性がある場合、削除されるキュー・マネージャーの **QMID** を指定してください。

## タスクの結果

キュー・マネージャー LONDON は、クラスターの一部ではなくなりました。しかし、依然として独立したキュー・マネージャーとして機能することができます。

## 次のタスク

クラスターの残りのメンバー上で以下のコマンドを発行して、これらの変更の結果を確認できます。

```
DISPLAY CLUSQMGR(LONDON)
```

キュー・マネージャーは、その自動定義クラスター送信側チャンネルが停止するまで表示され続けます。停止するまで待つこともできますし、次のコマンドを発行して、アクティブ・インスタンスがないかどうかモニターすることもできます。

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

変更がクラスター全体に伝搬して、このキュー・マネージャーに送達されるメッセージがなくなった後、LONDON の CLUSRCVR チャンネルを削除します。

```
DELETE CHANNEL(INVENTORY.LONDON)
```

除去されたキュー・マネージャーは、260 ページの『[クラスターにキュー・マネージャーを追加する](#)』の説明に従って、後でクラスターに再び追加することができます。除去されたキュー・マネージャーは、クラスターの残りのメンバーの情報を引き続き最大 90 日間キャッシュに入れます。このキャッシュの有効期限が切れるまでは待たないことにした場合、314 ページの『[キュー・マネージャーのクラスター化以前の状態への復元](#)』の説明に従って強制的に除去できます。

### 関連資料

[DELETE CHANNEL](#)  
[DISPLAY CHANNEL](#)  
[DISPLAY CHSTATUS](#)  
[DISPLAY CLUSQMGR](#)  
[STOP CHANNEL](#)  
[RESET CLUSTER](#)

## キュー・マネージャーのクラスター化以前の状態への復元

キュー・マネージャーには、クラスターから除去されても、残りのクラスター・メンバーに関する情報が保持されます。最終的にはこの情報は有効期限切れになり、自動的に削除されます。しかし、この情報をすぐに削除する場合は、このトピックの手順で行うことができます。

## 始める前に

対象となるキュー・マネージャーがクラスターから除去されており、クラスター内で作業を実行しなくなったことを前提としています。例えば、そのキューがクラスターからメッセージを受け取らなくなり、それらのキューでメッセージが届くのを待っているアプリケーションもない状態です。

## このタスクについて

キュー・マネージャーには、クラスターから除去されても、残りのクラスター・メンバーに関する情報が最大 90 日間保持されます。これにはシステム上の利点がある場合があり、特に、キュー・マネージャーがすぐにクラスターに再び加入する場合にはそう言えます。この情報は、最終的に有効期限切れになると、自動的に削除されます。しかし、この情報の手動での削除を選択する理由もあります。以下に例を示します。

- 以前クラスター・リソースを使用していたこのキュー・マネージャー上のすべてのアプリケーションを停止したことを確認することができます。残りのクラスター・メンバーの情報の有効期限が切れるまで、そのようなアプリケーションは伝送キューに書き込みを続けます。クラスターの情報が削除された後、そのようなアプリケーションがクラスター・リソースの使用を試みると、システムでエラー・メッセージが生成されます。
- キュー・マネージャーの状況情報を表示する際に、残りのクラスター・メンバーに関する有効期限情報を表示しない方がよい場合もあります。

このタスクでは、例として INVENTORY クラスターを使用します。LONDON キュー・マネージャーは、[310 ページの『クラスターからのキュー・マネージャーの削除: ベスト・プラクティス』](#)で説明されているように、INVENTORY クラスターから除去されています。クラスターの残りのメンバーの情報を削除するには、LONDON キュー・マネージャーで以下のコマンドを発行します。

## 手順

1. このキュー・マネージャーから、クラスター内の他のキュー・マネージャーのメモリーをすべて除去します。

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

2. すべてのクラスター・リソースが除去されるまで、キュー・マネージャーをモニターします。

```
DISPLAY CLUSQMGR(*) CLUSTER(INVENTORY)  
DISPLAY QCLUSTER(*) CLUSTER(INVENTORY)  
DISPLAY TOPIC(*) CLUSTER(INVENTORY)
```

## 関連情報

[クラスター](#)

[クラスター化と分散キューイングとの比較](#)

[クラスターのコンポーネント](#)

## キュー・マネージャーの保守

保守を実行するために、キュー・マネージャーをクラスターから中断および再開します。

## このタスクについて

クラスターの一部であるキュー・マネージャーの保守は、毎日のように行う必要があります。例えば、キューの中のデータのバックアップや、ソフトウェアに対する修正の適用が必要になります。キュー・マネージャーがいずれかのキューをホストしている場合は、その活動を中断する必要があります。保守が完了したら、キュー・マネージャーの活動を再開できます。

## 手順

1. キュー・マネージャーを中断するには、`SUSPEND QMGR runmqsc` コマンドを次のように発行します。

```
SUSPEND QMGR CLUSTER(SALES)
```

`SUSPEND runmqsc` コマンドは、このキュー・マネージャーが中断されたことを SALES クラスタ内のキュー・マネージャーに通知します。

`SUSPEND QMGR` コマンドの目的は、このキュー・マネージャーへのメッセージの送信をできれば避けるように、他のキュー・マネージャーにアドバイスすることだけです。キュー・マネージャーが使用禁止になるわけではありません。このキュー・マネージャーが処理する必要があるメッセージの中には、引き続きキュー・マネージャーに送信されるものもあります。例えば、このキュー・マネージャーがクラスタ・キューの唯一のホストである場合などです。

キュー・マネージャーが中断している間、ワークロード管理ルーチンは、メッセージをこのキュー・マネージャーに送信することを避けます。このキュー・マネージャーが処理する必要があるメッセージには、ローカル・キュー・マネージャーが送信するメッセージがあります。

IBM MQ は、可能な場合は常にローカル・キュー・マネージャーを選択するのではなく、ワークロード・balancing・アルゴリズムを使用して、どの宛先が適しているかを判別します。

- a) `SUSPEND QMGR` コマンドで `FORCE` オプションを使用して、以下のようにキュー・マネージャーの中断を強制します。

```
SUSPEND QMGR CLUSTER(SALES) MODE(FORCE)
```

`MODE(FORCE)` は、クラスタ内の他のキュー・マネージャーからのインバウンド・チャネルを強制的に停止します。`MODE(FORCE)` を指定しない場合は、デフォルトの `MODE(QUIESCE)` が適用されます。

2. 必要な保守タスクがあれば、すべて実行します。
3. 以下のように `RESUME QMGR runmqsc` コマンドを発行して、キュー・マネージャーを再開します。

```
RESUME QMGR CLUSTER(SALES)
```

## タスクの結果

`RESUME runmqsc` コマンドは、キュー・マネージャーが再び使用可能になったことを完全リポジトリに通知します。完全リポジトリ・キュー・マネージャーは、このキュー・マネージャーに関する情報の更新を要求していた他のキュー・マネージャーに、この通知情報を配布します。

## クラスタ伝送キューの保守

クラスタ伝送キューを使用可能な状態に保つためにあらゆる作業を行います。これらは、クラスタのパフォーマンスには不可欠です。  z/OS では、クラスタ伝送キューの `INDXTYPE` を `CORRELID` に設定します。

## 始める前に

- クラスタ伝送キューが満杯にならないようにしてください。
- `ALTER runmqsc` コマンドを実行して、誤って `GET DISABLED` または `PUT DISABLED` に設定されないように注意してください。
- クラスタ伝送キューを保管するメディア  (例えば、z/OS ページ・セット) が満杯にならないようにしてください。

## このタスクについて

z/OS

以下の手順は、z/OS にのみ適用されます。

### 手順

クラスター伝送キューの INDXTYPE を CORRELID に設定します。

## クラスター・キュー・マネージャーのリフレッシュ

REFRESH CLUSTER コマンドを使用して、ローカル・リポジトリから自動定義チャンネルおよび自動定義クラスター・オブジェクトを除去できます。メッセージは失われません。

### 始める前に

IBM サポートがこのコマンドの使用を要求する場合があります。このコマンドを安易に使用しないでください。例えば大規模クラスターでは、稼働中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、そのクラスターに悪影響が及ぶ可能性があります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。[クラスタリング: REFRESH CLUSTER の使用のベスト・プラクティス](#)を参照してください。

## このタスクについて

キュー・マネージャーはクラスター内でフレッシュ・スタートを行うことができます。通常的环境下、REFRESH CLUSTER コマンドを使用する必要はありません。

### 手順

キュー・マネージャーから REFRESH CLUSTER **MQSC** コマンドを実行して、自動定義クラスター・キュー・マネージャーおよびキュー・オブジェクトをローカル・リポジトリから除去します。

このコマンドにより、他のキュー・マネージャーを参照するオブジェクトのみが削除され、ローカル・キュー・マネージャーに関連したオブジェクトは除去されません。このコマンドにより、自動定義チャンネルも除去されます。クラスター伝送キューにメッセージがないチャンネルと、完全リポジトリ・キュー・マネージャーに接続されていないチャンネルも、このコマンドによって除去されます。

### タスクの結果

事実上、REFRESH CLUSTER コマンドを使用すると、キュー・マネージャーはフル・リポジトリ・コンテンツに従って、コールド・スタートすることができます。IBM MQ は、キューから失われたデータがないことを確認します。

### 関連情報

[クラスタ化: REFRESH CLUSTER の使用に関するベスト・プラクティス](#)

## クラスター・キュー・マネージャーのリカバリー

REFRESH CLUSTER **runmqsc** コマンドを使用して、キュー・マネージャーに関するクラスター情報を最新の状態にします。特定時点のバックアップからキュー・マネージャーをリカバリーした後で、以下の手順に従います。

### 始める前に

特定時点のバックアップから、クラスター・キュー・マネージャーがリストアされています。

## このタスクについて

クラスター内のキュー・マネージャーをリカバリーするには、キュー・マネージャーをリストアし、REFRESH CLUSTER **runmqsc** コマンドを使用して、クラスター情報を最新の状態にします。

**注:** 大規模クラスターでは、処理中のクラスターに **REFRESH CLUSTER** コマンドを使用すると、破壊的な影響を及ぼす恐れがあります。その後、クラスター・オブジェクトが 27 日間隔で対象のキュー・マネージャーすべてに状況の更新を自動的に送信する際にも同様のことが起こり得ます。大規模クラスターでのリフレッシュはクラスターのパフォーマンスと可用性に影響を与える可能性があるを参照してください。

## 手順

キュー・マネージャーが参加しているすべてのクラスターについて、リストアされたキュー・マネージャーで REFRESH CLUSTER コマンドを実行します。

## 次のタスク

その他のキュー・マネージャーで REFRESH CLUSTER コマンドを実行する必要はありません。

## 関連情報

[クラスター化: REFRESH CLUSTER の使用に関するベスト・プラクティス](#)

## 可用性のためのクラスター・チャンネルの構成

適切な構成手順に従って、ネットワークの偶発的な停止が生じた場合に、クラスター・チャンネルの稼働が問題なく続行されるようにします。

## 始める前に

クラスターを使用すると、チャンネルを定義する必要から解放されますが、引き続き保守は必要です。クラスター内のキュー・マネージャー間の通信には、分散キューイングで使用されているものと同じチャンネルの技術が使用されています。クラスター・チャンネルを理解するには、次の事項を理解する必要があります。

- チャンネルの動作方法
- チャンネル状況の検索方法
- チャンネル出口の使用方法

## このタスクについて

特に以下の点について考慮してください。

## 手順

クラスター・チャンネルを構成する際には、以下の点について検討してください。

- クラスター送信側チャンネルおよびクラスター受信側チャンネルの HBINT または KAINTE の値には、多数のハートビートまたはキープアライブのフローによりネットワークに負荷がかからないようなものを選択します。10 秒未満のインターバルを設定した場合に、ネットワークの速度が低下して設定した長さの遅延が発生すると、偽の障害が報告されます。
- 障害を起こしたチャンネル上に未確定のメッセージがあると、それは孤立メッセージになりますが、BATCHHB 値を設定して、このようなメッセージの発生を許可する時間を短縮します。バッチを埋めるのに長い時間がかかる場合、障害を起こしたチャンネル上に未確定バッチが生じる可能性が高まります。チャンネル上のメッセージ・トラフィックが散発的でメッセージのバースト間隔が長い場合、バッチの障害の発生の可能性が高くなります。
- チャンネルのクラスター送信側が障害を起こし、その後、ハートビートまたはキープアライブによってその障害が検出される前に再始動が試行された場合、問題が生じます。チャンネルのクラスター受信側がアクティブのままだと、チャンネル送信側の再始動が拒否されます。この障害を避けるために、クラスター送信側チャンネルが再始動を試行するときにはクラスター受信側チャンネルの終了と再始動が行われるように調整してください。

### **z/OS** IBM MQ for z/OS の場合

ALTER QMGR 上の ADOPTMCA および ADOPTCHK パラメーターを使用して、チャンネルのクラスター受信側がアクティブのままである問題を制御します。

## Multi マルチプラットフォームの場合

qm.ini ファイルまたは Windows NT レジストリーの AdoptNewMCA、AdoptNewMCATimeout、および AdoptNewMCACheck 属性を使用して、アクティブなままのチャンネルのクラスター受信側の問題を制御します。

## クラスターとの間のルーティング・メッセージ

キュー別名、キュー・マネージャー別名、およびリモート・キュー定義を使用して、クラスターを外部のキュー・マネージャーおよび他のクラスターに接続します。

クラスターとの間のメッセージのルーティングについては、以下のサブトピックを参照してください。

### 関連概念

#### [332 ページの『キュー・マネージャー別名とクラスター』](#)

クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

#### [336 ページの『キュー別名とクラスター』](#)

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

#### [335 ページの『応答先キュー別名およびクラスター』](#)

応答先キューの別名定義は、応答情報の代替名を指定するために使用します。応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

### 関連タスク

#### [236 ページの『キュー・マネージャー・クラスターの構成』](#)

クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

#### [248 ページの『新規クラスターのセットアップ』](#)

クラスター例をセットアップするには、これらの手順に従ってください。クラスターを設定する方法は、TCP/IP、LU 6.2、および単一の伝送キューまたは複数の伝送キューを使用する場合で別々に説明されています。あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信することによって、クラスターの動作をテストします。

### 関連情報

#### [クラスター](#)

#### [クラスター化と分散キューイングとの比較](#)

#### [クラスターのコンポーネント](#)

## クラスターへの要求/応答の構成

クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

## 始める前に

320 ページの図 54 は、DEMO と呼ばれるクラスターの外部にある QM3 という名前のキュー・マネージャーを示します。QM3 は、クラスターをサポートしない IBM MQ 製品のキュー・マネージャーの場合があります。QM3 は、Q3 という名前のキューをホストします。このキューは、次のように定義されています。

```
DEFINE QLOCAL(Q3)
```

クラスターの内側には、QM1 および QM2 と呼ばれる 2 つのキュー・マネージャーがあります。QM2 は、Q2 と呼ばれるクラスター・キューをホストしています。このキューは、次のように定義されています。

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO)
```

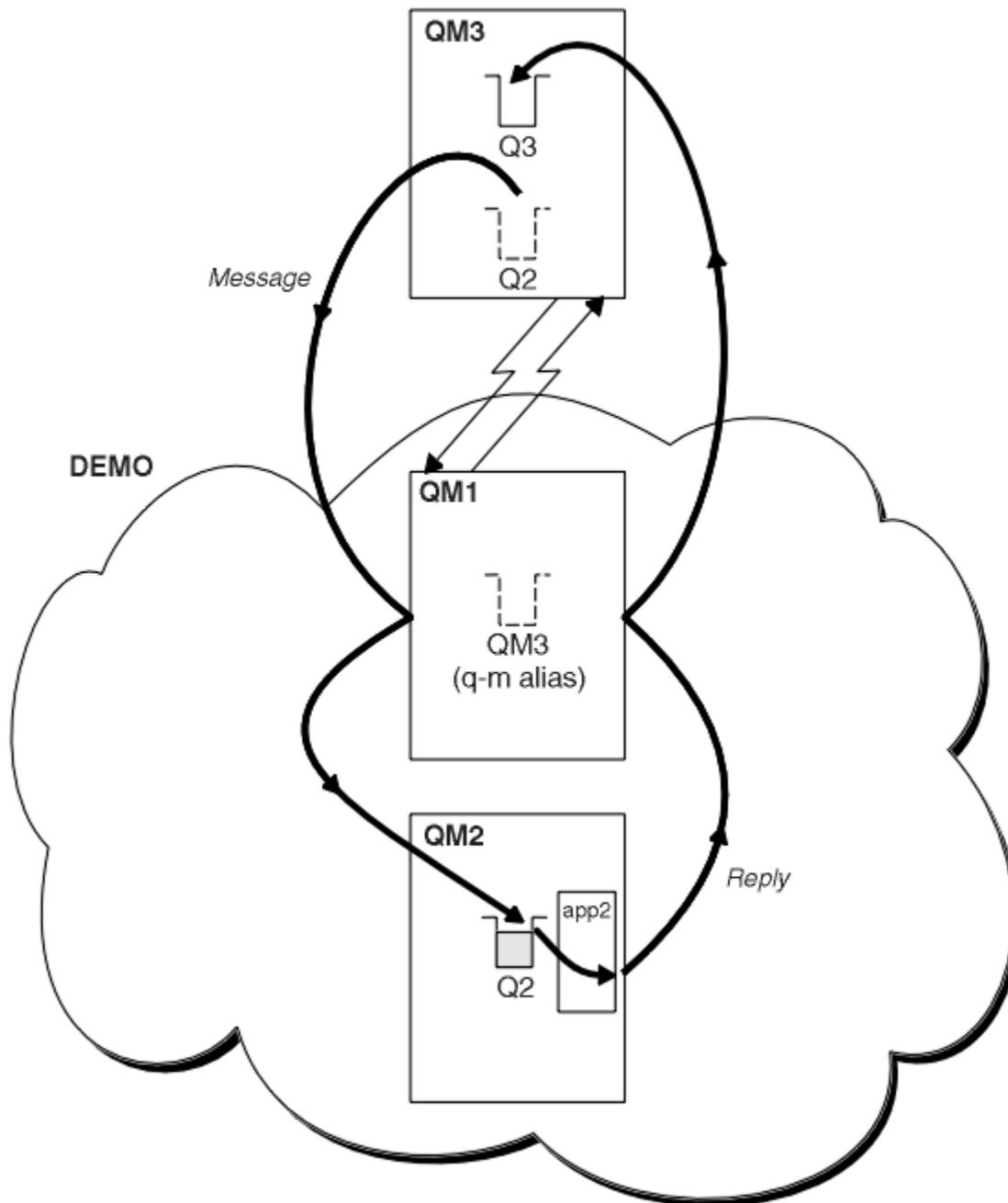


図 54. クラスターの外側にあるキュー・マネージャーからの書き込み

### このタスクについて

以下の手順のアドバイスに従い、要求および応答メッセージのパスをセットアップしてください。

### 手順

1. クラスターに要求メッセージを送信する。

クラスターの外部にあるキュー・マネージャーが、クラスターの内部にある QM2 のキュー Q2 にメッセージを書き込む方法について考えてみます。クラスターの外部にあるキュー・マネージャーには、メッセージを書き込む対象となるクラスターのキューごとに QREMOTE 定義が必要です。

- a) QM3 に Q2 のリモート・キューを定義する。

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(QM2) XMITQ(QM1)
```

QM3 はクラスターの一部ではないので、QM3 は、分散キューイング技術を使用して通信を行う必要があります。したがって、QM3 には、送信側チャンネルおよび QM1 への伝送キューも必要です。QM1 には、これに対応する受信側チャンネルが必要です。このチャンネルおよび伝送キューは、320 ページの図 54 の中では明示的に表示されていません。

この例では、QM3 のアプリケーションが MQPUT 呼び出しを発行してメッセージを Q2 に書き込みます。QREMOTE 定義により、そのメッセージは、QM1 伝送キューからメッセージを取得している送信側チャンネルを使用して、QM2 の Q2 に転送されます。

2. クラスターから応答メッセージを受信する。

キュー・マネージャーの別名を使用して、クラスターの外部にあるキュー・マネージャーへの応答の戻りパスを作成します。ゲートウェイ QM1 は、クラスターの外部にあるキュー・マネージャー QM3 の別名をキュー・マネージャーに通知します。QM3 のキュー・マネージャー別名定義にクラスター属性を追加することにより、クラスターの内側にあるキュー・マネージャーに QM3 を通知します。キュー・マネージャーの別名定義は、RNAME がブランクである以外は、リモート・キュー定義と同じです。

- a) QM1 に QM3 のキュー・マネージャー別名を定義します。

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

QM1 から QM3 に応答を転送するために使用する伝送キューの名前の選択について考慮する必要があります。QREMOTE 定義で XMITQ 属性を省略することにより、伝送キューの名前は暗黙的に QM3 になります。しかし、QM3 は、キュー・マネージャーの別名を使用して、クラスターの残りの部分にアドバタイズすることを想定しているのと同じ名前です。IBM MQ では、伝送キューとキュー・マネージャー別名の両方を同じ名前にすることはできません。解決策の 1 つとして、QM3 へメッセージを転送するための伝送キューをキュー・マネージャー別名と異なる名前で作成します。

- b) 伝送キュー名を QREMOTE 定義で指定する。

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO) XMITQ(QM3.XMIT)
```

新しいキュー・マネージャー別名により、QM3.XMIT という名前の新しい伝送キューと QM3 キュー・マネージャー別名が関連付けられます。これは、簡単で正確な解決策ですが、まったく問題がないわけではありません。これは、伝送キューがターゲット・キュー・マネージャーと同じ名前を持つという、伝送キューの命名規則に反しています。伝送キューの命名規則を維持する別の解決策はあるのでしょうか？

リクエスターは、QM3 から送信される要求メッセージで、応答先キュー・マネージャー名としてデフォルトで QM3 を渡すため、問題が発生します。QM2 のサーバーは、QM3 の応答先キュー・マネージャー名を使用して、応答で QM3 をアドレス指定します。この解決策では、QM1 が、応答メッセージを返信するためのキュー・マネージャー別名として QM3 を通知する必要があるため、QM1 は QM3 を伝送キューの名前として使用できなくなります。

キュー・マネージャー名への応答として QM3 を提供するデフォルトの代わりに、QM3 のアプリケーションは、QM1 に応答メッセージ用の応答先キュー・マネージャー別名を渡す必要があります。ゲートウェイ・キュー・マネージャー QM1 は、QM3 への応答に、QM3 そのものではなくキュー・マネージャー別名を通知して、伝送キューの名前と競合しないようにします。

- c) QM1 に QM3 のキュー・マネージャー別名を定義します。

```
DEFINE QREMOTE(QM3.ALIAS) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

構成コマンドに2つ変更を加える必要があります。

- i) QM1 の QREMOTE で、クラスターの他のキュー・マネージャーに対してキュー・マネージャー別名 QM3.ALIAS を通知し、この別名を実際のキュー・マネージャー QM3 の名前と関連付けます。QM3 は、再び、QM3 への応答キューを返信するための伝送キューの名前になります。
- ii) クライアント・アプリケーションで、要求メッセージを構成する際に、応答先キュー・マネージャーの名前として QM3.ALIAS を指定する必要があります。以下の2つの方法のいずれかで、クライアント・アプリケーションに対して QM3.ALIAS を指定できます。
  - MQMD 内の MQPUT によって構成されている応答先キュー・マネージャー名フィールドで、QM3.ALIAS をコーディングする。応答に動的キューを使用している場合は、この方法で行わなければなりません。
  - 応答先キュー名を指定する場合は、応答先キューではなく、応答先キュー別名の Q3.ALIAS を使用する。

```
DEFINE QREMOTE(Q3.ALIAS) RNAME(Q3) RQMNAME(QM3.ALIAS)
```

## 次のタスク

注：応答先キュー別名の使用を **AMQSREQ0** によって示すことはできません。AMQSREQ0 は、パラメーター 3 に指定されているキュー名か、デフォルトの SYSTEM.SAMPLE.REPLY モデル・キューを使用して、応答先キューをオープンします。MQPUT の応答先キュー・マネージャー別名を指定するためには、応答先キュー別名が含まれている別のパラメーターを指定してサンプルを変更する必要があります。

### 関連概念

#### キュー・マネージャー別名とクラスター

クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

#### 応答先キュー別名およびクラスター

応答先キューの別名定義は、応答情報の代替名を指定するために使用します。応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

#### キュー別名とクラスター

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

### 関連タスク

#### クラスターからの要求/応答の構成

クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

#### クラスター外部からのワークロード・バランスの構成

クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランスが行われます。

#### クラスター間のメッセージ・パスの構成

ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

#### 323 ページの『クラスターのターゲット・キュー・マネージャーの名前の非公表』

キュー・マネージャー名を指定せずに、クラスター内のキュー・マネージャーで定義されているクラスター・キューにメッセージを転送します。

クラスターのターゲット・キュー・マネージャーの名前の非公表  
キュー・マネージャー名を指定せずに、クラスター内のキュー・マネージャーで定義されているクラスター・キューにメッセージを転送します。

## 始める前に

- クラスター内部にあるキュー・マネージャーの名前が、クラスター外部にあるキュー・マネージャーに知られないようにします。
  - クラスター内部のキューをホストするキュー・マネージャーへの参照を解決すると、ワークロード・バランシングを行うための柔軟性がなくなります。
  - また、クラスター内のキューをホストするキュー・マネージャーを変更しにくくなります。
  - 代わりに、RQMNAME をクラスター管理者が指定したキュー・マネージャー別名に置き換えます。
  - 323 ページの『[クラスターのターゲット・キュー・マネージャーの名前の非公表](#)』では、キュー・マネージャー別名を使用して、クラスター外部のキュー・マネージャーをクラスター内部のキュー・マネージャーの管理から分離することについて説明されています。
- しかし、伝送キューの名前の付け方としては、ターゲット・キュー・マネージャーの名前を付けることが勧められています。伝送キューの名前により、クラスター内のキュー・マネージャーの名前が分かれます。従うルールを選択する必要があります。伝送キューの名前にキュー・マネージャー名を使用するか、それともクラスター名を使用するか選択できます。

### ゲートウェイ・キュー・マネージャー名を使用して伝送キューに名前を付ける

クラスター外部のキュー・マネージャーにゲートウェイ・キュー・マネージャー名を開示することは、クラスター・キュー・マネージャー名を公表しないというルールの例外として妥当です。

### クラスターの名前を使用して伝送キューに名前を付ける

ターゲット・キュー・マネージャーの名前を使用するという伝送キューの命名規則に従わない場合は、クラスター名を使用してください。

## このタスクについて

319 ページの『[クラスターへの要求/応答の構成](#)』の作業を変更して、クラスター内部にあるターゲット・キュー・マネージャーの名前を非公開にします。

## 手順

例 (324 ページの [図 55](#) を参照) で、ゲートウェイ・キュー・マネージャー QM1 に DEMO というキュー・マネージャー別名を定義します。

```
DEFINE QREMOTE(DEMO) RNAME(' ') RQMNAME(' ')
```

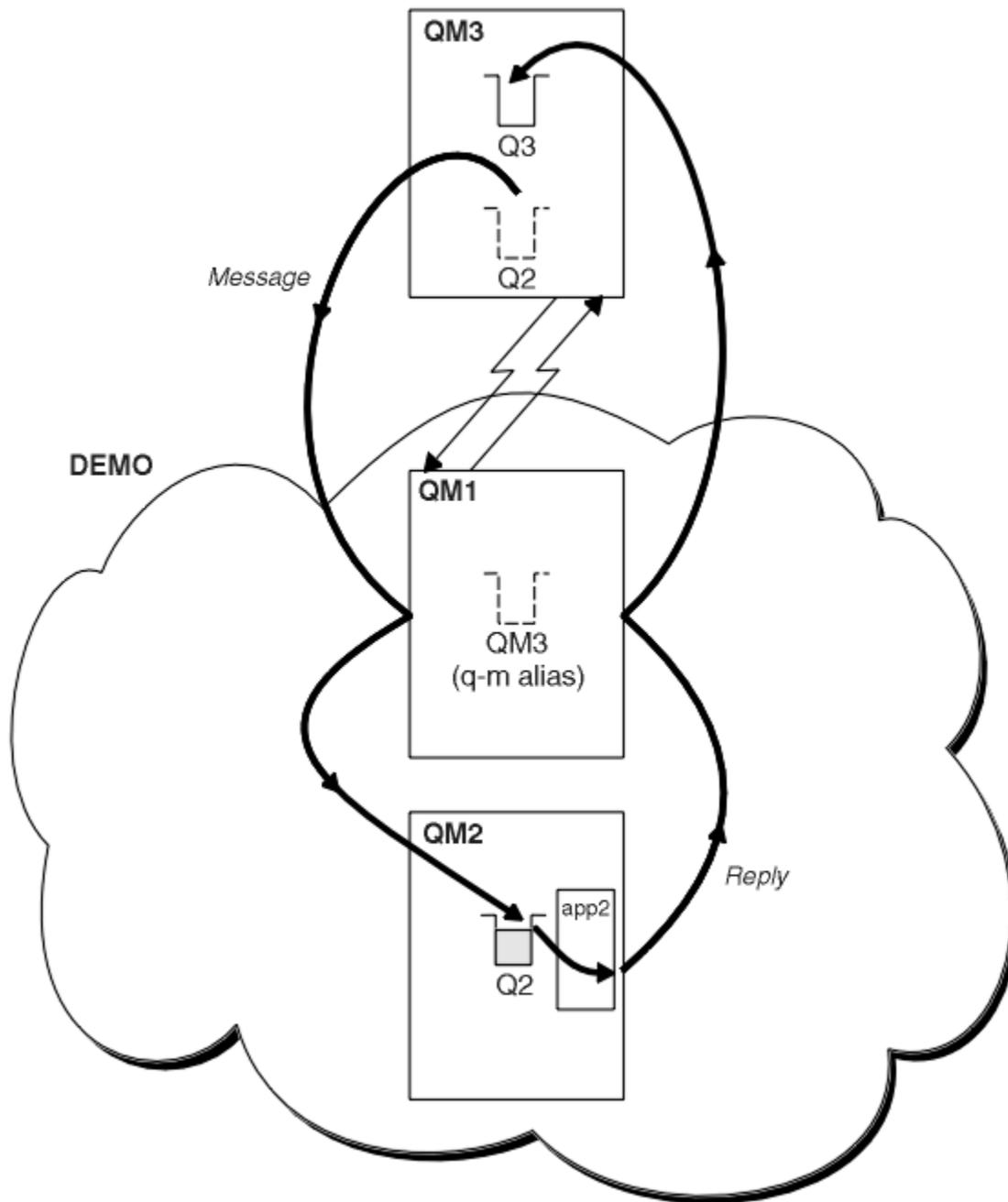


図 55. クラスターの外側にあるキュー・マネージャーからの書き込み

ゲートウェイ・キュー・マネージャーは、QM1 の QREMOTE 定義によって、キュー・マネージャー別名 DEMO を認識します。QM3 (クラスター外部のキュー・マネージャー) は、実際のキュー・マネージャー名を使用せずに、キュー・マネージャー別名 DEMO を使用してメッセージを DEMO のクラスター・キューに送信できます。

クラスターに接続する伝送キューの名前にクラスター名を使用する規則を採用する場合、Q2 のリモート・キュー定義は次のようになります。

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(DEMO) XMIT(DEMO)
```

## タスクの結果

DEMO の Q2 宛のメッセージは、DEMO 伝送キューに入れられます。伝送キューから送信側チャンネルによりゲートウェイ・キュー・マネージャー QM1 に転送されます。ゲートウェイ・キュー・マネージャーは、クラスター・キュー Q2 をホストする、クラスター内のいずれかのキュー・マネージャーにメッセージを転送します。

## クラスターからの要求/応答の構成

クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

## 始める前に

326 ページの図 56 は、クラスター DEMO 内部のキュー・マネージャー QM2 を示します。このキュー・マネージャーは、クラスター外部のキュー・マネージャーにホストされるキュー Q3 に要求を送信します。応答は、クラスター内部にある QM2 の Q2 に返されます。

クラスター外部にあるキュー・マネージャーと通信するために、クラスター内部にある 1 つ以上のキュー・マネージャーがゲートウェイとして機能します。ゲートウェイ・キュー・マネージャーには、クラスター外部にあるキュー・マネージャーへの通信パスがあります。この例では、QM1 がゲートウェイです。

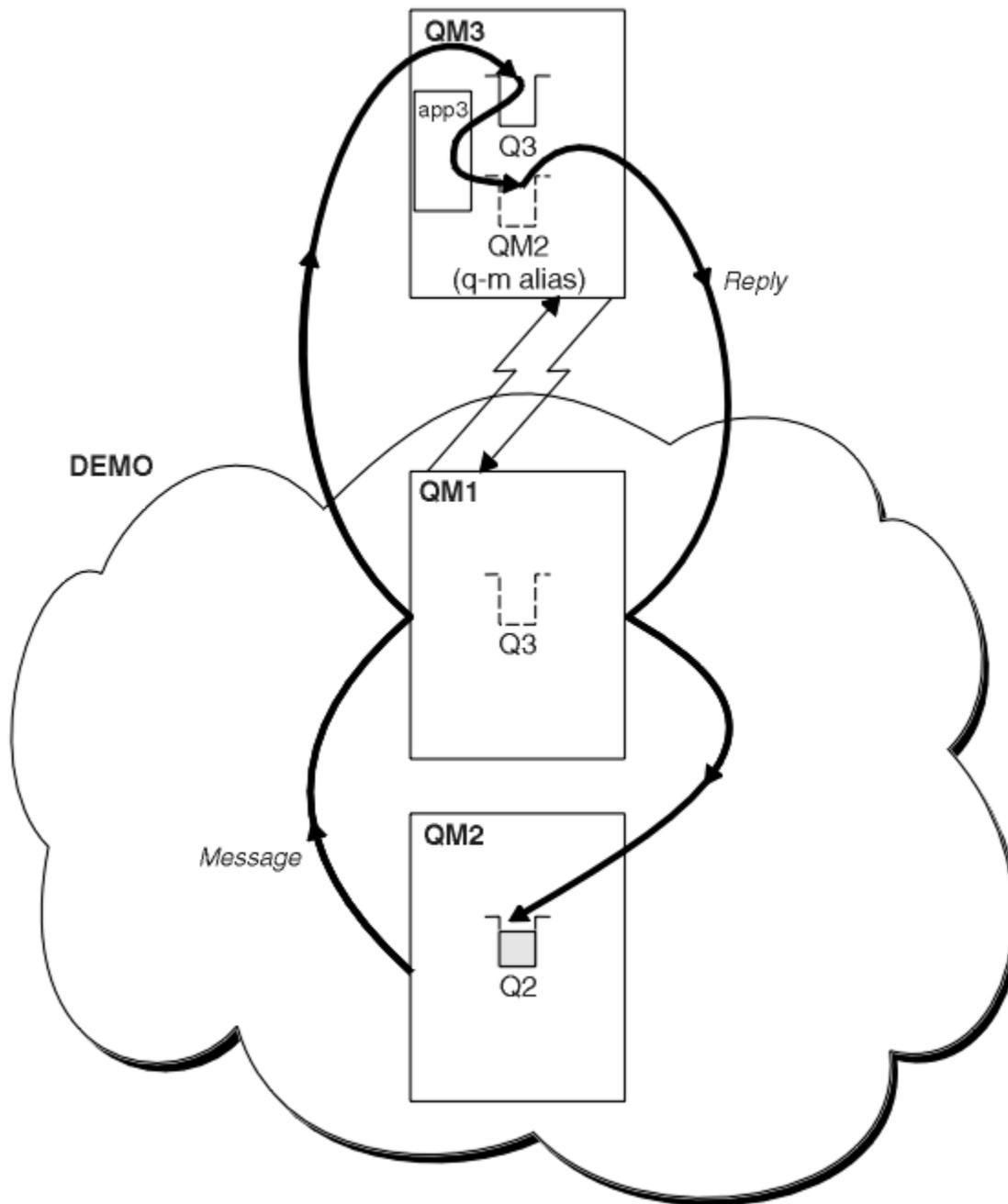


図 56. クラスターの外側にあるキュー・マネージャーへの書き込み

### このタスクについて

以下の指示に従い、要求および応答メッセージのパスをセットアップしてください。

### 手順

1. クラスターから要求メッセージを送信する。

クラスター内部にあるキュー・マネージャー QM2 が、クラスター外部にある QM3 のキュー Q3 にメッセージを書き込む方法について考えてみます。

- a) QM1 で QREMOTE 定義を作成して、リモート・キュー Q3 をクラスターに通知します。

```
DEFINE QREMOTE(Q3) RNAME(Q3) RQMNAME(QM3) CLUSTER(DEMO)
```

このキュー・マネージャーには、クラスター外部にあるキュー・マネージャーへの送信側チャンネルと伝送キューもあります。QM3には、これに対応する受信側チャンネルがあります。このチャンネルは、[326 ページの図 56](#)には示されていません。

QM2 のアプリケーションは、宛先キューと応答の送信先となるキューを指定して、MQPUT 呼び出しを発行します。宛先キューは Q3 で、応答先キューは Q2 です。

メッセージは、QM1 に送信されます。QM1 では、そのリモート・キュー定義を使用して、QM3 の Q3 へのキュー名を解決します。

2. クラスター外部にあるキュー・マネージャーから応答メッセージを受信する。

クラスター外部にあるキュー・マネージャーには、メッセージの送信先となるクラスター内のキュー・マネージャーごとにキュー・マネージャー別名が必要です。キュー・マネージャーの別名には、ゲートウェイ・キュー・マネージャーへの伝送キューの名前を指定する必要があります。この例では、QM3 には QM2 に関するキュー・マネージャーの別名定義が必要です。

- a) QM3 でキュー・マネージャー別名 QM2 を作成します。

```
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) XMITQ(QM1)
```

QM3 には QM1 への送信側チャンネルおよび伝送キューも必要になります。また、QM1 にはこれに対応する受信側チャンネルが必要になります。

QM3 のアプリケーション **app3** は、次に、MQPUT 呼び出しを発行してキュー名 Q2 およびキュー・マネージャー名 QM2 を指定することによって、QM2 に応答できます。

## 次のタスク

クラスターからの経路を複数定義することができます。

### 関連概念

#### キュー・マネージャー別名とクラスター

クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

#### 応答先キュー別名およびクラスター

応答先キューの別名定義は、応答情報の代替名を指定するために使用します。応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

#### キュー別名とクラスター

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

### 関連タスク

#### クラスターへの要求/応答の構成

クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

#### クラスター外部からのワークロード・バランシングの構成

クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

#### クラスター間のメッセージ・パスの構成

ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターと一緒に接続します。ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

## クラスター外部からのワークロード・バランシングの構成

クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

### 始める前に

319 ページの『クラスターへの要求/応答の構成』の 320 ページの図 54 に示されているように、例を構成します。

### このタスクについて

このシナリオでは、クラスター外部にあるキュー・マネージャー (329 ページの図 57 の QM3) は、要求をキュー Q2 に送信します。Q2 は、クラスター DEMO 内の 2 つのキュー・マネージャー QM2 および QM4 によってホストされます。両方のキュー・マネージャーは、ワークロード・バランシングを使用するために、デフォルトのバインド・オプション NOTFIXED で構成されています。クラスター外部にあるキュー・マネージャー QM3 からの要求は、QM1 を介して Q2 のいずれかのインスタンスに送信されます。

QM3 はクラスターの一部ではないので、分散キューイング技術を使用して通信します。このキュー・マネージャーには、送信側チャンネルと QM1 への伝送キューが必要です。QM1 には、これに対応する受信側チャンネルが必要です。このチャンネルおよび伝送キューは、329 ページの図 57 の中では明示的に表示されていません。

以下の手順は、319 ページの『クラスターへの要求/応答の構成』の 320 ページの図 54 を拡張しています。

### 手順

1. QM3 に Q2 の QREMOTE 定義を作成します。

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(Q3) XMITQ(QM1)
```

QM3 がメッセージを書き込むクラスター内のキューごとに QREMOTE 定義を作成します。

2. QM1 でキュー・マネージャー別名 Q3 を作成します。

```
DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ')
```

Q3 は実際のキュー・マネージャー名ではありません。クラスター内の、キュー・マネージャー別名 Q3 とブランク ' ' を同等にするキュー・マネージャー別名定義の名前です。

3. QM2 と QM4 のそれぞれで、Q2 というローカル・キューを定義します。

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

4. ゲートウェイ・キュー・マネージャー QM1 には、特別な定義はありません。

## タスクの結果

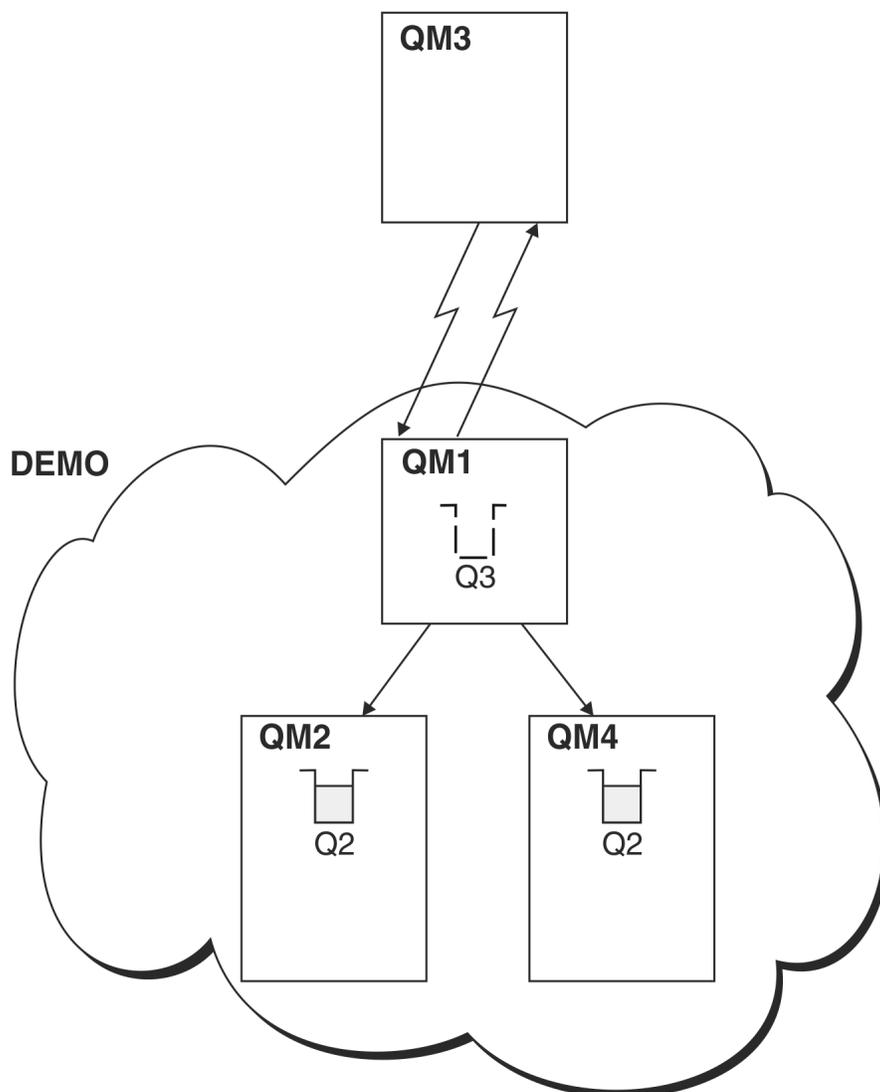


図 57. クラスターの外側にあるキュー・マネージャーからの書き込み

QM3 のアプリケーションが MQPUT 呼び出しを発行してメッセージを Q2 に書き込むとき、QM3 上には QREMOTE 定義があるため、このメッセージは、ゲートウェイ・キュー・マネージャー QM1 を経由してルーティングされます。QM1 はメッセージを受け取ると、このメッセージが Q2 というキューを対象としておりと認識し、ネーム・レゾリューションを実行します。QM1 は、自身のローカル定義を調べますが、Q2 については検出されません。次に、QM1 は自身のクラスター構成を調べ、クラスター DEMO 内の Q2 の 2 つのインスタンスを認識していることを突き止めます。これで QM1 は、ワークロード・บาลancingを使用して QM2 および QM4 に置かれている Q2 のインスタンス間でメッセージを分散できるようになりました。

### 関連概念

#### キュー・マネージャー別名とクラスター

クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランシングを取るには、キュー・マネージャー別名を使用します。

#### 応答先キュー別名およびクラスター

応答先キューの別名定義は、応答情報の代替名を指定するために使用します。応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

#### キュー別名とクラスター

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

## 関連タスク

### クラスターへの要求/応答の構成

クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

### クラスターからの要求/応答の構成

クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

### クラスター間のメッセージ・パスの構成

ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

## 関連情報

### キュー名の解決

### 名前の解決

## クラスター間のメッセージ・パスの構成

ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

## このタスクについて

すべてのキュー・マネージャーを1つの大型クラスターにまとめる代わりに、多数の小型クラスターを使用できます。この場合、各クラスター内の、1つ以上のキュー・マネージャーをブリッジとして機能させます。この長所は、複数のクラスター間でのキュー名およびキュー・マネージャー名の可視性を制限できることです。クラスターのオーバーラップを参照してください。別名を使用してキューの名前とキュー・マネージャーの名前を変更し、名前が競合しないようにしたり、ローカル命名規則に適合させたりすることができます。

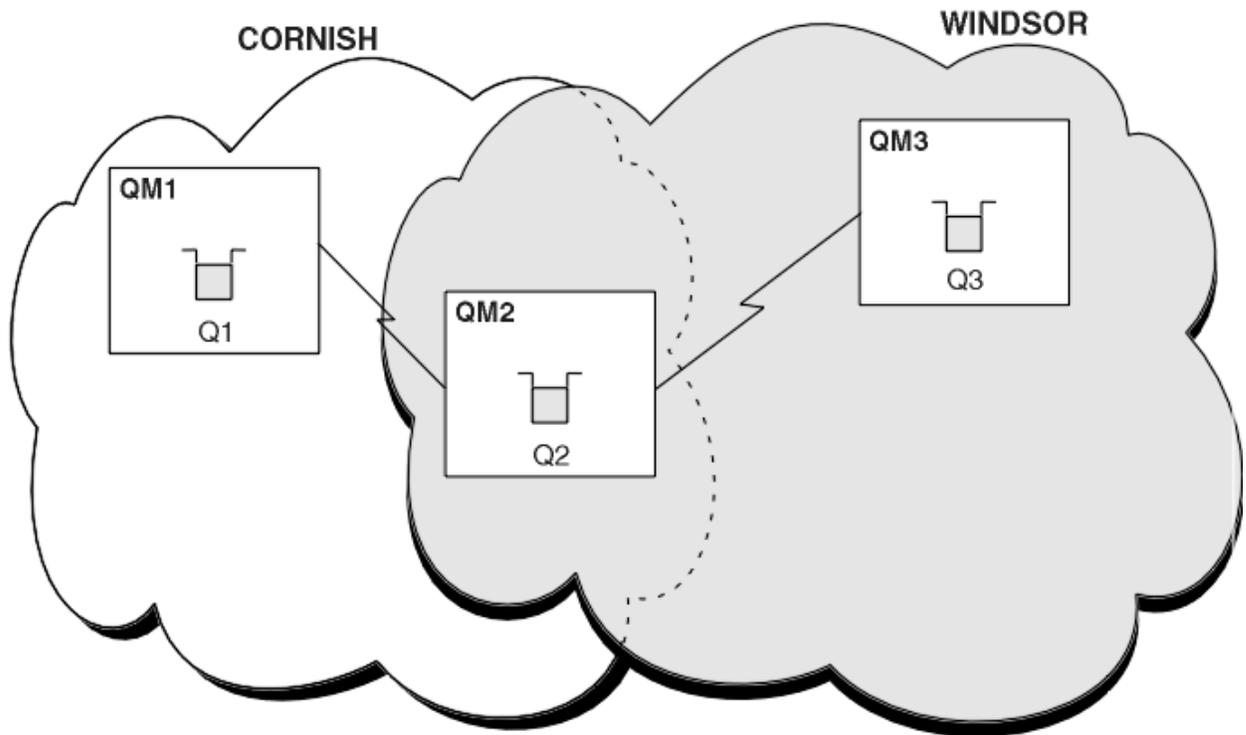


図 58. 複数のクラスターにわたるブリッジ化

331 ページの図 58 には、2 つのクラスターと両者の間をつなぐブリッジを示します。ブリッジは、複数の場合もあります。

次の手順を使用して、クラスターを構成してください。

## 手順

1. QM1 にクラスター・キュー Q1 を定義する。

```
DEFINE QLOCAL(Q1) CLUSTER(CORNISH)
```

2. QM3 にクラスター・キュー Q3 を定義する。

```
DEFINE QLOCAL(Q3) CLUSTER(WINDSOR)
```

3. QM2 で、CORNISHWINDSOR と呼ばれる名前リストを作成する。これには両方のクラスターの名前が格納されます。

```
DEFINE NAMELIST(CORNISHWINDSOR) DESCR('CornishWindsor namelist')
NAMES(CORNISH, WINDSOR)
```

4. QM2 にクラスター・キュー Q2 を定義する

```
DEFINE QLOCAL(Q2) CLUSNL(CORNISHWINDSOR)
```

## 次のタスク

QM2 は、両方のクラスターのメンバーであり、両者の間のブリッジでもあります。ブリッジを介して認識させる必要があるキューごとに、ブリッジの QALIAS 定義が必要になります。331 ページの図 58 の例では、QM2 で次の定義が必要です。

```
DEFINE QALIAS(MYQ3) TARGQ(Q3) CLUSTER(CORNISH) DEFBIND(NOTFIXED)
```

CORNISH 内のキュー・マネージャー (例えば QM1) に接続されたアプリケーションは、キュー別名を使用して、Q3 にメッセージを書き込むことができます。Q3 は MYQ3 として参照されます。メッセージは、QM3 の Q3 に転送されます。

キューをオープンするときに、DEFBIND を NOTFIXED か QDEF のいずれかに設定する必要があります。DEFBIND がデフォルト (OPEN) のままの場合には、キュー・マネージャーは別名定義を、そのホストとなるブリッジ・キュー・マネージャーに解決します。この場合、ブリッジはメッセージを転送しません。

認識させたいキュー・マネージャーごとに、キュー・マネージャーの別名定義が必要です。例えば、QM2 で次の定義が必要です。

```
DEFINE QREMOTE(QM1) RNAME(' ') RQMNAME(QM1) CLUSTER(WINDSOR)
```

MQOPEN 呼び出しに QM1 の名前を明示的に指定することによって、WINDSOR 内のキュー・マネージャー (例えば QM3) に接続されたアプリケーションが QM1 のいずれかのキューにメッセージを書き込むことができます。

### 関連概念

#### キュー・マネージャー別名とクラスター

クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

#### 応答先キュー別名およびクラスター

応答先キューの別名定義は、応答情報の代替名を指定するために使用します。応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

#### キュー別名とクラスター

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

### 関連タスク

#### クラスターへの要求/応答の構成

クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

#### クラスターからの要求/応答の構成

クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

#### クラスター外部からのワークロード・バランスの構成

クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランスが行われます。

### キュー・マネージャー別名とクラスター

クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

キュー・マネージャーの別名は、ブランクの RNAME を持つリモート・キュー定義を使用して作成されますが、これには以下に示す 5 つの使用方法があります。

### メッセージ送信時のキュー・マネージャー名の再マップ

キュー・マネージャーの別名を使用すると、MQOPEN 呼び出しで指定されたキュー・マネージャー名を別のキュー・マネージャーに再マップすることができます。クラスター・キュー・マネージャーも可能です。例えば、キュー・マネージャーには、次のようなキュー・マネージャーの別名定義が存在する場合があります。

```
DEFINE QREMOTE(YORK) RNAME(' ') RQMNAME(CLUSQM)
```

CLUSQM というキュー・マネージャーの別名として YORK を使用できます。これが定義されたキュー・マネージャーのアプリケーションがキュー・マネージャー YORK にメッセージを書き込むと、ローカル・キュー・マネージャーは名前を CLUSQM に解決します。ローカル・キュー・マネージャーが CLUSQM という名前でない場合、メッセージはクラスター伝送キューに書き込まれて CLUSQM に転送されます。伝送ヘッダーも変更され、YORK の代わりに CLUSQM が使用されます。

**注:** 定義は、それを行うキュー・マネージャーにのみ適用されます。クラスター全体に別名を通知するには、リモート・キュー定義に CLUSTER 属性を追加する必要があります。次に、他のキュー・マネージャーからの YORK 宛だったメッセージが、CLUSQM に送信されます。

### メッセージ送信時の伝送キューの変更または指定

別名割り当てを使用して、非クラスター・システムにクラスターを参加させることができます。例えば、クラスター ITALY 内のキュー・マネージャーが、そのクラスター外の PALERMO というキュー・マネージャーと通信することができます。通信するには、クラスター内のキュー・マネージャーの 1 つがゲートウェイの機能を果たす必要があります。ゲートウェイ・キュー・マネージャーから、次のコマンドを発行します。

```
DEFINE QREMOTE(ROME) RNAME(' ') RQMNAME(PALERMO) XMITQ(X) CLUSTER(ITALY)
```

このコマンドは、キュー・マネージャー別名定義です。ROME をキュー・マネージャーと定義して通知します。クラスター ITALY 内のキュー・マネージャーからのメッセージは、このキュー・マネージャーを介して、PALERMO の宛先にマルチホップして到達できます。キュー・マネージャー名を ROME に設定して開かれたキューに書き込まれたメッセージは、キュー・マネージャー別名が定義されたゲートウェイ・キュー・マネージャーに送信されます。到達したメッセージは伝送キュー X に書き込まれ、非クラスター・チャンネルによってキュー・マネージャー PALERMO に転送されます。

この例で ROME という名前を選択したことは、重要ではありません。QREMOTE と RQMNAME の値は、同じ場合があります。

### メッセージ受信時の宛先指定

キュー・マネージャーはメッセージを受信すると、宛先キューとキュー・マネージャーの名前を伝送ヘッダーから抽出します。伝送ヘッダー内のキュー・マネージャーと同じ名前のキュー・マネージャー別名定義を探します。見つかると、伝送ヘッダー内のキュー・マネージャー名の代わりに、キュー・マネージャー別名定義での RQMNAME を使用します。

キュー・マネージャーの別名をこのように使用する理由は、次の 2 つです。

- メッセージを別のキュー・マネージャーに送信するため
- キュー・マネージャー名を変更して、ローカル・キュー・マネージャーと同じ名前にするため

### ゲートウェイ・キュー・マネージャーでキュー・マネージャー別名を使用して、それぞれ異なるクラスター内にあるキュー・マネージャーの間でメッセージをルーティングする

アプリケーションは、キュー・マネージャー別名を使用して、別のクラスター内のキューにメッセージを送信できます。このキューは、クラスター・キューである必要はありません。キューは、1 つのクラスター内に定義されます。アプリケーションは、別のクラスター内にあるキュー・マネージャーに接続されます。ゲートウェイ・キュー・マネージャーが、この 2 つのクラスターを接続します。キュー

一がクラスター・キューとして定義されていない場合、正しいルーティングが行われるためには、アプリケーションがキュー名とクラスター化されたキュー・マネージャー別名を使用してキューをオープンする必要があります。構成例については、286 ページの『ゲートウェイ・キュー・マネージャーを使用した2つのオーバーラップするクラスターの作成』を参照してください。図1に示されている応答メッセージ・フローは、その構成例から引用しています。

以下の図に、一時動的キュー RQ に返される応答メッセージが辿るパスを示します。サーバー・アプリケーションは、QM3 に接続され、キュー・マネージャー名 QM2 を使用して、応答キューをオープンします。キュー・マネージャー名 QM2 は、QM1 上でクラスター・キュー・マネージャー別名として定義されます。QM3 は、応答メッセージを QM1 にルーティングします。QM1 は、そのメッセージを QM2 にルーティングします。

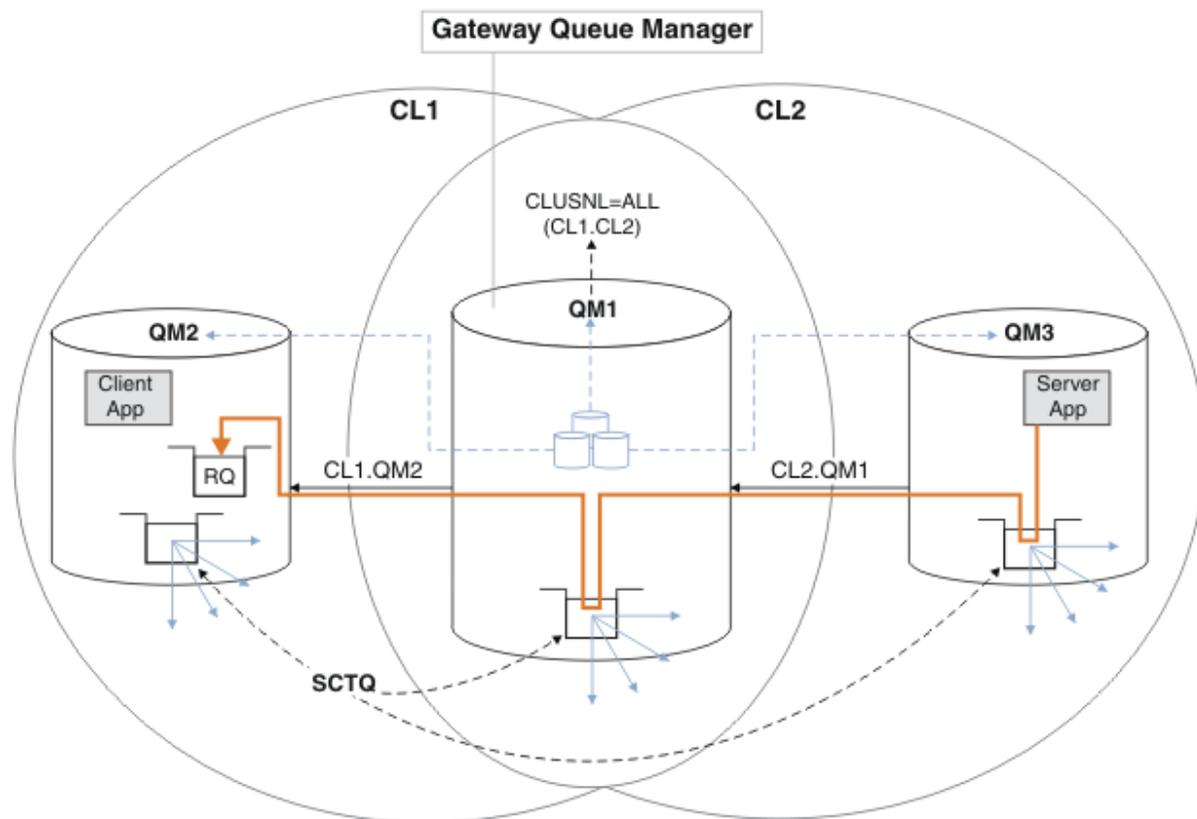


図 59. 応答メッセージを異なるクラスターに返すためのキュー・マネージャー別名の使用

このルーティングは、次のようにして行われます。QM1 には、各クラスター内のすべてのキュー・マネージャーのキュー・マネージャー別名定義があります。これらの別名は、すべてのクラスターでクラスター化されます。それぞれの別名からキュー・マネージャーへと向かうグレーの破線矢印は、各キュー・マネージャー別名が、クラスターのうちの少なくとも1つにある実際のキュー・マネージャーに解決されることを示します。この例では、QM2 別名が CL1 と CL2 の両方のクラスターでクラスター化されています。この別名は、CL1 内にある実際のキュー・マネージャー QM2 に解決されます。サーバー・アプリケーションは、キュー名 RQ への応答、およびキュー・マネージャー名 QM2 への応答を使用して応答メッセージを作成します。このメッセージは、QM1 にルーティングされます。これは、キュー・マネージャー別名定義 QM2 はクラスター CL2 内の QM1 に定義されており、キュー・マネージャー QM2 はクラスター CL2 のメンバーではないためです。メッセージをターゲット・キュー・マネージャーに送信できないことから、この別名定義を持つキュー・マネージャーにメッセージが送信されません。

QM1 は、QM2 に転送するために、メッセージを QM1 上のクラスター伝送キューに配置します。QM1 での QM2 のキュー・マネージャー別名定義は、QM2 を実際のターゲット・キューとして定義しているため、QM1 はメッセージを QM2 にルーティングします。別名定義が参照できるのは、実際の定義だけで、自身を参照することはできないため、この定義は循環しません。QM1 と QM2 は両方とも同じクラスタ

ー CL1 内にあることから、実際の定義は QM1 によって解決されます。QM1 は、CL1 のリポジトリから QM2 の接続情報を見つけ出し、メッセージを QM2 にルーティングします。メッセージを QM1 によって転送するためには、サーバー・アプリケーションが、オプション DEFBIND を MQBND\_BIND\_NOT\_FIXED に設定して応答キューをオープンしていなければなりません。サーバー・アプリケーションがオプション MQBND\_BIND\_ON\_OPEN を設定して応答キューをオープンしている場合、メッセージは転送されず、送達不能キューに入れられます。

### キュー・マネージャーをクラスターへのゲートウェイとして使用して、クラスター外から来るメッセージのワークロード・バランシングを行う

クラスター内の複数のキュー・マネージャーに EDINBURGH というキューを定義します。クラスター外からそのキューに来るメッセージのワークロードのバランスを取るためのクラスター化メカニズムが必要です。

クラスターの外側にあるキュー・マネージャーは、クラスター内の 1 つのキュー・マネージャーへの伝送キューと送信側チャンネルを必要とします。このキューをゲートウェイ・キュー・マネージャーといいます。デフォルトのワークロード・バランシング・メカニズムを利用するためには、以下のいずれかのルールが該当しなければなりません。

- ゲートウェイ・キュー・マネージャーに、EDINBURGH キューのインスタンスが含まれていないこと。
- ゲートウェイ・キュー・マネージャーが ALTER QMGR で CLWLUSEQ(ANY) を指定していること。

クラスター外に起因するワークロード・バランシングの例については、[328 ページの『クラスター外部からのワークロード・バランシングの構成』](#)を参照してください。

### 関連概念

#### 応答先キュー別名およびクラスター

応答先キューの別名定義は、応答情報の代替名を指定するために使用します。応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

#### キュー別名とクラスター

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

### 関連タスク

#### クラスターへの要求/応答の構成

クラスター外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

#### クラスターからの要求/応答の構成

クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

#### クラスター外部からのワークロード・バランシングの構成

クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

#### クラスター間のメッセージ・パスの構成

ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターと一緒に接続します。ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

### 応答先キュー別名およびクラスター

応答先キューの別名定義は、応答情報の代替名を指定するために使用します。応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

以下に例を示します。

- キュー・マネージャー VENICE にあるアプリケーションは、MQPUT 呼び出しを使用して、メッセージをキュー・マネージャー PISA に送信します。アプリケーションは、以下の応答先キュー情報をメッセージ記述子に提供します。

```
ReplyToQ=' QUEUE '
ReplyToQMgr=' '
```

- QUEUE に送信された応答を PISA の OTHERQ で受信できるようにするには、VENICE で応答先キューの別名として使用するリモート・キューの定義を作成します。この別名が有効なのは、別名が作成されたシステム上だけです。

```
DEFINE QREMOTE(QUEUE) RNAME(OTHERQ) RQMNAME(PISA)
```

RQMNAME 自体がクラスター・キュー・マネージャーである場合でも、RQMNAME と QREMOTE に同じ名前を指定することができます。

## 関連概念

### キュー・マネージャー別名とクラスター

クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

### キュー別名とクラスター

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

## 関連タスク

### クラスターへの要求/応答の構成

クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

### クラスターからの要求/応答の構成

クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

### クラスター外部からのワークロード・バランシングの構成

クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

### クラスター間のメッセージ・パスの構成

ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

## キュー別名とクラスター

キュー別名を使用して、クラスター・キューの名前を非表示にしたり、キューをクラスター化したり、さまざまな属性やさまざまなアクセス制御を使用したりすることができます。

QALIAS 定義は、キューを認識するための別名を作成するために使用されます。別名を作成する理由には、以下のようにさまざまなものがあります。

- 別のキューを使用してアプリケーションを始動するが、そのアプリケーションは変更しない場合。
- メッセージの書き込み先であるキューの実名をアプリケーションに認識させない場合。
- キューが定義された場所での命名規則とは異なる命名規則を適用している場合。
- 使用しているアプリケーションがキューに実名でアクセスすることは許可されず、別名によってのみアクセスが許可される場合。

キュー・マネージャーの QALIAS 定義は、DEFINE QALIAS コマンドを使用して作成します。例えば、以下のコマンドを実行します。

```
DEFINE QALIAS(PUBLIC) TARGQ(LOCAL) CLUSTER(C)
```

このコマンドは、クラスター C 内のキュー・マネージャーに PUBLIC というキューを通知します。PUBLIC は、LOCAL というキューに解決される別名です。PUBLIC に送信されたメッセージは、LOCAL と呼ばれるキューに送信されます。

キューの別名定義を使用して、キュー名をクラスター・キューに解決することもできます。例えば、以下のコマンドを実行します。

```
DEFINE QALIAS(PRIVATE) TARGQ(PUBLIC)
```

このコマンドによって、キュー・マネージャーは PRIVATE という名前を使用して、PUBLIC という名前でクラスター内の他の場所に通知されたキューにアクセスすることができます。この定義には CLUSTER 属性が設定されていないので、この定義はこの定義を作成したキュー・マネージャーだけに適用されます。

## 関連概念

### キュー・マネージャー別名とクラスター

クラスターとの間でメッセージを送信するときにキュー・マネージャーの名前を隠蔽するには、およびクラスターに送信されるメッセージのワークロード・バランスを取るには、キュー・マネージャー別名を使用します。

### 応答先キュー別名およびクラスター

応答先キューの別名定義は、応答情報の代替名を指定するために使用します。応答先キューの別名定義は、分散キューイング環境の場合とまったく同様に、クラスターと一緒に使用できます。

## 関連タスク

### クラスターへの要求/応答の構成

クラスターの外部にあるキュー・マネージャーからの要求/応答メッセージ・パスを構成します。クラスターとの間の通信パスとしてゲートウェイ・キュー・マネージャーを使用して、クラスター内部の詳細を公表しないようにします。

### クラスターからの要求/応答の構成

クラスターからクラスター外部にあるキュー・マネージャーへの要求/応答メッセージ・パスを構成します。ゲートウェイ・キュー・マネージャーを使用して、クラスター内部のキュー・マネージャーがクラスター外部と通信する方法の詳細を非公開にします。

### クラスター外部からのワークロード・バランシングの構成

クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・バランシングが行われます。

### クラスター間のメッセージ・パスの構成

ゲートウェイ・キュー・マネージャーを使用して、複数のクラスターを一緒に接続します。ゲートウェイ・キュー・マネージャーでクラスター・キューまたはクラスター・キュー・マネージャーの別名を定義して、すべてのクラスターでキューまたはキュー・マネージャーを認識できるようにします。

## クラスターによるワークロードの管理

クラスター内の異なるキュー・マネージャーにキューの複数インスタンスを定義することによって、キュー・サービス作業を複数のサーバーに分散させることができます。障害が発生した場合にメッセージが別のキュー・マネージャーに再キューイングされるのを妨げる要因はいくつかあります。

システム管理を軽減するためにクラスターをセットアップすることに加えて、複数のキュー・マネージャーが同じキューの 1 つのインスタンスをホストするクラスターを作成することができます。

クラスター内の複数のキュー・マネージャーが互いにクローンになるように、クラスターを編成できます。各キュー・マネージャーは同じアプリケーションを実行でき、同じキューのローカル定義を持つことができます。  例えば、z/OS 並列シスプレックスでは、クローン・アプリケーションが共有 Db2 ま

たは仮想記憶アクセス方式 (VSAM) データベースのデータにアクセスできます。アプリケーションのインスタンスをいくつか備えることによって、キュー・マネージャーの間でワークロードを分散させることができます。アプリケーションの各インスタンスはメッセージを受信し、それぞれ独立して実行されます。

このようなクラスター編成には、次のような利点があります。

- キューおよびアプリケーションの可用性の向上。
- メッセージのスループットの高速化。
- ネットワーク内でのワークロード分布の均等化。

特定のキューのインスタンスをホストするどのキュー・マネージャーも、そのキューを宛先とするメッセージを処理できるので、アプリケーションはメッセージ送信時にキュー・マネージャーを指定しません。1つのクラスターに同じキューのインスタンスが複数含まれる場合、IBM MQ はメッセージを送付するキュー・マネージャーを選択します。最適な宛先は、キュー・マネージャーとキューの可用性とともに、キュー・マネージャー、キュー、およびチャネルに関連付けられたいくつかのクラスター・ワークロード固有の属性に基づいて選択されます。[クラスターでのワークロード・บาลancing](#)を参照してください。

**z/OS** IBM MQ for z/OS では、キュー共有グループ内のキュー・マネージャーは共有キューとしてクラスター・キューをホストすることができます。共有クラスター・キューは、同一キュー共有グループ内のすべてのキュー・マネージャーが使用可能です。例えば、[同じキューに複数のインスタンスがある場合のクラスター](#)では、キュー・マネージャー QM2 と QM4 のどちらかまたは両方を共有キュー・マネージャーにすることができます。それぞれにキュー Q3 の定義があります。QM4 と同じキュー共有グループ内のどのキュー・マネージャーも、共有キュー Q3 に書き込まれたメッセージを読み取ることができます。キュー共有グループそれぞれに最大 32 のキュー・マネージャーを含めることができ、各キュー・マネージャーが同じデータにアクセスできます。キュー共有は、メッセージのスループットを著しく高めます。

ワークロード管理のためのクラスター構成について詳しくは、以下のサブトピックを参照してください。

## 関連概念

[319 ページの『クラスターとの間のルーティング・メッセージ』](#)

キュー別名、キュー・マネージャー別名、およびリモート・キュー定義を使用して、クラスターを外部のキュー・マネージャーおよび他のクラスターに接続します。

## 関連タスク

[236 ページの『キュー・マネージャー・クラスターの構成』](#)

クラスターによって提供されるキュー・マネージャーを相互接続するためのメカニズムにより、初期構成と継続的な管理を簡単に行えます。クラスター・コンポーネントを定義し、クラスターを作成および管理できます。

[248 ページの『新規クラスターのセットアップ』](#)

クラスター例をセットアップするには、これらの手順に従ってください。クラスターを設定する方法は、TCP/IP、LU 6.2、および単一の伝送キューまたは複数の伝送キューを使用する場合で別々に説明されています。あるキュー・マネージャーから別のキュー・マネージャーにメッセージを送信することによって、クラスターの動作をテストします。

[328 ページの『クラスター外部からのワークロード・บาลancingの構成』](#)

クラスター外部にあるキュー・マネージャーからクラスター・キューのコピーへのメッセージ・パスを構成します。その結果、クラスター外部からクラスター・キューの各インスタンスへの要求のワークロード・บาลancingが行われます。

## 関連情報

[クラスター化と分散キューイングとの比較](#)

[分散キューイングとクラスター](#)

[クラスターのコンポーネント](#)

[クラスター・チャネル](#)

[クラスター・キューが MQPUT に対して無効になったときの処置](#)

[クラスター送信側チャネルで設定されたワークロード・บาลancingが機能しない](#)

[クラスター・キュー・モニターのサンプル・プログラム \(AMQSCLM\)](#)

[クラスター・ワークロード出口の作成とコンパイル](#)

## 同じキューに複数のインスタンスがある場合のクラスターの例

この例では、1つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。1つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか1つのキュー・マネージャーに送信することを選択したりできます。

339 ページの図 60 は、キュー Q3 の定義が複数あるクラスターを示しています。QM1 のアプリケーションが Q3 へメッセージを書き込む場合、Q3 のどのインスタンスがメッセージを処理するかは必ずしも分かりません。Q3 のローカル・インスタンスがある QM2 または QM4 でアプリケーションを実行している場合は、デフォルトでは Q3 のローカル・インスタンスがオープンします。CLWLUSEQ キュー属性を設定すると、キューのローカル・インスタンスをキューのリモート・インスタンスと同様に処理できます。

MQOPEN のオプション DefBind は、ターゲットのキュー・マネージャーが選択されるのが、MQOPEN 呼び出しを発行する際か、それとも伝送キューからメッセージを転送する際かを制御します。

DefBind を MQBND\_BIND\_NOT\_FIXED に設定すると、メッセージの伝送時に使用可能になるキューのインスタンスにメッセージを送信できます。これにより、次のような問題が避けられます。

- メッセージがターゲット・キュー・マネージャーに着信したときに、ターゲット・キューが使用不可になっている。
- キューの状態が変更されている。
- クラスター・キュー別名を使用してメッセージが書き込まれているのに、クラスター・キュー別名のインスタンスが定義されているキュー・マネージャーにターゲット・キューのインスタンスが存在しない。

伝送時にこれらの問題のいずれかが見つかった場合、ターゲット・キューの別の使用可能なインスタンスが検索され、メッセージが転送されます。使用可能なキューのインスタンスがない場合は、送達不能キューにメッセージが入れられます。

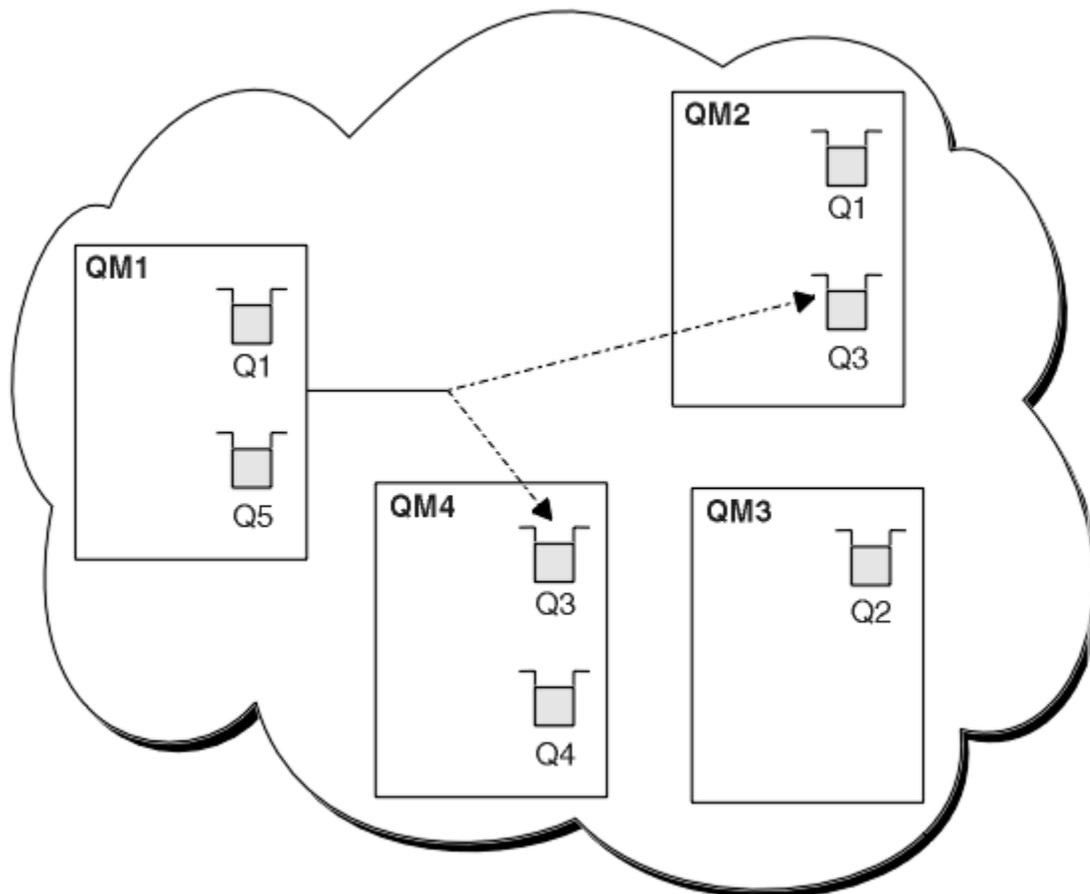


図 60. 同じキューの複数のインスタンスがあるクラスター

メッセージを転送できなくなる可能性がある1つの要因として、MQBND\_BIND\_ON\_OPENを使用してメッセージが固定のキュー・マネージャーまたはチャンネルに割り当てられている場合があります。MQOPEN時にバインドされるメッセージは、別のチャンネルに再割り振りされることはありません。メッセージの再割り振りには、クラスター・チャンネルの障害処理が実際に行われているときにのみ行われることに注意してください。チャンネルが既に失敗している場合は、再割り振りは行われません。

宛先キュー・マネージャーがサービス休止になると、システムはメッセージを転送しようとします。この場合、メッセージが失われたり重複メッセージが作られたりしてメッセージの完全性が損なわれることはありません。キュー・マネージャーの故障によりメッセージの完全性に疑問がある場合、そのメッセージの転送は行われません。

**z/OS** IBM MQ for z/OS では、メッセージ再割り振りプロセスが完了するまで、チャンネルは完全に停止しません。モードを FORCE または TERMINATE に設定した状態でチャンネルを停止すると、このプロセスが中断するため、その場合は一部の BIND\_NOT\_FIXED メッセージがすでに別のチャンネルに再割り振りされているか、メッセージの順番が狂っている可能性があります。

**注:** **z/OS**

1. 同じキューに複数のインスタンスがあるクラスターを設定する前に、メッセージ相互に依存関係がないことを確認してください。例えば、特定の順序で処理する必要や同じキュー・マネージャーで処理する必要がないようにします。
2. 同じキューの複数のインスタンスの定義は同じにしてください。同じにしない場合、MQINQ 呼び出しが違くと、得られる結果にも違いが生じます。

## 関連概念

アプリケーション・プログラミングおよびクラスター

同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

## 関連タスク

キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

クラスター内での2つのネットワークの使用

以下の指示に従って、2つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

クラスター内での1次ネットワークと2次ネットワークの使用

以下の説明に従って、あるネットワークを1次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。バックアップ・ネットワークは、1次ネットワークに問題が発生した場合に使用します。

バックアップとして機能するキューの追加

以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

使用されるチャンネルの数の制限

以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャンネルの数を制限します。

キューをホストするより強力なキュー・マネージャーの追加

以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの2倍のメッセージ数を処理できるようにします。

## キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

## 始める前に

注: クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

### シナリオ

- クラスターへの新規キュー・マネージャーの追加での説明に従って、INVENTORY クラスターをセットアップしました。このクラスターには、3 つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK は完全リポジトリを保有し、PARIS は部分リポジトリを保有しています。在庫管理アプリケーションは、ニューヨークのシステムで動作します。このシステムは、キュー・マネージャー NEWYORK に接続されています。このアプリケーションは、INVENTQ キューのメッセージが到着すると実行されます。
- パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

## このタスクについて

以下の手順に従って、ローカルでキューをホストするキュー・マネージャーを追加します。

### 手順

1. PARIS キュー・マネージャーを変更します。

パリのアプリケーションがパリとニューヨークの INVENTQ を使用できるようにするには、キュー・マネージャーに通知する必要があります。PARIS で、以下のコマンドを発行します。

```
ALTER QMGR CLWLUSEQ(ANY)
```

2. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。

これを調べる前に、在庫管理アプリケーションがメッセージの処理順序に依存しないことを確認してください。詳細については、[メッセージの類縁性の処理](#)を参照してください。

3. 在庫管理アプリケーションをパリのシステムにインストールする
4. クラスター・キュー INVENTQ を定義する。

INVENTQ キューは、キュー・マネージャー NEWYORK に既にホストされていますが、PARIS によってもホストされます。キュー・マネージャー PARIS に次のように定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

これですべての定義が完了しました。IBM MQ for z/OS でまだチャンネル・イニシエーターを始動していない場合は始動してください。すべてのプラットフォームで、キュー・マネージャー PARIS 上でリスナー・プログラムを開始してください。リスナーは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャンネルを起動します。

## タスクの結果

[342 ページの図 61](#) は、このタスクによってセットアップされるクラスターを示します。

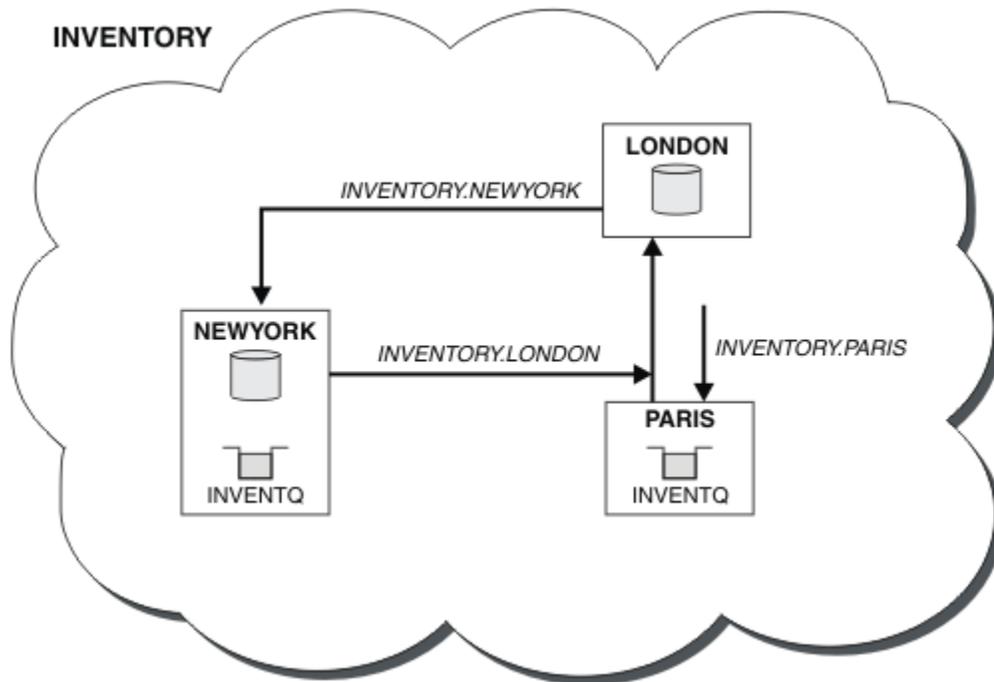


図 61. 3つのキュー・マネージャーで構成される INVENTORY クラスタ

このクラスタの変更では、キュー・マネージャー NEWYORK および LONDON のいずれも変更されませんでした。これらのキュー・マネージャーの完全リポジトリは、PARIS の INVENTQ にメッセージを送信できるようにするために必要な情報によって、自動的に更新されます。

## 次のタスク

現在、INVENTQ キューと在庫管理アプリケーションは、このクラスタ内の2つのキュー・マネージャーによってホストされています。これにより、これらの可用性が増大してメッセージのスループットが向上し、ワークロードが2つのキュー・マネージャーに分散されるようになります。キュー・マネージャー LONDON、NEWYORK、PARIS のいずれかによって INVENTQ に書き込まれたメッセージは、PARIS または NEWYORK に交互に経路指定されるため、ワークロードのバランスを取ることができます。

## 関連概念

### 同じキューに複数のインスタンスがある場合のクラスタの例

この例では、1つのキューに複数のインスタンスがあるクラスタにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。1つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか1つのキュー・マネージャーに送信することを選択したりできます。

### アプリケーション・プログラミングおよびクラスタ

同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

## 関連タスク

### クラスタ内での2つのネットワークの使用

以下の指示に従って、2つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

### クラスタ内での1次ネットワークと2次ネットワークの使用

以下の説明に従って、あるネットワークを1次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。バックアップ・ネットワークは、1次ネットワークに問題が発生した場合に使用します。

### バックアップとして機能するキューの追加

以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

### 使用されるチャンネルの数の制限

以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャンネルの数を制限します。

### キューをホストするより強力なキュー・マネージャーの追加

以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの2倍のメッセージ数を処理できるようにします。

## クラスター内での2つのネットワークの使用

以下の指示に従って、2つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

## 始める前に

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも1つの完全リポジトリーが常に使用可能でなければなりません。この作業を開始する前に、リポジトリーが使用可能であることを確認してください。

### シナリオ

- 『クラスターへの新規キュー・マネージャーの追加』の説明に従って、INVENTORY クラスターがセットアップされています。このクラスターには、3つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK は完全リポジトリーを保有し、PARIS は部分リポジトリーを保有しています。在庫管理アプリケーションは、ニューヨークのシステムで動作します。このシステムは、キュー・マネージャー NEWYORK に接続されています。このアプリケーションは、INVENTQ キューのメッセージが到着すると実行されます。
- 新しい店舗が、2つの異なるネットワークが存在する TOKYO に追加されます。どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

## このタスクについて

以下の手順に従って、クラスター内の2つのネットワークを使用します。

## 手順

1. TOKYO が最初に参照する完全リポジトリーを決める。

クラスター内の各キュー・マネージャーは、クラスターに関する情報を収集するために、いずれかの完全リポジトリーを参照する必要があります。キュー・マネージャーは、独自の部分リポジトリーを作成します。どのリポジトリーが選択されるかは、あまり重要ではありません。この例では、NEWYORK が選択されます。新しいキュー・マネージャーをクラスターに追加すると、このキュー・マネージャーは両方のリポジトリーと通信します。

2. CLUSRCVR チャンネルを定義する。

クラスター内の各キュー・マネージャーに、メッセージを受信できるクラスター受信側を定義する必要があります。このキュー・マネージャーは、各ネットワークで通信する必要があります。

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETB.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network B for TOKYO')
```

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETA.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network A for TOKYO')
```

3. キュー・マネージャー TOKYO に CLUSSDR チャンネルを定義する

クラスター内の各キュー・マネージャーは、その最初の完全リポジトリへのメッセージの送信に使用できるクラスター送信側チャンネルを1つ定義する必要があります。この場合、NEWYORKを選択したので、TOKYOには次の定義が必要になります。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from TOKYO to repository at NEWYORK')
```

これですべての定義が完了しました。IBM MQ for z/OS でまだチャンネル・イニシエーターを開始していない場合は開始します。すべてのプラットフォームで、キュー・マネージャー PARIS 上でリスナー・プログラムを開始してください。リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャンネルを起動します。

## タスクの結果

344 ページの図 62 は、このタスクによってセットアップされるクラスターを示します。

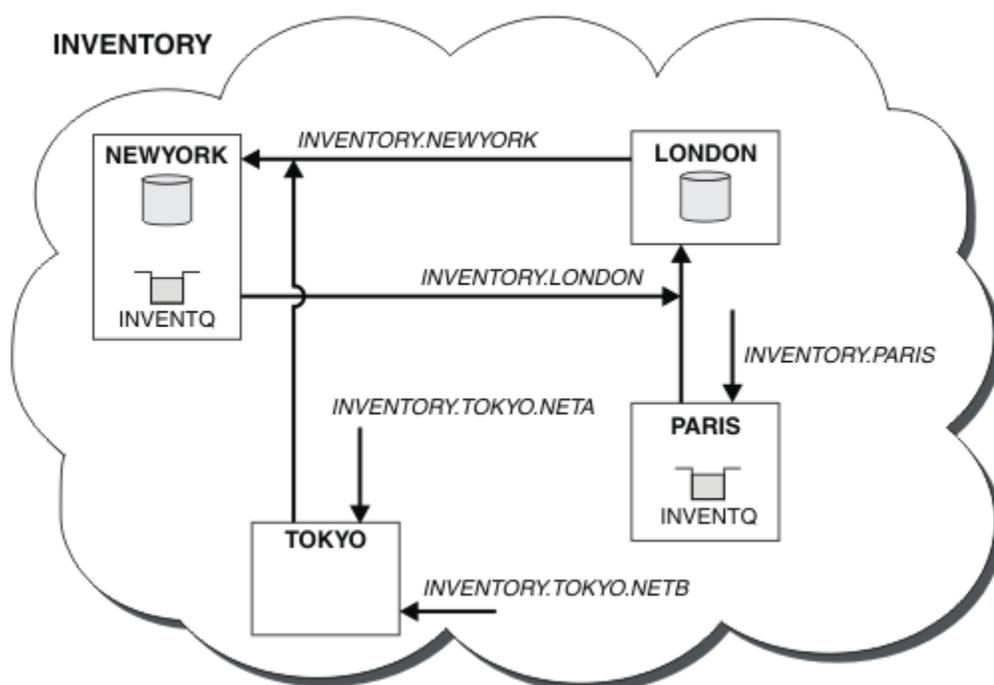


図 62. 4つのキュー・マネージャーで構成される INVENTORY クラスター

3つの定義のみを作成して、クラスターにキュー・マネージャー TOKYO を追加し、2つの異なるネットワーク経路を使用できるようにしました。

### 関連概念

#### 同じキューに複数のインスタンスがある場合のクラスターの例

この例では、1つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。1つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか1つのキュー・マネージャーに送信することを選択したりできます。

#### アプリケーション・プログラミングおよびクラスター

同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

### 関連タスク

キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

#### クラスター内での 1 次ネットワークと 2 次ネットワークの使用

以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

#### バックアップとして機能するキューの追加

以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

#### 使用されるチャンネルの数の制限

以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャンネルの数を制限します。

#### キューをホストするより強力なキュー・マネージャーの追加

以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

#### 260 ページの『クラスターにキュー・マネージャーを追加する』

以下の指示に従って、作成したクラスターにキュー・マネージャーを追加します。クラスター・キューおよびトピックへのメッセージは、単一のクラスター伝送キュー SYSTEM.CLUSTER.TRANSMIT.QUEUE を使用して転送されます。

### **クラスター内での 1 次ネットワークと 2 次ネットワークの使用**

以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

### **始める前に**

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

シナリオ

- 343 ページの『クラスター内での 2 つのネットワークの使用』での説明に従って、INVENTORY クラスターを設定しました。このクラスターには、4 つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK はどちらも完全リポジトリを保有し、PARIS と TOKYO は部分リポジトリを保有しています。在庫管理アプリケーションは、ニューヨークのシステムで実行され、キュー・マネージャー NEWYORK に接続されます。TOKYO キュー・マネージャーには、通信可能な 2 つの異なるネットワークがあります。
- ネットワークの 1 つを 1 次ネットワークにし、もう 1 つのネットワークをバックアップ・ネットワークにするつもりです。さらに、1 次ネットワークに問題が発生した場合にバックアップ・ネットワークを使用することを計画しています。

### **このタスクについて**

NETPRTY 属性を使用して、クラスター内の 1 次ネットワークと 2 次ネットワークを構成します。

### **手順**

TOKYO で既存の CLUSRCVR チャンネルを変更する。

ネットワーク A チャンネルが 1 次チャンネルであり、ネットワーク B チャンネルが 2 次チャンネルであることを示すために、次のコマンドを使用します。

```
a) ALTER CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) NETPRTY(2) DESCR('Main cluster-receiver channel for TOKYO')
```

```
b) ALTER CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) NETPRTY(1)
   DESC('Backup cluster-receiver channel for TOKYO')
```

## 次のタスク

異なるネットワーク優先度を使用するようにチャンネルを構成することにより、クラスターに対して、1次ネットワークと2次ネットワークが存在することを定義しました。これらのチャンネルを使用するクラスター内のキュー・マネージャーは、1次ネットワークが使用可能には必ずこれを自動的に使用します。キュー・マネージャーは、1次ネットワークが使用不可の場合は2次ネットワークを使用するためにフェイルオーバーします。

### 関連概念

#### 同じキューに複数のインスタンスがある場合のクラスターの例

この例では、1つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。1つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか1つのキュー・マネージャーに送信することを選択したりできます。

#### アプリケーション・プログラミングおよびクラスター

同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

### 関連タスク

#### キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQのインスタンスを追加します。

#### クラスター内での2つのネットワークの使用

以下の指示に従って、2つの異なるネットワークが存在するTOKYOに新しい店舗を追加します。どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

#### バックアップとして機能するキューの追加

以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

#### 使用されるチャンネルの数の制限

以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャンネルの数を制限します。

#### キューをホストするより強力なキュー・マネージャーの追加

以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの2倍のメッセージ数を処理できるようにします。

## バックアップとして機能するキューの追加

以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

## 始める前に

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも1つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

### シナリオ

- 260ページの『クラスターにキュー・マネージャーを追加する』での説明に従って、INVENTORYクラスターを設定しました。このクラスターには、3つのキュー・マネージャーが格納されています。このうち、LONDONとNEWYORKは完全リポジトリを保有し、PARISは部分リポジトリを保有しています。在庫管理アプリケーションは、ニューヨークのシステムで動作します。このシステムは、キュー・マネージャーNEWYORKに接続されています。このアプリケーションは、INVENTQキューのメッセージが到着すると実行されます。

- 現在ニューヨークで稼働している在庫管理システムのバックアップを提供するため、シカゴで新しい店舗の準備が進められています。シカゴのシステムは、ニューヨークのシステムに問題が発生した場合のみ使用されます。

## このタスクについて

以下の手順に従って、バックアップとして機能するキューを追加します。

### 手順

#### 1. CHICAGO が最初に参照する完全リポジトリを決める

クラスター内の各キュー・マネージャーは、クラスターに関する情報を収集するために、いずれかの完全リポジトリを参照する必要があります。キュー・マネージャーは、独自の部分リポジトリを作成します。特定のキュー・マネージャーにどのリポジトリが選択されるかは、あまり重要ではありません。この例では、NEWYORK が選択されます。新しいキュー・マネージャーをクラスターに追加すると、このキュー・マネージャーは両方のリポジトリと通信します。

#### 2. CLUSRCVR チャンネルを定義する。

クラスター内の各キュー・マネージャーに、メッセージを受信できるクラスター受信側を定義する必要があります。CHICAGO には、以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.CHICAGO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(CHICAGO.CMSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel for CHICAGO')
```

#### 3. キュー・マネージャー CHICAGO に CLUSSDR チャンネルを定義する。

クラスター内の各キュー・マネージャーは、その最初の完全リポジトリへのメッセージの送信に使用できるクラスター送信側チャンネルを 1 つ定義する必要があります。この場合、NEWYORK を選択したので、CHICAGO には次の定義が必要になります。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from CHICAGO to repository at NEWYORK')
```

#### 4. 既存のクラスター・キュー INVENTQ を変更する。

キュー・マネージャー NEWYORK がホストしている INVENTQ は、キューのメイン・インスタンスです。

```
ALTER QLOCAL(INVENTQ) CLWLPRTY(2)
```

#### 5. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。

これを調べる前に、在庫管理アプリケーションがメッセージの処理順序に依存しないことを確認してください。

#### 6. 在庫管理アプリケーションを CHICAGO にインストールする。

#### 7. バックアップ・クラスター・キュー INVENTQ を定義する

INVENTQ は、キュー・マネージャー NEWYORK にすでにホストされていますが、CHICAGO によってもバックアップとしてホストされます。キュー・マネージャー CHICAGO に次のように INVENTQ を定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) CLWLPRTY(1)
```

これですべての定義が完了しました。IBM MQ for z/OS でまだチャンネル・イニシエーターを開始していない場合は開始します。すべてのプラットフォームで、キュー・マネージャー CHICAGO でリスナー・プログラムを始動します。リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャンネルを起動します。

## タスクの結果

348 ページの図 63 は、このタスクによってセットアップされるクラスターを示します。

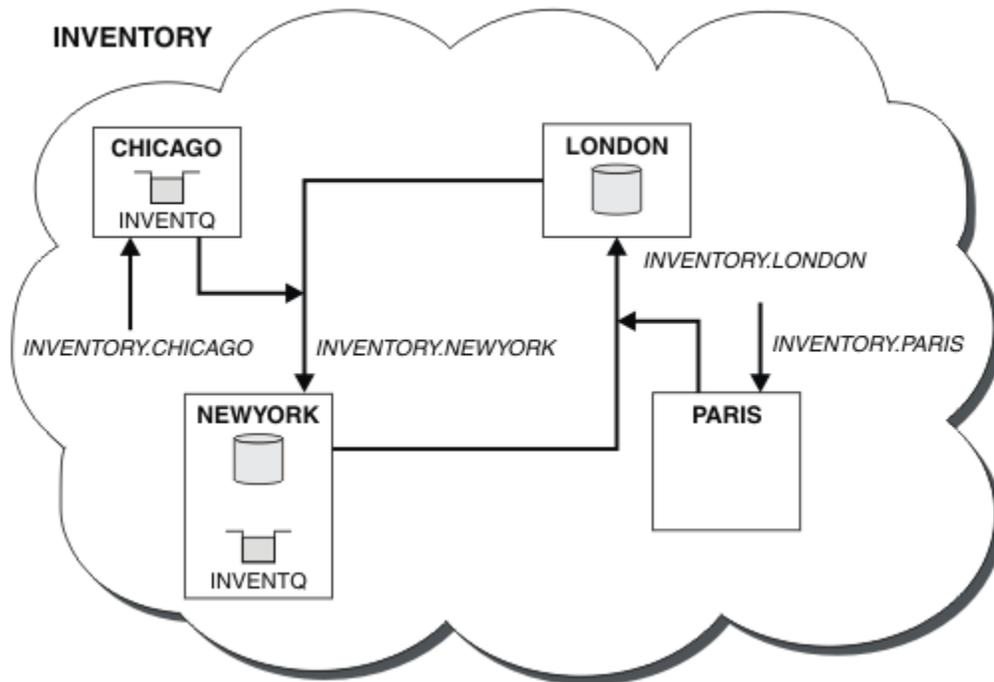


図 63. 4 つのキュー・マネージャーで構成される INVENTORY クラスタ

現在、INVENTQ キューと在庫管理アプリケーションは、このクラスタ内の 2 つのキュー・マネージャーによってホストされています。CHICAGO キュー・マネージャーはバックアップです。INVENTQ に書き込まれたメッセージは、CHICAGO の代わりに、送信されたときに使用不可でない限り NEWYORK に経路指定されます。

**注：**

リモート・キュー・マネージャーの可用性は、そのキュー・マネージャーに対するチャンネルの状況に基づきます。チャンネルを開始する際に、チャンネルの状態は何度か変わります。その中には、クラスタのワークロード管理アルゴリズムの結果より優先度の低い状態もあります。これは、実際に、優先順位がより高い(プライマリー)宛先に対するチャンネルが開始されていても、優先順位がより低い(バックアップ)宛先が選択される可能性があることを意味します。

バックアップ宛先に対してメッセージが送信されないようにする必要がある場合は、CLWLPRTY を使用しないでください。個別のキューを使用するか、または CLWLRANK を使用してプライマリーからバックアップに手動で切り替えることを検討してください。

**関連概念**

同じキューに複数のインスタンスがある場合のクラスタの例

この例では、1 つのキューに複数のインスタンスがあるクラスタにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。1 つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

アプリケーション・プログラミングおよびクラスタ

同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

**関連タスク**

キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

クラスタ内での 2 つのネットワークの使用

以下の指示に従って、2つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

#### クラスター内での1次ネットワークと2次ネットワークの使用

以下の説明に従って、あるネットワークを1次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。バックアップ・ネットワークは、1次ネットワークに問題が発生した場合に使用します。

#### 使用されるチャンネルの数の制限

以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャンネルの数を制限します。

#### キューをホストするより強力なキュー・マネージャーの追加

以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの2倍のメッセージ数を処理できるようにします。

### **使用されるチャンネルの数の制限**

以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャンネルの数を制限します。

### **始める前に**

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも1つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

#### シナリオ

- 価格チェック・アプリケーションはさまざまなキュー・マネージャーにインストールされます。使用されるチャンネルの数を少なく抑えるために、各サーバーが実行するアクティブ・チャンネルの数が制限されます。このアプリケーションは、PRICEQ キューにメッセージが到着すると実行されます。
- 4つのサーバー・キュー・マネージャーが価格チェック・アプリケーションをホストします。2つの照会キュー・マネージャーがメッセージを PRICEQ に送信して価格を照会します。残り2つのキュー・マネージャーは、完全リポジトリとして構成されます。

### **このタスクについて**

以下の手順に従って、使用されるチャンネルの数を制限します。

### **手順**

1. 2つの完全リポジトリを選択します。

価格チェック・クラスターの完全リポジトリとなる2つのキュー・マネージャーを選択します。これらは、REPOS1 および REPOS2 と呼ばれます。

以下のコマンドを発行します。

```
ALTER QMGR REPOS(PRICECHECK)
```

2. 各キュー・マネージャーで CLUSRCVR チャンネルを定義する。

クラスター内の各キュー・マネージャーで、クラスター受信側チャンネルとクラスター送信側チャンネルを定義してください。どちらを最初に定義しても構いません。

```
DEFINE CHANNEL(PRICECHECK.SERVE1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNAME(SERVER1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-receiver channel')
```

3. 各キュー・マネージャーで CLUSSDR チャンネルを定義する。

各キュー・マネージャーで CLUSSDR を定義して、そのキュー・マネージャーをいずれかの完全リポジトリ・キュー・マネージャーにリンクします。

```
DEFINE CHANNEL(PRICECHECK.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(REPOS1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-sender channel to
repository queue manager')
```

4. 価格チェック・アプリケーションをインストールする

5. すべてのサーバー・キュー・マネージャーで PRICEQ キューを定義する

各サーバー・キュー・マネージャーで次のコマンドを発行します。

```
DEFINE QLOCAL(PRICEQ) CLUSTER(PRICECHECK)
```

6. 照会で使用されるチャンネルの数を制限する。

各照会キュー・マネージャーで、次のコマンドを発行して、使用されるアクティブ・チャンネルの数を制限します。

```
ALTER QMGR CLWLMRUC(2)
```

7. IBM MQ for z/OS でまだチャンネル・イニシエーターを始動していない場合は始動する。すべてのプラットフォームで、リスナー・プログラムを始動する。

リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャンネルを起動します。

## タスクの結果

350 ページの図 64 は、このタスクによってセットアップされるクラスターを示します。

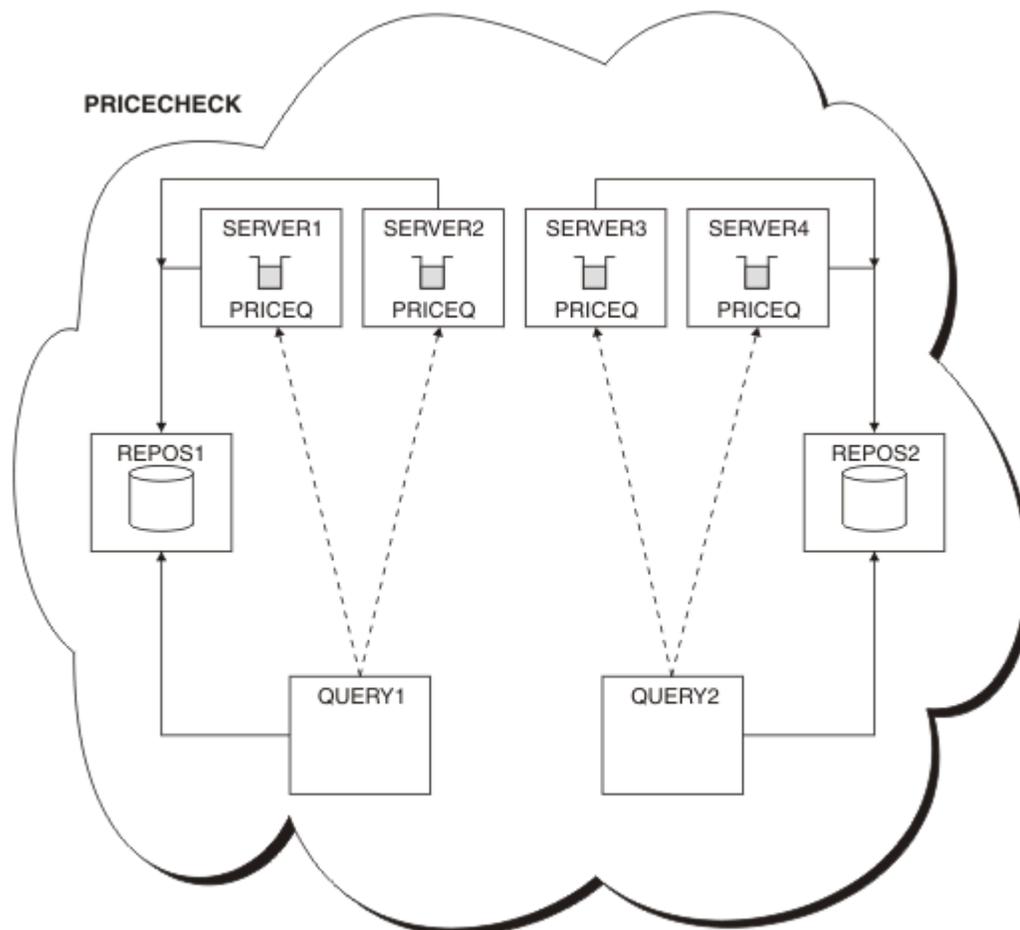


図 64. 4つのサーバー・キュー・マネージャー、2つのリポジトリ、および2つの照会キュー・マネージャーを持つ PRICESCHECK クラスター

PRICECHECK クラスターには、使用可能な PRICEQ キューの 4 つのインスタンスがありますが、各照会キュー・マネージャーはそのうちの 2 つのみを使用します。例えば、QUERY1 キュー・マネージャーは、SERVER1 および SERVER2 キュー・マネージャーへのアクティブ・チャンネルのみを持っています。SERVER1 が使用不可になった場合、QUERY1 キュー・マネージャーは、SERVER3 などの別のキュー・マネージャーを使用し始めます。

## 関連概念

### 同じキューに複数のインスタンスがある場合のクラスターの例

この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。1 つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか 1 つのキュー・マネージャーに送信することを選択したりできます。

### アプリケーション・プログラミングおよびクラスター

同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

## 関連タスク

### キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

### クラスター内での 2 つのネットワークの使用

以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

### クラスター内での 1 次ネットワークと 2 次ネットワークの使用

以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

### バックアップとして機能するキューの追加

以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

### キューをホストするより強力なキュー・マネージャーの追加

以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

## キューをホストするより強力なキュー・マネージャーの追加

以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの 2 倍のメッセージ数を処理できるようにします。

## 始める前に

**注:** クラスターへの変更をクラスター全体に伝搬させるには、少なくとも 1 つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

### シナリオ

- 260 ページの『クラスターにキュー・マネージャーを追加する』での説明に従って、INVENTORY クラスターを設定しました。このクラスターには、3 つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK はどちらも完全リポジトリを保有します。PARIS は部分リポジトリを保有して、INVENTQ からメッセージを書き込みます。在庫管理アプリケーションは、ニューヨークのシステムで動作します。このシステムは、キュー・マネージャー NEWYORK に接続されています。このアプリケーションは、INVENTQ キューのメッセージが到着すると実行されます。

- ロサンゼルスでは、新しい店舗の準備が進められています。処理能力を増強するために、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させます。新しいキュー・マネージャーは、ニューヨークの2倍のメッセージ数を処理できます。

## このタスクについて

以下の手順に従って、キューをホストする、より強力なキュー・マネージャーを追加します。

### 手順

1. LOSANGELES が最初に参照する完全リポジトリを決める。
2. クラスター内の各キュー・マネージャーは、クラスターに関する情報を収集するために、いずれかの完全リポジトリを参照する必要があります。キュー・マネージャーは、独自の部分リポジトリを作成します。どのリポジトリが選択されるかは、あまり重要ではありません。この例では、NEWYORK が選択されます。新しいキュー・マネージャーをクラスターに追加すると、このキュー・マネージャーは両方のリポジトリと通信します。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from LOSANGELES to repository at NEWYORK')
```

3. キュー・マネージャー LOSANGELES に CLUSRCVR チャンネルを定義する。  
クラスター内の各キュー・マネージャーにクラスター受信側チャンネルを定義する必要があります。このチャンネルにより、キュー・マネージャーはメッセージを受信することができます。LOSANGELES には、以下のように定義します。

```
DEFINE CHANNEL(INVENTORY.LOSANGELES) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(LOSANGELES.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager LOSANGELES')
CLWLWGHT(2)
```

クラスター受信側チャンネルは、このキュー・マネージャーがクラスター INVENTORY 内の他のキュー・マネージャーからメッセージを受信できる状態になったことを通知します。CLWLWGHT を 2 に設定すると、ロサンゼルスのカスタマー・マネージャーがニューヨークの2倍の在庫メッセージを受け取れるようになります (NEWYORK のチャンネルを 1 に設定した場合)。

4. キュー・マネージャー NEWYORK で CLUSRCVR チャンネルを変更する。  
ロサンゼルスのカスタマー・マネージャーがニューヨークの2倍の在庫メッセージを受け取るようにします。そのために、クラスター受信側チャンネルの定義を変更します。  

```
ALTER CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) CLWLWGHT(1)
```
5. 在庫管理アプリケーションを調べてメッセージの類縁性を調べる。  
これを調べる前に、在庫管理アプリケーションがメッセージの処理順序に依存しないことを確認してください。
6. 在庫管理アプリケーションをロサンゼルスのシステムにインストールする。
7. クラスター・キュー INVENTQ を定義する。

INVENTQ キューは、キュー・マネージャー NEWYORK に既にホストされていますが、LOSANGELES によってもホストされます。キュー・マネージャー LOSANGELES に次のように INVENTQ キューを定義します。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

これですべての定義が完了しました。IBM MQ for z/OS でまだチャンネル・イニシエーターを開始していない場合は開始します。すべてのプラットフォームで、キュー・マネージャー LOSANGELES 上でリスナー・プログラムを開始します。リスナー・プログラムは、着信ネットワーク要求がないかどうか listen し、必要な場合にクラスター受信側チャンネルを起動します。

## タスクの結果

351 ページの『キューをホストするより強力なキュー・マネージャーの追加』は、このタスクによってセッティングされるクラスターを示します。

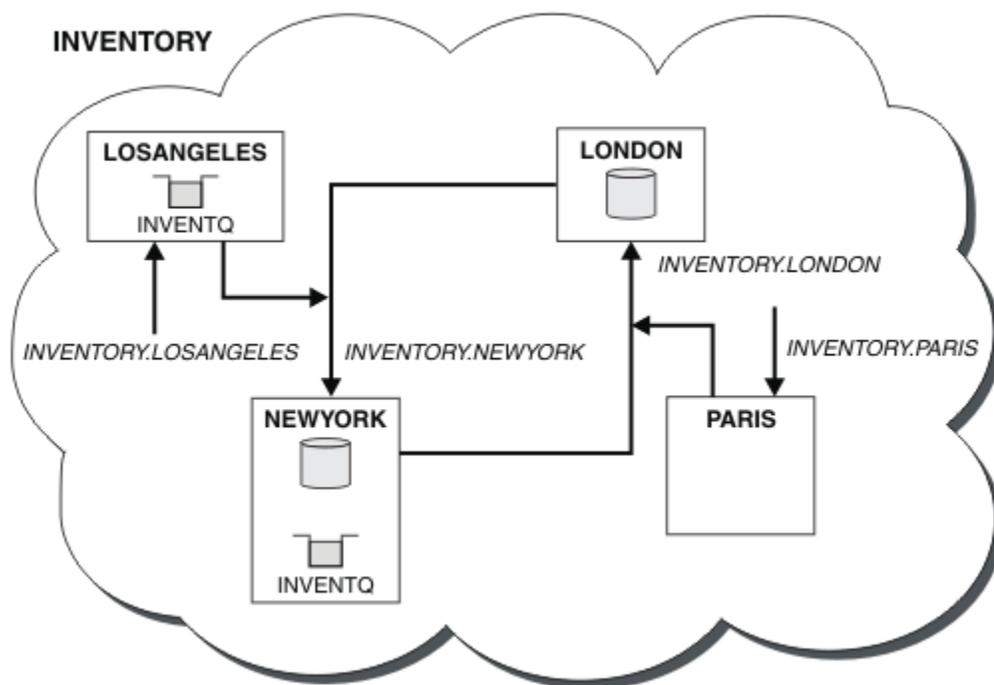


図 65. 4つのキュー・マネージャーが格納されている INVENTORY クラスター

クラスターをこのように変更するとき、キュー・マネージャー LONDON および PARIS を変更する必要はありませんでした。これらのキュー・マネージャーのリポジトリは、LOSANGELES の INVENTQ にメッセージを送信できるように設定するために必要な情報によって、自動的に更新されます。

## 次のタスク

INVENTQ キューと在庫管理アプリケーションは、このクラスター内の2つのキュー・マネージャーによってホストされます。この構成により、可用性が増大してメッセージのスループットが向上し、ワークロードが2つのキュー・マネージャーに分散されるようになります。LOSANGELES または NEWYORK によって INVENTQ に書き込まれたメッセージは、可能なときは必ずローカル・キュー・マネージャーのインスタンスによって処理されます。LONDON または PARIS によって書き込まれたメッセージは、LOSANGELES または NEWYORK に経路指定され、2倍のメッセージ数が LOSANGELES に送信されます。

### 関連概念

#### 同じキューに複数のインスタンスがある場合のクラスターの例

この例では、1つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。1つのメッセージをキューの特定のインスタンスに強制的に送信したり、一連のメッセージをいずれか1つのキュー・マネージャーに送信することを選択したりできます。

#### アプリケーション・プログラミングおよびクラスター

同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

### 関連タスク

[キューをローカル側でホストするキュー・マネージャーの追加](#)

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

#### クラスター内での 2 つのネットワークの使用

以下の指示に従って、2 つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

#### クラスター内での 1 次ネットワークと 2 次ネットワークの使用

以下の説明に従って、あるネットワークを 1 次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。バックアップ・ネットワークは、1 次ネットワークに問題が発生した場合に使用します。

#### バックアップとして機能するキューの追加

以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

#### 使用されるチャンネルの数の制限

以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャンネルの数を制限します。

## アプリケーション・プログラミングおよびクラスター

同じキューの複数インスタンスを活用するために、プログラミング変更を行う必要はありません。ただし、一部のプログラムは、メッセージのシーケンスがキューの同じインスタンスに送信されない限り、正しく動作しません。

アプリケーションは MQOPEN 呼び出しを使用して、キューをオープンすることができます。アプリケーションは MQPUT 呼び出しを使用して、メッセージをオープン・キューに書き込みます。また、MQPUT1 呼び出しを使用すると、まだオープンしていないキューに単一のメッセージを書き込むことができます。

同じキューの複数のインスタンスを持つクラスターを設定する場合、アプリケーション・プログラミングの考慮事項は特にありません。しかし、クラスター化をワークロード管理に役立てるためには、アプリケーションの変更が必要な場合があります。同じキューの複数の定義があるネットワークを設定する場合は、メッセージの類縁性について、アプリケーションを確認してください。

例えば、質問と回答の形式の一連のメッセージの流れを使用する 2 つのアプリケーションがあるとします。質問を送信したキュー・マネージャーと同じキュー・マネージャーに回答を戻すことが想定されます。この場合、ワークロード管理ルーチンが、応答キューのコピーをホストしているどのキュー・マネージャーにもメッセージを送信しないことが重要です。

メッセージの順次処理を必要とするアプリケーションもあります (例えば、順番に取り出す必要があるメッセージのバッチを送信するデータベース複製アプリケーション)。セグメント化されたメッセージを使用するときも、類縁性の問題が起こることがあります。

## ローカル・バージョンまたはリモート・バージョンの宛先キューのオープン

キュー・マネージャーが、ローカル・バージョンまたはリモート・バージョンの宛先キューのいずれを使用するかを選択する方法を理解してください。

1. キュー・マネージャーは、メッセージの読み取りまたはキューの属性設定を行う場合には、ローカル・バージョンの宛先キューをオープンします。
2. 以下の条件が 1 つ以上該当する場合、キュー・マネージャーは、メッセージを書き込む宛先キューの任意のインスタンスをオープンします。
  - ローカル・バージョンの宛先キューが存在していません。
  - キュー・マネージャーが ALTER QMGR で CLWLUSEQ(ANY) を指定しています。
  - キュー・マネージャーのキューが CLWLUSEQ(ANY) を指定しています。

### 関連概念

#### 同じキューに複数のインスタンスがある場合のクラスターの例

この例では、1 つのキューに複数のインスタンスがあるクラスターにおいて、各メッセージがキューの複数のインスタンスにルーティングされます。1 つのメッセージをキューの特定のインスタンスに強制的に送

信したり、一連のメッセージをいずれか1つのキュー・マネージャーに送信することを選択したりできます。

## 関連タスク

### キューをローカル側でホストするキュー・マネージャーの追加

以下の手順に従って、パリおよびニューヨークで在庫管理アプリケーション・システムを実行するための処理能力を増強するために、INVENTQ のインスタンスを追加します。

### クラスター内での2つのネットワークの使用

以下の指示に従って、2つの異なるネットワークが存在する TOKYO に新しい店舗を追加します。どちらのネットワークも、東京でのキュー・マネージャーとの通信に使用できる必要があります。

### クラスター内での1次ネットワークと2次ネットワークの使用

以下の説明に従って、あるネットワークを1次ネットワークにし、別のネットワークをバックアップ・ネットワークにします。バックアップ・ネットワークは、1次ネットワークに問題が発生した場合に使用します。

### バックアップとして機能するキューの追加

以下の説明に従って、現在ニューヨークで稼働している在庫管理システムのバックアップをシカゴで提供します。シカゴのシステムは、ニューヨークのシステムに問題が発生した場合にのみ使用されます。

### 使用されるチャンネルの数の制限

以下の説明に従って、価格チェック・アプリケーションがさまざまなキュー・マネージャーにインストールされている場合に、各サーバーが実行するアクティブ・チャンネルの数を制限します。

### キューをホストするより強力なキュー・マネージャーの追加

以下の説明に従って、ニューヨークだけでなくロサンゼルスでも在庫管理システムを稼働させて、処理能力を増強します。その際、ロサンゼルスではニューヨークの2倍のメッセージ数を処理できるようにします。

## メッセージの類縁性の処理

メッセージの類縁性が、優れたプログラミング設計に含まれることはほとんどありません。クラスタリングを最大限に使用するには、メッセージの類縁性を除去する必要があります。メッセージ類縁性を除去できない場合は、関連メッセージが、必ず同じチャンネルにより同じキュー・マネージャーに送信されるようにすることができます。

メッセージの類縁性があるアプリケーションがある場合、クラスターの使用を開始する前に、メッセージの類縁性を除去してください。

メッセージの類縁性を除去すると、アプリケーションの可用性が向上します。アプリケーションは、メッセージの類縁性があるメッセージのバッチをキュー・マネージャーに送信します。バッチの一部のみを受信した後に、キュー・マネージャーに障害が起こります。送信側のキュー・マネージャーは、メッセージの続きを送信するには、キュー・マネージャーが復旧するのを待ってから、完了していないメッセージのバッチを処理する必要があります。

メッセージの類縁性を除去すると、アプリケーションのスケラビリティも改善されます。類縁性のあるメッセージのバッチは、宛先キュー・マネージャーのリソースを、後続のメッセージを待つ間ロックすることがあります。これらのリソースのロックが長期間になると、他のアプリケーションの処理の妨げになります。

さらに、メッセージの類縁性があると、クラスター・ワークロード管理ルーチンで最適のキュー・マネージャーを選択することができなくなります。

類縁性を除去するには、次の可能性を検討してください。

- メッセージに状態情報を入れておく
- すべてのキュー・マネージャーからアクセスできる不揮発性記憶装置 (例えば、Db2 データベース) に状態情報を保持する
- 複数のキュー・マネージャーからアクセスできるように読み取り専用データを複製する

メッセージの類縁性を除去するためにアプリケーションを変更するのが適切でない場合は、いろいろな解決法があります。

## MQOPEN 呼び出しで特定の宛先を指定する

それぞれの MQOPEN 呼び出しにリモート・キュー名およびキュー・マネージャー名を指定すると、そのオブジェクト・ハンドルを使用してキューに書き込まれるメッセージはすべて同じキュー・マネージャー (ローカル・キュー・マネージャーの場合もある) に送信されます。

それぞれの MQOPEN 呼び出しにリモート・キュー名およびキュー・マネージャー名を指定することには、次の欠点があります。

- ワークロード・バランシングが実行されません。 クラスター・ワークロード・バランシングの利点を利用できません。
- 宛先キュー・マネージャーがリモートであって、そのキュー・マネージャーに複数のチャンネルが接続されている場合、メッセージがいろいろな経路で送信されるため、メッセージの順序が保持されない可能性があります。
- ローカル・キュー・マネージャーに宛先キュー・マネージャーと同じ名前の伝送キューの定義がある場合は、メッセージはその伝送キューに送信され、クラスター伝送キューには送信されません。

## 応答先キュー・マネージャー・フィールドにキュー・マネージャー名を戻す

バッチの最初のメッセージを受け取るキュー・マネージャーは、その名前を応答の中で戻すことができます。これはメッセージ記述子の ReplyToQMgr フィールドを使用して行われます。送信側のキュー・マネージャーは、応答先キュー・マネージャー名を抜き出し、後続のすべてのメッセージにその名前を指定することができます。

応答からの ReplyToQMgr 情報を使用することには、次のような欠点があります。

- 要求中のキュー・マネージャーは最初のメッセージに対する応答を待つ必要がある
- 後続のメッセージを送信する前に、ReplyToQMgr 情報を見つけて使用するために追加のコードを作成する必要があります
- キュー・マネージャーへの経路が複数ある場合、メッセージの順序が保持されないことがある

## MQOPEN 呼び出しで MQOO\_BIND\_ON\_OPEN オプションを設定する

MQOPEN 呼び出しの MQOO\_BIND\_ON\_OPEN オプションを使用して、すべてのメッセージを強制的に同じ宛先に書き込みます。クラスターで メッセージ・グループ を使用する場合は、グループ内のすべてのメッセージが同じ宛先で処理されるように、MQOO\_BIND\_ON\_OPEN または MQOO\_BIND\_ON\_GROUP のいずれかを指定する必要があります。

キューをオープンして MQOO\_BIND\_ON\_OPEN を指定すると、そのキューに送信されるメッセージはすべてそのキューの同じインスタンスに強制的に送信されます。MQOO\_BIND\_ON\_OPEN はすべてのメッセージを同じキュー・マネージャーにバインドし、しかも同じ経路にバインドします。例えば、同じ宛先への経路として IP 経路と NetBIOS 経路がある場合、キューをオープンするときにその 1 つが選択され、獲得されたオブジェクト・ハンドルを使用して同じキューに書き込まれるすべてのメッセージにこの選択が適用されます。

MQOO\_BIND\_ON\_OPEN を指定すると、すべてのメッセージが同じ宛先に経路指定されます。このため、メッセージの類縁性があるアプリケーションが分析されません。宛先が使用できない場合は、それが再び使用可能になるまでメッセージは伝送キューにとどまります。

MQOO\_BIND\_ON\_OPEN は、キューをオープンするときオブジェクト記述子にキュー・マネージャー名を指定する場合にも適用されます。指定したキュー・マネージャーには複数の経路があることがあります。例えば、複数のネットワーク・パスがある場合や、別のキュー・マネージャーが別名を定義している場合があります。MQOO\_BIND\_ON\_OPEN を指定すると、キューをオープンするときに経路が選択されます。

注：これがお勧めする方法です。ただし、キュー・マネージャーがクラスター・キューの別名を通知するマルチホップ構成では、この方法は使用できません。また、アプリケーションがメッセージ・グループごとに同じキュー・マネージャーの別々のキューを使用する場合も、この方法は役に立ちません。

MQOPEN 呼び出しで MQOO\_BIND\_ON\_OPEN を指定する代わりに、キュー定義を変更することもできます。キュー定義に DEFBIND (OPEN) を指定し、MQOPEN 呼び出しの DefBind オプションでデフォルトの MQOO\_BIND\_AS\_Q\_DEF を使用するようにします。

## MQOPEN 呼び出しで MQOO\_BIND\_ON\_GROUP オプションを設定する

MQOPEN 呼び出しの MQOO\_BIND\_ON\_GROUP オプションを使用して、グループ内のすべてのメッセージを強制的に同じ宛先へ書き込みます。クラスターで メッセージ・グループ を使用する場合は、グループ内のすべてのメッセージが同じ宛先で処理されるように、MQOO\_BIND\_ON\_OPEN または MQOO\_BIND\_ON\_GROUP のいずれかを指定する必要があります。

キューをオープンして MQOO\_BIND\_ON\_GROUP を指定すると、そのキューに送信されるグループのメッセージはすべてそのキューの同じインスタンスに強制的に送信されます。MQOO\_BIND\_ON\_GROUP はグループのすべてのメッセージを同じキュー・マネージャーにバインドし、しかも同じ経路にバインドします。例えば、同じ宛先への経路として IP 経路と NetBIOS 経路がある場合、キューをオープンするときにその 1 つが選択され、獲得されたオブジェクト・ハンドルを使用して同じキューに書き込まれるグループのすべてのメッセージにこの選択が適用されます。

MQOO\_BIND\_ON\_GROUP を指定すると、グループのすべてのメッセージが強制的に同じ宛先に経路指定されます。このため、メッセージの類縁性があるアプリケーションが分析されません。宛先が使用できない場合は、それが再び使用可能になるまでメッセージは伝送キューにとどまります。

MQOO\_BIND\_ON\_GROUP は、キューをオープンするときオブジェクト記述子にキュー・マネージャー名を指定する場合にも適用されます。指定したキュー・マネージャーには複数の経路があることがあります。例えば、複数のネットワーク・パスがある場合や、別のキュー・マネージャーが別名を定義している場合があります。MQOO\_BIND\_ON\_GROUP を指定すると、キューをオープンするときに経路が選択されます。

MQOO\_BIND\_ON\_GROUP を有効にするには、MQPUT に MQPMO\_LOGICAL\_ORDER Put オプションを組み込む必要があります。メッセージの MQMD の **GroupId** を MQGI\_NONE に設定できます。また、メッセージの MQMD **MsgFlags** フィールド内に次のメッセージ・フラグを含める必要があります。

- グループの最後のメッセージ: MQMF\_LAST\_MSG\_IN\_GROUP
- グループの他のすべてのメッセージ: MQMF\_MSG\_IN\_GROUP

MQOO\_BIND\_ON\_GROUP が指定されていても、メッセージがグループ化されていない場合、MQOO\_BIND\_NOT\_FIXED が指定された場合と同じ振る舞いになります。

注: これはグループ内のメッセージが同じ宛先に送信されることを確実にする推奨手法です。ただし、キュー・マネージャーがクラスター・キューの別名を通知するマルチホップ構成では機能しません。

MQOPEN 呼び出しで MQOO\_BIND\_ON\_GROUP を指定する代わりに、キュー定義を変更することもできます。キュー定義に DEFBIND(GROUP) を指定し、MQOPEN 呼び出しの DefBind オプションでデフォルトの MQOO\_BIND\_AS\_Q\_DEF を使用するようになります。

## カスタマイズされたクラスター・ワークロード出口プログラムを作成する

アプリケーションを変更する代わりに、クラスター・ワークロード出口プログラムを作成して、メッセージの類縁性の問題を回避することもできます。クラスター・ワークロード出口プログラムを作成することは簡単ではないので、お勧めできる解決策ではありません。このプログラムは、メッセージの内容を検査して類縁性を認識するように設計する必要があります。そして、類縁性の認識に基づいて、ワークロード管理ユーティリティーにすべての関連メッセージを同じキュー・マネージャーに転送させる必要があります。

## パブリッシュ/サブスクライブ・メッセージングの構成

キュー型パブリッシュ/サブスクライブの開始、停止、および状況の表示を行えます。ストリームの追加および除去、ブローカー階層に対するキュー・マネージャーの追加および削除も行えます。

### 手順

- キュー型パブリッシュ/サブスクライブの制御の詳細については、以下のサブトピックを参照してください。
  - [358 ページの『キューに入れられたパブリッシュ/サブスクライブのメッセージ属性の設定』](#)

- 359 ページの『キューに入れられたパブリッシュ/サブスクライブの開始』
- 360 ページの『キューに入れられたパブリッシュ/サブスクライブの停止』
- 360 ページの『ストリームの追加』
- 361 ページの『ストリームの削除』
- 362 ページの『サブスクリプション・ポイントの追加』
- 371 ページの『パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合』

## キューに入れられたパブリッシュ/サブスクライブのメッセージ属性の設定

パブリッシュ/サブスクライブ・メッセージの一部の属性の振る舞いは、キュー・マネージャー属性によって制御します。その他の属性は、qm.ini ファイルの *Broker* スタンザで制御します。

### このタスクについて

以下のパブリッシュ/サブスクライブ属性を設定することができます。詳しくは、キュー・マネージャーのパラメーターを参照してください

表 25. パブリッシュ/サブスクライブ構成パラメーター	
説明	MQSC パラメーター名
コマンド・メッセージの再試行回数	PSRTYCNT
配信不能なコマンド入力メッセージの廃棄	PSNPMSG
配信不能なコマンド応答メッセージの後の振る舞い	PSNPRES
同期点でのコマンド・メッセージの処理	PSSYNCPT

Broker スタンザは、以下の構成設定を管理するために使用します。

- PersistentPublishRetry=yes | force

Yes を指定した場合に、キュー型パブリッシュ/サブスクライブ・インターフェース経由の持続メッセージのパブリケーションに失敗すると、否定応答を要求していなければ、パブリッシュ操作が再試行されません。

否定応答メッセージを要求していた場合は、否定応答が送信され、それ以上の再試行は行われません。

Force を指定した場合に、キュー型パブリッシュ/サブスクライブ・インターフェース経由の持続メッセージのパブリケーションに失敗すると、正常に処理されるまでパブリッシュ操作が再試行されます。否定応答は送信されません。

- NonPersistentPublishRetry=yes | force

Yes を指定した場合に、キュー型パブリッシュ/サブスクライブ・インターフェース経由の非持続メッセージのパブリケーションに失敗すると、否定応答を要求していなければ、パブリッシュ操作が再試行されません。

否定応答メッセージを要求していた場合は、否定応答が送信され、それ以上の再試行は行われません。

Force を指定した場合に、キュー型パブリッシュ/サブスクライブ・インターフェース経由の非持続メッセージのパブリケーションに失敗すると、正常に処理されるまでパブリッシュ操作が再試行されます。否定応答は送信されません。

**注:** 非持続メッセージに対してこの機能を有効にする場合は、NonPersistentPublishRetry 値の設定に加えて、必ず、キュー・マネージャー属性 **PSSYNCPT** も Yes に設定してください。

同期点では STREAM キューの **MQGET** が実行されるようになったため、これを行うと、非持続パブリケーションの処理のパフォーマンスにも影響を与える可能性があります。

- PublishBatchSize=number

ブローカーは通常、同期点でパブリッシュ・メッセージを処理します。パブリケーションごとに個別にコミットすると効率が落ちる可能性があり、状況によっては、ブローカーは複数のパブリッシュ・メッセージを1つの作業単位として処理できます。このパラメーターでは、1つの作業単位で処理できるパブリッシュ・メッセージの最大数を指定します。

PublishBatchSize のデフォルト値は5です。

- PublishBatchInterval=*number*

ブローカーは通常、同期点でパブリッシュ・メッセージを処理します。パブリケーションごとに個別にコミットすると効率が落ちる可能性があり、状況によっては、ブローカーは複数のパブリッシュ・メッセージを1つの作業単位として処理できます。このパラメーターでは、バッチに含まれている最初のメッセージから、その同じバッチに含まれている後続パブリケーションまでの間の最大時間をミリ秒単位で指定します。

メッセージがすぐに使用可能であれば、バッチの間隔を0にすると、最大 PublishBatchSize 個のメッセージを処理できます。

PublishBatchInterval のデフォルト値はゼロです。

## 手順

IBM MQ エクスプローラー、プログラマブル・コマンド、または **runmqsc** コマンドを使用して、パブリッシュ/サブスクライブの振る舞いを制御するキュー・マネージャー属性を変更します。

### 例

```
ALTER QMGR PSNPRES(SAFE)
```

## キューに入れられたパブリッシュ/サブスクライブの開始

キュー・マネージャーの PSMODE 属性を設定することで、キューに入れられたパブリッシュ/サブスクライブを開始します。

### 始める前に

パブリッシュ/サブスクライブの次の3つのモードを理解するには、[PSMODE](#) の説明を参照してください。

- COMPAT
- DISABLED
- ENABLED

### このタスクについて

QMGR PSMODE 属性を設定して、キューに入れられたパブリッシュ/サブスクライブ・インターフェース (ブローカーとも呼ばれる) またはパブリッシュ/サブスクライブ・エンジン (バージョン7パブリッシュ/サブスクライブとも呼ばれる) のいずれか、あるいはその両方を開始します。キューに入れられたパブリッシュ/サブスクライブを開始するには、PSMODE を ENABLED に設定する必要があります。デフォルトは ENABLED です。

## 手順

キューに入れられたパブリッシュ/サブスクライブ・インターフェースがまだ使用可能になっていなければ、IBM MQ Explorer または **runmqsc** コマンドを使用して使用可能にします。

### 例

```
ALTER QMGR PSMODE (ENABLED)
```

## 次のタスク

IBM MQ は、キューに入れられたパブリッシュ/サブスクライブ・コマンドおよびパブリッシュ/サブスクライブの Message Queue Interface (MQI) 呼び出しを処理します。

## キューに入れられたパブリッシュ/サブスクライブの停止

キュー・マネージャーの PSMODE 属性を設定することで、キューに入れられたパブリッシュ/サブスクライブを停止します。

### 始める前に

パブリッシュ/サブスクライブの次の 3 つのモードを理解するには、[PSMODE](#) の説明を参照してください。

- COMPAT
- DISABLED
- ENABLED

### このタスクについて

QMGR PSMODE 属性を設定して、キューに入れられたパブリッシュ/サブスクライブ・インターフェース (ブローカーとも呼ばれる) またはパブリッシュ/サブスクライブ・エンジン (バージョン 7 パブリッシュ/サブスクライブとも呼ばれる) のいずれか、あるいはその両方を停止します。キューに入れられたパブリッシュ/サブスクライブを停止するには、PSMODE を COMPAT に設定する必要があります。キューに入れられたパブリッシュ/サブスクライブ・エンジンを完全に停止するには、PSMODE を DISABLED に設定します。

### 手順

キューに入れられたパブリッシュ/サブスクライブ・インターフェースを無効にするには、IBM MQ Explorer または `runmqsc` コマンドを使用します。

### 例

```
ALTER QMGR PSMODE (COMPAT)
```

## ストリームの追加

アプリケーション間でデータを分離できるようにするため、または IBM WebSphere MQ 6 パブリッシュ/サブスクライブ階層の相互協調処理を可能にするために、ストリームを手動で追加することができます。

### 始める前に

パブリッシュ/サブスクライブ・ストリームの動作についてよく理解しておいてください。[ストリームおよびトピック](#)を参照します。

### このタスクについて

PCF コマンド、`runmqsc`、または IBM MQ Explorer を使用して、以下のステップを実行します。

注: ステップ 1 と 2 は、任意の順序で実行できます。ステップ 3 だけは、ステップ 1 と 2 の両方が完了した後に実行してください。

### 手順

1. IBM WebSphere MQ 6 のストリームと同じ名前のローカル・キューを定義します。
2. IBM WebSphere MQ 6 のストリームと同じ名前のローカル・トピックを定義します。
3. 名前リスト SYSTEM.QPUBSUB.QUEUE.NAMELIST にキューの名前を追加します。
4. パブリッシュ/サブスクライブ階層内にある IBM WebSphere MQ 7.1 以上のすべてのキュー・マネージャーについて繰り返します。

## 'Sport' の追加

ストリーム 'Sport' を共有する例では、IBM WebSphere MQ 6 キュー・マネージャーと 7.1 キュー・マネージャーが同じパブリッシュ/サブスクライブ階層で動作しています。IBM WebSphere MQ 6 の各キュー・マネージャーは、'Sport' というストリームを共有します。例は、IBM WebSphere MQ 6 のストリーム 'Sport' と共有のトピック・ストリング 'Sport' を使用して、IBM WebSphere MQ 7.1 のキュー・マネージャー上に 'Sport' というキューおよびトピックを作成する方法を示しています。

IBM WebSphere MQ 7.1 のパブリッシュ・アプリケーションが、トピック・ストリング 'Soccer/Results' を使用してトピック 'Sport' にパブリッシュを行うと、結果としてトピック・ストリング 'Sport/Soccer/Results' が作成されます。IBM WebSphere MQ 7.1 のキュー・マネージャーでは、トピック・ストリング 'Soccer/Results' があるトピック 'Sport' へのサブスクライバーが、パブリケーションを受け取ります。

IBM WebSphere MQ 6 のキュー・マネージャーでは、トピック・ストリング 'Soccer/Results' があるストリーム 'Sport' へのサブスクライバーが、パブリケーションを受け取ります。

```
runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
  1 : define qlocal('Sport')
AMQ8006: IBM MQ queue created.
define topic('Sport') topicstr('Sport')
  2 : define topic('Sport') topicstr('Sport')
AMQ8690: IBM MQ topic created.
alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
  3 : alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
AMQ8551: IBM MQ namelist changed.
```

**注:** `alter namelist` コマンドには、名前リスト・オブジェクト内の既存の名前と、追加する新規の名前の両方を指定する必要があります。

## 次のタスク

ストリームに関する情報が、階層内の他のブローカーに渡されます。

ブローカーがバージョン 6 の場合は、IBM WebSphere MQ 6 ブローカーとして管理します。つまり、ストリーム・キューを手動で作成するか、ストリーム・キューを必要に応じてブローカーに動的に作成させるかを選択できるということです。キューは、モデル・キュー定義 `SYSTEM.BROKER.MODEL.STREAM` に基づいています。

ブローカーがバージョン 7.1 の場合は、階層内の各 IBM WebSphere MQ 7.1 キュー・マネージャーを手動で構成する必要があります。

## ストリームの削除

IBM WebSphere MQ 7.1 以降のキュー・マネージャーから、ストリームを削除することができます。

### 始める前に

ストリームを削除する前に、そのストリームへのサブスクリプションが残っていないことを確認し、そのストリームを使用するすべてのアプリケーションを静止する必要があります。削除されたストリームにパブリケーションが流れ続けた場合、システムをクリーンな処理状態に復元するために、多くの管理努力が必要となります。

### 手順

1. このストリームをホストする、すべての接続済みのブローカーを検索します。
2. すべてのブローカー上にある、このストリームへのすべてのサブスクリプションを取り消します。
3. 名前リスト `SYSTEM.QPUBSUB.QUEUE.NAMELIST` からキュー (ストリームと同じ名前を持つもの) を削除します。

4. ストリームと同じ名前を持つキューからのすべてのメッセージを削除または消去します。
5. ストリームと同じ名前を持つキューを削除します。
6. 関連するトピック・オブジェクトを削除します。

## 次のタスク

そのストリームをホスティングしている、他のすべての接続されている IBM WebSphere MQ 7.1 以降のキュー・マネージャーに対して、ステップ 3 から 5 までを繰り返します。

## サブスクリプション・ポイントの追加

IBM Integration Bus の旧バージョンから移行した既存のキュー型パブリッシュ/サブスクライブ・アプリケーションを、新規のサブスクリプション・ポイントで拡張する方法を示します。

### 始める前に

1. サブスクリプション・ポイントが `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST` でまだ定義されていないことを確認してください。
2. サブスクリプション・ポイントと同じ名前を持つトピック・オブジェクトまたはトピック・ストリングが存在するかどうかを確認します。

### このタスクについて

IBM WebSphere MQ 7.1 (およびそれ以降) アプリケーションはサブスクリプション・ポイントを使用しませんが、サブスクリプション・ポイント移行メカニズムを使用して、サブスクリプション・ポイントを使用する既存のアプリケーションと相互運用することができます。

**重要:** サブスクリプション・ポイント移行メカニズムは、IBM MQ 8.0 から削除されています。既存のアプリケーションを移行する必要がある場合は、使用している製品のバージョン用の資料に示された手順を実行してから、最新バージョンに移行する必要があります。

MQRFH1 ヘッダーを使用するキュー型パブリッシュ/サブスクライブするプログラム (または IBM WebSphere MQ 6 以前から移行されたプログラム) では、サブスクリプションポイントは機能しません。

IBM WebSphere MQ 7.1 以降用に作成された統合パブリッシュ/サブスクライブ・アプリケーションを使用するためにサブスクリプション・ポイントを追加する必要はありません。

### 手順

1. サブスクリプション・ポイントの名前を `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST` に追加します。
  - z/OS では、**NLTYPE** はデフォルトで `NONE` になっています。
  - 同じパブリッシュ/サブスクライブ・トポロジー内で接続されているすべてのキュー・マネージャーに対して、このステップを繰り返します。
2. サブスクリプション・ポイントの名前と一致するトピック・ストリングを持つトピック・オブジェクトを追加し、できればこれにサブスクリプション・ポイントの名前を付けます。
  - サブスクリプション・ポイントがクラスター内にある場合は、トピック・オブジェクトをクラスター・トピック・ホスト上のクラスター・トピックとして追加します。
  - サブスクリプション・ポイントの名前と同じトピック・ストリングを持つトピック・オブジェクトが存在している場合は、その既存のトピック・オブジェクトを使用します。サブスクリプション・ポイントが既存のトピックを再利用した場合の結果について、理解しておく必要があります。既存のトピックが既存のアプリケーションの一部である場合には、同じ名前を持つ 2 つのトピック間の衝突を解決する必要があります。
  - サブスクリプション・ポイントと同じ名前を持つトピック・オブジェクトは存在しているものの、トピック・ストリングが異なる場合は、別の名前を持つトピックを作成します。
3. **Topic** の属性 `WILDCARD` を値 `BLOCK` に設定します。

#または\*のサブスクリプションをブロック化すると、ワイルドカード・サブスクリプションがサブスクリプション・ポイントに分離されます。[ワイルドカードおよびサブスクリプション・ポイント](#)を参照してください。

4. トピック・オブジェクトで、任意の必要な属性を設定します。

## 例

この例では、2つのサブスクリプション・ポイント USD と GBP を追加する `runmqsc` コマンド・ファイルを示します。

```
DEFINE TOPIC(USD) TOPICSTR(USD)
DEFINE TOPIC(GBP) TOPICSTR(GBP) WILDCARD(BLOCK)
ALTER NL(SYSTEM.QPUBSUB.SUBPOINT.NAMELIST) NAMES(SYSTEM.BROKER.DEFAULT.SUBPOINT, USD, GBP)
```

## 注:

1. デフォルトのサブスクリプション・ポイントを、**ALTER** コマンドを使用して追加されたサブスクリプション・ポイントのリストに組み込みます。**ALTER** は、名前リスト内の既存の名前を削除します。
2. 名前リストを変更する前に、トピックを削除します。キュー・マネージャーが名前リストを検査するのは、キュー・マネージャーの開始時および名前リストの変更時のみです。

## 分散パブリッシュ/サブスクライブ・ネットワークの構成

分散パブリッシュ/サブスクライブ・トポロジーにまとめて接続されるキュー・マネージャーは、共通のフェデレーテッド・トピック・スペースを共有します。1つのキュー・マネージャー上で作成されるサブスクリプションは、トポロジー内の別のキュー・マネージャーに接続されたアプリケーションによってパブリッシュされるメッセージを受け取ることができます。

クラスターまたは階層の中で複数のキュー・マネージャーをまとめて接続することによって、作成されるトピック・スペースの範囲を制御できます。パブリッシュ/サブスクライブ・クラスターでは、クラスター内のトピック・スペースの各ブランチで、トピック・オブジェクトを「クラスター化」する必要があります。階層内では、階層の「親」を識別するように各キュー・マネージャーを構成する必要があります。

それぞれのパブリケーションおよびサブスクリプションをローカルまたはグローバルとして選択することにより、トポロジー内のパブリケーションおよびサブスクリプションのフローをさらに制御できます。ローカル・パブリケーションおよびサブスクリプションは、パブリッシャーまたはサブスクライバーの接続先であるキュー・マネージャーを超えて伝搬されることはありません。

### 関連情報

[分散パブリッシュ/サブスクライブのネットワーク](#)

[パブリケーション有効範囲](#)

[サブスクリプション有効範囲](#)

[トピック・スペース](#)

[クラスター・トピックの定義](#)

## パブリッシュ/サブスクライブ・クラスターの構成

キュー・マネージャーでトピックを定義します。トピックをクラスター・トピックにするには、**CLUSTER** プロパティを設定します。このトピックのパブリケーションとサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティを設定します。

## 始める前に

一部のクラスター構成では、直接ルーティング型パブリッシュ/サブスクライブのオーバーヘッドに対応できません。この構成を使用する前に、[パブリッシュ/サブスクライブ・クラスターの設計](#)で詳述されている考慮事項およびオプションを調べてください。

クラスターへの変更をクラスター全体に伝搬させるには、少なくとも1つの完全リポジトリが常に使用可能でなければなりません。この作業を開始する前に、リポジトリが使用可能であることを確認してください。

[パブリッシュ/サブスクライブ・クラスターのルーティング: 動作に関する注意も参照してください。](#)

シナリオ

- 260 ページの『[クラスターにキュー・マネージャーを追加する](#)』での説明に従って、INVENTORY クラスターを設定しました。このクラスターには、3つのキュー・マネージャーが格納されています。このうち、LONDON と NEWYORK は完全リポジトリを保有し、PARIS は部分リポジトリを保有しています。

## このタスクについて

クラスター内のキュー・マネージャーにトピックを定義する際に、トピックがクラスター・トピックであるかどうかを指定する必要があります。クラスター・トピックである場合、このトピックのパブリケーションおよびサブスクリプションに使用するクラスター内のルーティングも指定する必要があります。トピックをクラスター・トピックにするには、TOPIC オブジェクトの **CLUSTER** プロパティをクラスターの名前を使用して構成します。クラスター内の1つのキュー・マネージャーにクラスター・トピックを定義することによって、そのトピックをクラスター全体で使用できるようにします。クラスター内で使用するメッセージ・ルーティングを選択するには、TOPIC オブジェクトの **CLROUTE** プロパティを以下のいずれかの値に設定します。

- **DIRECT**
- **TOPICHOST**

デフォルトでは、トピック・ルーティングは **DIRECT** です。これは IBM MQ 8.0 より前の唯一のオプションでした。直接経路指定されたクラスター・トピックをキュー・マネージャーで構成すると、クラスター内のすべてのキュー・マネージャーがクラスター内の他のすべてのキュー・マネージャーを認識するようになります。各キュー・マネージャーは、パブリッシュ操作およびサブスクライブ操作を実行するときに、クラスター内の他のすべてのキュー・マネージャーに直接接続できます。[直接ルーティング型パブリッシュ/サブスクライブ・クラスター](#)を参照してください。

IBM MQ 8.0 以降、代わりにトピックの経路指定を **TOPICHOST** として構成できるようになりました。トピック・ホスト経路指定を使用すると、クラスター内のすべてのキュー・マネージャーは、経路指定されたトピック定義をホストするクラスター・キュー・マネージャー（つまり、トピック・オブジェクトを定義したキュー・マネージャー）を認識するようになります。パブリッシュ操作およびサブスクライブ操作を行うとき、クラスター内のキュー・マネージャーは、それらのトピック・ホスト・キュー・マネージャーにのみ接続し、相互に直接接続されることはありません。トピック・ホスト・キュー・マネージャーは、パブリケーションがパブリッシュされるキュー・マネージャーから、一致するサブスクリプションがあるキュー・マネージャーへのパブリケーションの経路指定を担当します。[トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスター](#)を参照してください。

注：トピック・オブジェクトがクラスター化された後（**CLUSTER** プロパティを設定することによって）、**CLROUTE** プロパティの値を変更することはできません。値を変更するには、その前にオブジェクトのクラスター化を解除（**CLUSTER** を ' ' に設定）する必要があります。トピックのクラスター化を解除すると、トピック定義はローカル・トピックに変換されます。これによって、パブリケーションがリモート・キュー・マネージャーのサブスクリプションに送信されない期間ができます。この変更を行う場合は、この点を考慮する必要があります。[別のキュー・マネージャーのクラスター・トピックと同じ名前で非クラスター・トピックを定義する効果を参照してください。](#)クラスター化されている状態で **CLROUTE** プロパティの値を変更しようとすると、システムは `MQRCCF_CLROUTE_NOT_ALTERABLE` 例外を生成します。

## 手順

1. トピックをホストするキュー・マネージャーを選択します。

どのクラスター・キュー・マネージャーでもトピックをホストできます。3つのキュー・マネージャー（LONDON、NEWYORK、または PARIS）のいずれかを選択し、TOPIC オブジェクトのプロパティを構成します。直接ルーティングを使用する場合、どのキュー・マネージャーを選択しても操作に違いは生じません。トピック・ホスト・ルーティングを使用する場合、選択したキュー・マネージャーには、パブリケーションのルーティングを行う追加の責任が生じます。そのため、トピック・ホスト・ルーティングにおいては、性能の良いシステムにホストされる、ネットワーク接続の状態が良いキュー・マネージャーを選択してください。

## 2. キュー・マネージャーでトピックを定義します。

トピックをクラスター・トピックにするには、トピックを定義するときにクラスター名を使用し、このトピックのパブリケーションおよびサブスクリプションに使用するルーティングを設定します。例えば、LONDON キュー・マネージャー上の直接ルーティング・クラスター・トピックを作成するには、次のようにトピックを作成します。

```
DEFINE TOPIC(INVENTORY) TOPICSTR('/INVENTORY') CLUSTER(INVENTORY) CLROUTE(DIRECT)
```

クラスター内の1つのキュー・マネージャーにクラスター・トピックを定義することによって、そのトピックをクラスター全体で使用できるようにします。

**CLROUTE** の使用について詳しくは、[DEFINE TOPIC \(CLROUTE\)](#) および [パブリッシュ/サブスクライブ・クラスターのルーティング: 動作に関する注意](#) を参照してください。

## タスクの結果

クラスターは、トピックのパブリケーションおよびサブスクリプションを受信する準備が整いました。

## 次のタスク

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターを構成した場合には、このトピック用にたいいて2つ目のトピック・ホストを追加することになります。367 ページの『[トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加](#)』を参照してください。

お客様の組織が地理的に分散しているなどの理由で、いくつかの別個のパブリッシュ/サブスクライブ・クラスターが存在する場合、一部のクラスター・トピックをすべてのクラスターに伝搬することもできます。階層でクラスターに接続すると、これを行うことができます。374 ページの『[複数のクラスターのトピック・スペースの結合](#)』を参照してください。あるクラスターから別のクラスターへ流れるパブリケーションを制御することもできます。376 ページの『[複数のクラスター内でのトピック・スペースの結合および分離](#)』を参照してください。

## 関連概念

### [パブリケーションとサブスクリプションの有効範囲の結合](#)

IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

### [パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合](#)

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

## 関連タスク

### [別のキュー・マネージャーへのクラスター・トピック定義の移動](#)

トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

### [トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加](#)

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。この方式を利用して、アベイラビリティとワークロード・バランスを向上させることができます。同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

### [パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続](#)

階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

[パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断](#)

パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

## 関連情報

[パブリッシュ/サブスクライブ・クラスターの設計](#)

[分散パブリッシュ/サブスクライブのトラブルシューティング](#)

[クラスター化されたパブリッシュ/サブスクライブの禁止](#)

## 別のキュー・マネージャーへのクラスター・トピック定義の移動

トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

## このタスクについて

1つのクラスター内に、同じクラスター・トピック・オブジェクトの複数の定義を持つことができます。これは、トピック・ホスト・ルーティング型クラスターにとっては通常の状態であり、直接ルーティング型クラスターにとっては異常な状態です。詳しくは、[同じ名前の複数のクラスター・トピック定義を参照](#)してください。

パブリケーションのフローを中断せずに、クラスター内の別のキュー・マネージャーにクラスター・トピック定義を移動させるには、以下のステップを実行します。この手順では、定義をキュー・マネージャー QM1 からキュー・マネージャー QM2 に移動します。

## 手順

1. クラスター・トピック定義の複製を QM2 上に作成します。

直接ルーティング型の場合は、すべての属性を QM1 の定義と一致するように設定します。

トピック・ホスト・ルーティング型の場合は、まず PUB(DISABLED) として新しいトピック・ホストを定義します。これによって、QM2 はクラスター内のサブスクリプションを認識できますが、ルーティング・パブリケーションを開始することはできません。

2. クラスター内に情報が伝搬されるのを待ちます。

新しいクラスター・トピック定義がフル・リポジトリ・キュー・マネージャーによってクラスター内のすべてのキュー・マネージャーに伝搬されるのを待ちます。 **DISPLAY CLUSTER** コマンドを使用して各クラスター・メンバーのクラスター・トピックを表示し、QM2 から発信された定義があるかどうかを確認します。

トピック・ホスト・ルーティング型の場合は、QM2 上の新しいトピック・ホストがすべてのサブスクリプションを認識するのを待ちます。QM2 が認識しているプロキシ・サブスクリプションと、QM1 が認識しているプロキシ・サブスクリプションを比較します。キュー・マネージャーのプロキシ・サブスクリプションを表示する 1 つの方法として、以下のように **runmqsc** コマンドを発行する方法があります。

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

3. トピック・ホスト・ルーティング型の場合は、QM2 上のトピック・ホストを PUB(ENABLED) と再定義してから、QM1 上のトピック・ホストを PUB(DISABLED) として再定義します。

これで、QM2 上の新規トピック・ホストが他のキュー・マネージャーのすべてのサブスクリプションを認識し、トピック・ホストがルーティング・パブリケーションを開始できるようになりました。

PUB(DISABLED) 設定を使用して QM1 のメッセージ・トラフィックを静止し、クラスター・トピック定義を削除するときに QM1 を通過するパブリケーションがない状態にしてください。

4. QM1 のクラスター・トピック定義を削除します。

キュー・マネージャーが使用可能である場合は、QM1 から定義を削除するだけです。使用不可である場合は、両方の定義が存在する状態で QM1 が再始動するまで実行するか、または強制的に削除されるまで実行する必要があります。

QM1 が長期にわたって使用不可のままである場合で、その間に QM2 のクラスター・トピック定義を変更する必要がある場合は、QM2 定義は QM1 定義より新しいので、通常は優先されます。

この間に、QM1 と QM2 の定義の間で差異がある場合は、両方のキュー・マネージャーのエラー・ログに、クラスター・トピック定義が競合していることを警告するエラーが書き込まれます。

例えばハードウェアが故障した結果予期せぬ使用廃止が発生したために、QM1 が決してクラスターに戻されない場合、最後の手段として、**RESET CLUSTER** コマンドを使用してキュー・マネージャーを強制的に排除することができます。**RESET CLUSTER** コマンドを実行すると、自動的に、ターゲット・キュー・マネージャーでホストされているすべてのトピック・オブジェクトが削除されます。

## 関連概念

### パブリケーションとサブスクリプションの有効範囲の結合

IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

### パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

## 関連タスク

### パブリッシュ/サブスクライブ・クラスターの構成

キュー・マネージャーでトピックを定義します。トピックをクラスター・トピックにするには、**CLUSTER** プロパティを設定します。このトピックのパブリケーションとサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティを設定します。

### トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。この方式を利用して、アベイラビリティとワークロード・バランスを向上させることができます。同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

### パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続

階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

### パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断

パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

## トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。この方式を利用して、アベイラビリティとワークロード・バランスを向上させることができます。同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

## 始める前に

複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することが機能上役立つのは、トピック・ホスト・ルーティング型クラスターの場合に限られます。直接ルーティング型クラスター

ーでは、一致する複数のトピックを定義しても動作に変化はありません。この作業は、トピック・ホスト・ルーティング型クラスターのみが適用対象です。

この作業は、[同じ名前の複数のクラスター・トピック定義](#)という記事、特に以下のセクションを読んでいただくことを前提としています。

- [トピック・ホストで経路指定されるクラスター内の複数のクラスター・トピック定義](#)
- [PUB パラメーターの特別な処理](#)

## このタスクについて

キュー・マネージャーがルーティング型トピック・ホストにされた場合、まずキュー・マネージャーはクラスター内でサブスクライブされているすべての関連トピックの存在を検知する必要があります。トピック・ホストがさらに追加されるときにそれらのトピックにパブリケーションがパブリッシュされ、かつその新規ホストがクラスター内の他のキュー・マネージャー上のサブスクリプションの存在を検知する前にその新規ホストに送付されるパブリケーションがあるなら、そのホストはそのパブリケーションをそれらのサブスクリプションに転送しません。この結果、サブスクリプションでパブリケーションの欠落が生じます。

パブリケーションは、クラスター・トピック・オブジェクトの **PUB** パラメーターを明示的に **DISABLED** に設定したトピック・ホスト・キュー・マネージャーを介してルーティングされないため、この設定を使用して、トピック・ホストを追加するプロセス中にサブスクリプションがパブリケーションを見逃さないようにすることができます。

**注:** キュー・マネージャーは **PUB (DISABLED)** として定義されているクラスター・トピックをホストしますが、そのキュー・マネージャーに接続されているパブリッシャーはメッセージをパブリッシュできません。また、そのキュー・マネージャー上の一致するサブスクリプションは、クラスター内の他のキュー・マネージャーにパブリッシュされたパブリケーションを受信しません。このため、サブスクリプションが存在し、パブリッシュ・アプリケーションが接続されているキュー・マネージャーでトピック・ホスト・ルーティング型トピックを定義する際には、注意深く検討する必要があります。

## 手順

1. 新規トピック・ホストを構成し、最初にその新規トピック・ホストを **PUB (DISABLED)** として定義します。

これにより、新規トピック・ホストはクラスター内のサブスクリプションを検知しますが、パブリケーションの送付を開始しません。

トピック・ホストの構成については、[363 ページの『パブリッシュ/サブスクライブ・クラスターの構成』](#)を参照してください。

2. 新規トピック・ホストがすべてのサブスクリプションを検知した時点を見極めます。

これを行うには、新規トピック・ホストで認識されているプロキシ・サブスクリプションと、既存のトピック・ホストで認識されているプロキシ・サブスクリプションとを比較します。プロキシ・サブスクリプションを表示する1つの方法は、次の **runmqsc** コマンドを発行することです。 **DISPLAY SUB(\*) SUBTYPE (PROXY)**

3. 新しいトピック・ホストを **PUB (ENABLED)** として再定義します。

新規トピック・ホストが他のキュー・マネージャー上のすべてのサブスクリプションを検知した後、トピックではパブリケーションの送付を開始することができます。

## 関連概念

### [パブリケーションとサブスクリプションの有効範囲の結合](#)

IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

### [パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合](#)

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

## 関連タスク

### パブリッシュ/サブスクライブ・クラスターの構成

キュー・マネージャーでトピックを定義します。トピックをクラスター・トピックにするには、**CLUSTER** プロパティを設定します。このトピックのパブリケーションとサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティを設定します。

### 別のキュー・マネージャーへのクラスター・トピック定義の移動

トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

### パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続

階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

### パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断

パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

## パブリケーションとサブスクリプションの有効範囲の結合

IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

パブリケーションは、パブリッシュ/サブスクライブ・トポロジー内で接続されているすべてのキュー・マネージャーに流れるようにすることも、ローカル・キュー・マネージャーのみに流れるようにすることもできます。プロキシ・サブスクリプションに関しても同様です。どのパブリケーションがサブスクリプションに一致するかは、それら 2 つのフローの組み合わせによって制御されます。

パブリケーションとサブスクリプションは両方とも、有効範囲を QMGR または ALL に設定できます。パブリッシャーとサブスクライバーが両方とも同じキュー・マネージャーに接続されている場合、有効範囲の設定が、サブスクライバーがパブリッシャーからどのパブリケーションを受け取るかに影響を及ぼすことはありません。

パブリッシャーとサブスクライバーが異なるキュー・マネージャーに接続されている場合、リモート・パブリケーションを受け取るには、両方の設定を ALL にする必要があります。

例えば、パブリッシャーが異なるキュー・マネージャーに接続されているとします。サブスクライバーがあらゆるパブリッシャーからパブリケーションを受け取るようにするには、サブスクリプションの有効範囲を ALL に設定します。その後、パブリッシャーごとに、そのパブリケーションの有効範囲をそのパブリッシャーにローカルのサブスクライバーに制限するかどうかを決定できます。

サブスクライバーが異なるキュー・マネージャーに接続されているとします。あるパブリッシャーからのパブリケーションをすべてのサブスクライバーに送信するようにする場合、パブリケーション有効範囲を ALL に設定します。サブスクライバーが同じキュー・マネージャーに接続されているパブリッシャーだけからパブリケーションを受け取るようにする場合、サブスクリプション有効範囲を QMGR に設定します。

### 例: フットボールの試合結果サービス

フットボール・リーグのメンバーのチームであるとして。各チームは、キュー・マネージャーをパブリッシュ/サブスクライブ・クラスター内の他のすべてのチームと接続させています。

チームは、トピック `Football/result/Home team name/Away team name` を使用して、ホーム・グラウンドで行われたすべてのゲーム結果を公開します。イタリック体のストリングはトピック名の変数で、パブリケーションは試合の結果です。

また、各クラブは、トピック・ストリング `Football/myteam/Home team name/Away team name` を使用して、そのクラブの結果のみをリパブリッシュします。

両方のトピックがクラスター全体にパブリッシュされます。

リーグは以下のサブスクリプションをセットアップして、どのチームのファンも3つの関心のある方法で試合結果をサブスクライブできるようにします。

SUBSCOPE(QMGR) を使用してクラスター・トピックをセットアップすることができます。トピック定義はクラスターの各メンバーに伝搬されますが、サブスクリプションの有効範囲はローカル・キュー・マネージャーのみです。そのため、各キュー・マネージャーのサブスクライバーは、同じサブスクリプションからのさまざまなパブリケーションを受け取ります。

### すべての試合結果を受け取る

```
DEFINE TOPIC(A) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(ALL)
```

### ホームの試合結果すべてを受け取る

```
DEFINE TOPIC(B) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(QMGR)
```

サブスクリプションの有効範囲は QMGR なので、ホーム・グラウンドでパブリッシュされた試合結果のみが一致します。

### 自分のチームの試合結果すべてを受け取る

```
DEFINE TOPIC(C) TOPICSTR('Football/myteam/') CLUSTER SUBSCOPE(QMGR)
```

サブスクリプションの有効範囲は QMGR なので、ローカルでリパブリッシュされた、ローカルのチームの結果のみが一致します。

## 関連概念

### パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

## 関連タスク

### パブリッシュ/サブスクライブ・クラスターの構成

キュー・マネージャーでトピックを定義します。トピックをクラスター・トピックにするには、**CLUSTER** プロパティを設定します。このトピックのパブリケーションとサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティを設定します。

### 別のキュー・マネージャーへのクラスター・トピック定義の移動

トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

### トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。この方式を利用して、アベイラビリティとワークロード・バランスを向上させることができます。同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

### パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続

階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

### パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断

パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

## 関連情報

### 分散パブリッシュ/サブスクライブのネットワーク

[パブリケーション有効範囲](#)

[サブスクリプション有効範囲](#)

## パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

**CLUSTER**、**PUBSCOPE**、および **SUBSCOPE** 属性のビルディング・ブロック、パブリッシュ/サブスクライブ・クラスター、およびパブリッシュ/サブスクライブ階層を使用して、さまざまなパブリッシュ/サブスクライブ・トピック・スペースを作成できます。

以下のシナリオでは、1つのキュー・マネージャーからパブリッシュ/サブスクライブ・クラスターに拡大する例に始まり、さまざまなパブリッシュ/サブスクライブ・トポロジーについて説明します。

### 関連概念

[パブリケーションとサブスクリプションの有効範囲の結合](#)

IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

### 関連タスク

[パブリッシュ/サブスクライブ・クラスターの構成](#)

キュー・マネージャーでトピックを定義します。トピックをクラスター・トピックにするには、**CLUSTER** プロパティを設定します。このトピックのパブリケーションとサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティを設定します。

[別のキュー・マネージャーへのクラスター・トピック定義の移動](#)

トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

[トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加](#)

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。この方式を利用して、アベイラビリティとワークロード・バランスを向上させることができます。同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

[パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続](#)

階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

[パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断](#)

パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

### 関連情報

[分散パブリッシュ/サブスクライブのネットワーク](#)

[トピック・スペース](#)

[クラスター・トピックの定義](#)

## パブリッシュ/サブスクライブ・クラスターでの単一のトピック・スペースの作成

複数のキュー・マネージャー上で実行されるように、パブリッシュ/サブスクライブ・システムを拡大します。各パブリッシャーおよびサブスクライバーに単一で同一のトピック・スペースを提供するために、パブリッシュ/サブスクライブ・クラスターを使用します。

### 始める前に

単一のバージョンのキュー・マネージャーにパブリッシュ/サブスクライブ・システムを実装しています。

SYSTEM.BASE.TOPIC の属性の継承に依存するのではなく、常に独自のルート・トピックを使用してトピック・スペースを作成してください。パブリッシュ/サブスクライブ・システムをクラスターに拡大する場合、クラスター・トピック・ホストで、ルート・トピックをクラスター・トピックとして定義でき、そうするとすべてのトピックがクラスター全体で共有されます。

## このタスクについて

より多くのパブリッシャーおよびサブスクライバーをサポートするためにシステムを拡大し、すべてのトピックをクラスター全体で可視にしようとしています。

## 手順

1. パブリッシュ/サブスクライブ・システムで使用するクラスターを作成します。  
従来のクラスターが既に存在する場合、パフォーマンス上の理由で、新しいパブリッシュ/サブスクライブ・システム用に新しいクラスターをセットアップの方が良いでしょう。両方のクラスターのクラスター・リポジトリに、同じサーバーを使用できます。
2. クラスター・トピック・ホストにするキュー・マネージャーを1つ(おそらく、リポジトリのいずれか)選択します。
3. パブリッシュ/サブスクライブ・クラスター全体で可視にするすべてのトピックが、管理トピック・オブジェクトに解決されるようにします。  
パブリッシュ/サブスクライブ・クラスターの名前を指定する **CLUSTER** 属性を設定します。

## 次のタスク

パブリッシャーおよびサブスクライバー・アプリケーションをクラスター内の任意のキュー・マネージャーに接続します。

**CLUSTER** 属性を持つ管理トピック・オブジェクトを作成します。これらのトピックもクラスター全体に伝搬されます。パブリッシャーおよびサブスクライバー・プログラムは、クラスター内の異なるキュー・マネージャーに接続しても動作が変わらないように、管理トピックを使用します。

すべてのキュー・マネージャーで **SYSTEM.BASE.TOPIC** をクラスター・トピックのように機能させる必要がある場合は、キュー・マネージャーごとに変更する必要があります。

## 関連タスク

既存の IBM WebSphere MQ 6 トピック・スペースへのバージョン 7 以降のキュー・マネージャーの追加  
既存の IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・システムを拡張して、バージョン 7 以降のキュー・マネージャーと同じトピック・スペースを共有して相互運用するようにします。

### 複数のクラスターのトピック・スペースの結合

複数のクラスターにまたがるトピック・スペースを作成します。1つのクラスターでトピックにパブリッシュし、別のクラスターでそれに対してサブスクライブします。

### 複数のクラスター内でのトピック・スペースの結合および分離

いくつかのトピック・スペースを特定のクラスターに分離して、他のトピック・スペースを結合し、接続されているすべてのクラスター内でそれらにアクセスできるようにします。

### 複数のクラスター内のトピック・スペースに対するパブリッシュおよびサブスクライブ

オーバーラップされたクラスターを使用して、複数のクラスター内のトピックに対するパブリッシュおよびサブスクライブを行います。クラスター内でトピック・スペースがオーバーラップしない限り、この手法を使用できます。

## 関連情報

分散パブリッシュ/サブスクライブのネットワーク

トピック・スペース

クラスター・トピックの定義

## 既存の IBM WebSphere MQ 6 トピック・スペースへのバージョン 7 以降のキュー・マネージャーの追加

既存の IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・システムを拡張して、バージョン 7 以降のキュー・マネージャーと同じトピック・スペースを共有して相互運用するようにします。

### 始める前に

IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・システムが既に存在しています。

新しいサーバーに IBM WebSphere MQ 7 以降をインストールしており、キュー・マネージャーを構成しました。

### このタスクについて

既存の IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・システムを拡張して、バージョン 7 以降のキュー・マネージャーと連携するようにしようとしています。

キュー・パブリッシュ/サブスクライブ・インターフェースを使用する IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・システムの開発を行うことにしました。IBM WebSphere MQ 7 以降の MQI を使用して、システムに拡張を追加しようとしています。現時点では、キュー・パブリッシュ/サブスクライブ・アプリケーションを作成し直すという計画はありません。

将来的には、IBM WebSphere MQ 6 キュー・マネージャーを IBM WebSphere MQ 7 以降にアップグレードすることを考えています。現時点では、IBM WebSphere MQ 7 以降のキュー・マネージャー上で既存のキュー・パブリッシュ/サブスクライブ・アプリケーションを引き続き実行するつもりです。

### 手順

1. 送信側と受信側のチャンネルを 1 セット作成し、IBM WebSphere MQ 7 以降のキュー・マネージャーと IBM WebSphere MQ 6 キュー・マネージャーの 1 つを双方向で接続します。
2. ターゲット・キュー・マネージャーの名前を指定して、2 つの伝送キューを作成します。何らかの理由で、伝送キュー名としてターゲット・キュー・マネージャーの名前を使用できない場合は、キュー・マネージャーの別名を使用します。
3. 送信側チャンネルをトリガーするように、伝送キューを構成します。
4. IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・システムがストリームを使用する場合は、[ストリームの追加](#)の説明に従って、IBM WebSphere MQ 7 以降のキュー・マネージャーにストリームを追加します。
5. IBM WebSphere MQ 7 以降のキュー・マネージャーの **PSMODE** が **ENABLE** に設定されていることを検査します。
6. その **PARENT** 属性を変更して、IBM WebSphere MQ 6 キュー・マネージャーのいずれかを参照するようにします。
7. キュー・マネージャー間の親子関係の状況が両方向でアクティブであることを検査します。

### 次のタスク

この作業を完了すると、IBM WebSphere MQ 6 と 7 以降のキュー・マネージャーの両方が同じトピック・スペースを共有します。例えば、以下の作業をすべて行えます。

- IBM WebSphere MQ 6 と 7 以降のキュー・マネージャーの間でパブリケーションとサブスクリプションを交換します。
- 既存の IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・プログラムを IBM WebSphere MQ 7 以降のキュー・マネージャーで実行する。
- IBM WebSphere MQ 6 または 7 以降のキュー・マネージャーのトピック・スペースを表示および変更します。
- IBM WebSphere MQ 7 以降のパブリッシュ/サブスクライブ・アプリケーションを作成し、IBM WebSphere MQ 7 以降のキュー・マネージャーで実行する。

- IBM WebSphere MQ 7 以降のアプリケーションで新しいパブリケーションおよびサブスクリプションを作成し、それらを IBM WebSphere MQ 6 アプリケーションと交換する。

## 関連タスク

### パブリッシュ/サブスクライブ・クラスターでの単一のトピック・スペースの作成

複数のキュー・マネージャー上で実行されるように、パブリッシュ/サブスクライブ・システムを拡大します。各パブリッシャーおよびサブスクライバーに単一で同一のトピック・スペースを提供するために、パブリッシュ/サブスクライブ・クラスターを使用します。

### 複数のクラスターのトピック・スペースの結合

複数のクラスターにまたがるトピック・スペースを作成します。1つのクラスターでトピックにパブリッシュし、別のクラスターでそれに対してサブスクライブします。

### 複数のクラスター内でのトピック・スペースの結合および分離

いくつかのトピック・スペースを特定のクラスターに分離して、他のトピック・スペースを結合し、接続されているすべてのクラスター内でそれらにアクセスできるようにします。

### 複数のクラスター内のトピック・スペースに対するパブリッシュおよびサブスクライブ

オーバーラップされたクラスターを使用して、複数のクラスター内のトピックに対するパブリッシュおよびサブスクライブを行います。クラスター内でトピック・スペースがオーバーラップしない限り、この手法を使用できます。

## 関連情報

### 分散パブリッシュ/サブスクライブのネットワーク

### トピック・スペース

### クラスター・トピックの定義

## 複数のクラスターのトピック・スペースの結合

複数のクラスターにまたがるトピック・スペースを作成します。1つのクラスターでトピックにパブリッシュし、別のクラスターでそれに対してサブスクライブします。

## 始める前に

この作業では、直接ルーティング型パブリッシュ/サブスクライブ・クラスターが既に存在しており、いくつかのクラスター・トピックをすべてのクラスターに伝搬させようとしていることを前提としています。

**注:** トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターの場合は、この作業を実行できません。

## このタスクについて

あるクラスターから別のクラスターへパブリケーションを伝搬するには、それらのクラスターをまとめて1つの階層に結合する必要があります。[375 ページの図 66](#) を参照してください。階層型の接続は、接続されているキュー・マネージャー間でサブスクリプションおよびパブリケーションを伝搬します。クラスターは各クラスター内でクラスター・トピックを伝搬しますが、クラスター間では伝搬しません。

これらの2つのメカニズムの組み合わせにより、すべてのクラスター間でクラスター・トピックが伝搬されます。クラスターごとに、クラスター・トピック定義を繰り返す必要があります。

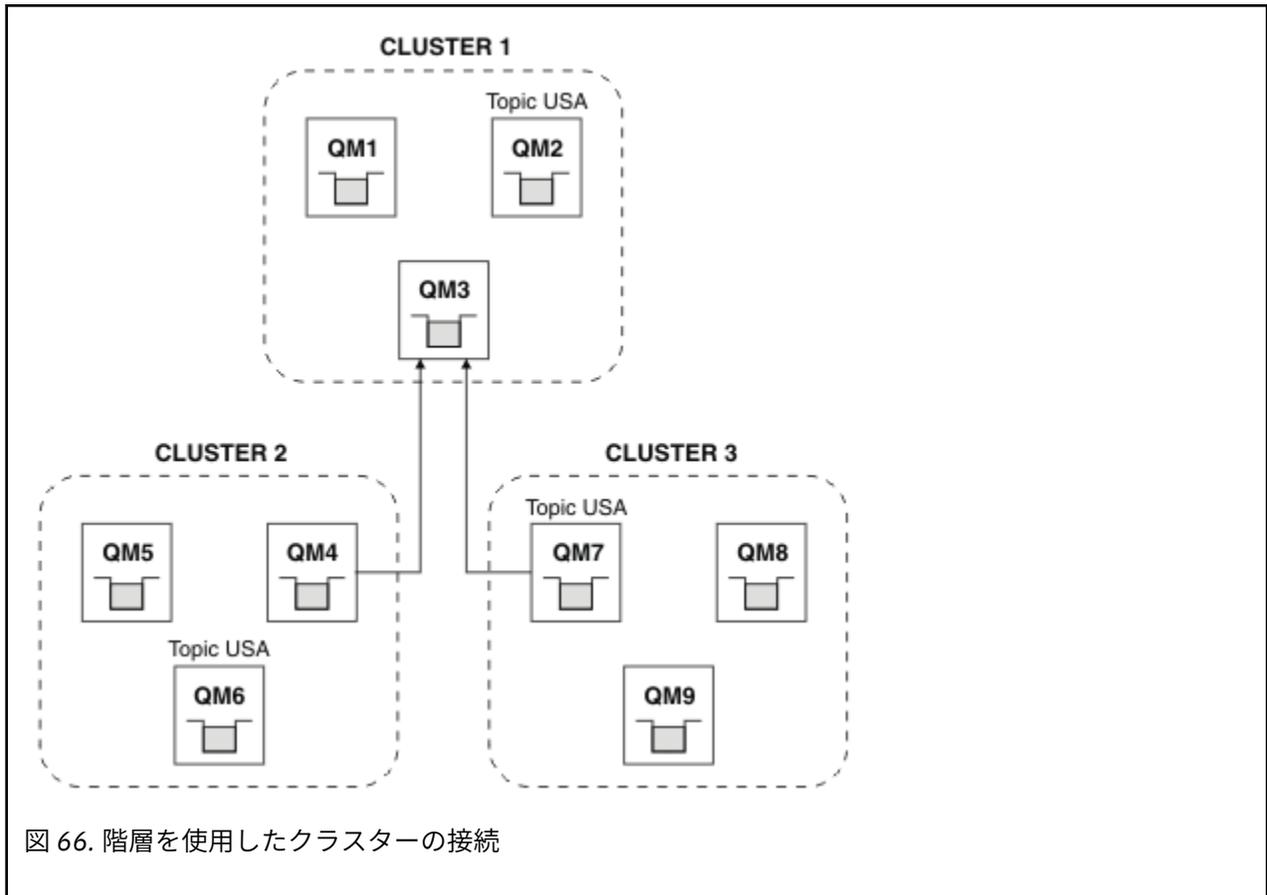


図 66. 階層を使用したクラスターの接続

以下のステップにより、クラスターが階層に接続されます。

## 手順

1. 二つの送信側と受信側チャンネルのセットを作成して、QM3 と QM4、QM3 と QM7 を両方の方向に接続します。従来の送信側と受信側のチャンネルおよび伝送キュー (クラスターではなく) を使用して、階層に接続しなければなりません。
2. ターゲット・キュー・マネージャーの名前を指定して、3 つの伝送キューを作成します。何らかの理由で、伝送キュー名としてターゲット・キュー・マネージャーの名前を使用できない場合は、キュー・マネージャーの別名を使用します。
3. 送信側チャンネルをトリガーするように、伝送キューを構成します。
4. QM3、QM4、および QM7 の **PSMODE** が **ENABLE** に設定されていることを確認します。
5. QM4 および QM7 の **PARENT** 属性を QM3 に変更します。
6. キュー・マネージャー間の親子関係の状況が両方向でアクティブであることを検査します。
7. 属性 **CLUSTER** (「CLUSTER 1」)、**CLUSTER** (「CLUSTER 2」)、および **CLUSTER** (「CLUSTER 3」) を使用してクラスター 1、2、および 3 内の 3 つのクラスター・トピック・ホスト・キュー・マネージャーのそれぞれに管理トピック USA を作成します。クラスター・トピック・ホストは、階層的に接続されたキュー・マネージャーである必要はありません。

## 次のタスク

これで、375 ページの図 66 でクラスター・トピック「USA」をパブリッシュまたはサブスクライブできます。パブリケーション/サブスクリプションは、3 つのすべてのクラスター内のパブリッシャーおよびサブスクライバーに流れます。

USA を他のクラスター内のクラスター・トピックとして作成しなかったとします。USA のみで定義されている場合、QM7、パブリケーションとサブスクリプションは USAQM7、QM8、QM9 と QM3 の間で交換されます。QM7、QM8、QM9 で実行されているパブリッシャーとサブスクライバーは管理トピックの属性 USA を

継承します。QM3 上のパブリッシャーおよびサブスクライバーは、QM3 上の SYSTEM.BASE.TOPIC の属性を継承します。

[376 ページの『複数のクラスター内でのトピック・スペースの結合および分離』](#)も参照してください。

## 関連タスク

[パブリッシュ/サブスクライブ・クラスターでの単一のトピック・スペースの作成](#)

複数のキュー・マネージャー上で実行されるように、パブリッシュ/サブスクライブ・システムを拡大します。各パブリッシャーおよびサブスクライバーに単一で同一のトピック・スペースを提供するために、パブリッシュ/サブスクライブ・クラスターを使用します。

[既存の IBM WebSphere MQ 6 トピック・スペースへのバージョン 7 以降のキュー・マネージャーの追加](#)  
[既存の IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・システムを拡張して、バージョン 7 以降のキュー・マネージャーと同じトピック・スペースを共有して相互運用するようにします。](#)

[複数のクラスター内でのトピック・スペースの結合および分離](#)

いくつかのトピック・スペースを特定のクラスターに分離して、他のトピック・スペースを結合し、接続されているすべてのクラスター内でそれらにアクセスできるようにします。

[複数のクラスター内のトピック・スペースに対するパブリッシュおよびサブスクライブ](#)

オーバーラップされたクラスターを使用して、複数のクラスター内のトピックに対するパブリッシュおよびサブスクライブを行います。クラスター内でトピック・スペースがオーバーラップしない限り、この手法を使用できます。

## 関連情報

[分散パブリッシュ/サブスクライブのネットワーク](#)

[トピック・スペース](#)

[クラスター・トピックの定義](#)

## 複数のクラスター内でのトピック・スペースの結合および分離

いくつかのトピック・スペースを特定のクラスターに分離して、他のトピック・スペースを結合し、接続されているすべてのクラスター内でそれらにアクセスできるようにします。

## 始める前に

[374 ページの『複数のクラスターのトピック・スペースの結合』](#)のトピックを調べてください。ブリッジとしてさらにキュー・マネージャーを追加しなくても、このトピックの手順で十分に必要を満たせる場合があります。

**注:** この作業は、直接ルーティング型パブリッシュ/サブスクライブ・クラスターを使用する場合にのみ実行できます。トピック・ホスト・ルーティング型クラスターを使用して実行することはできません。

## このタスクについて

すべてのクラスター間で共有されないクラスター・トピックを分離させることにより、[374 ページの『複数のクラスターのトピック・スペースの結合』](#)の [375 ページの図 66](#) で示されているトポロジーを向上させることができる可能性があります。どのクラスターにも含まれていないブリッジング・キュー・マネージャーを作成することにより、クラスターを分離します ([377 ページの図 67](#) を参照)。ブリッジング・キュー・マネージャーを使用して、どのパブリケーションおよびサブスクリプションが、あるクラスターから別のクラスターへ流れるようにするかをフィルター処理します。

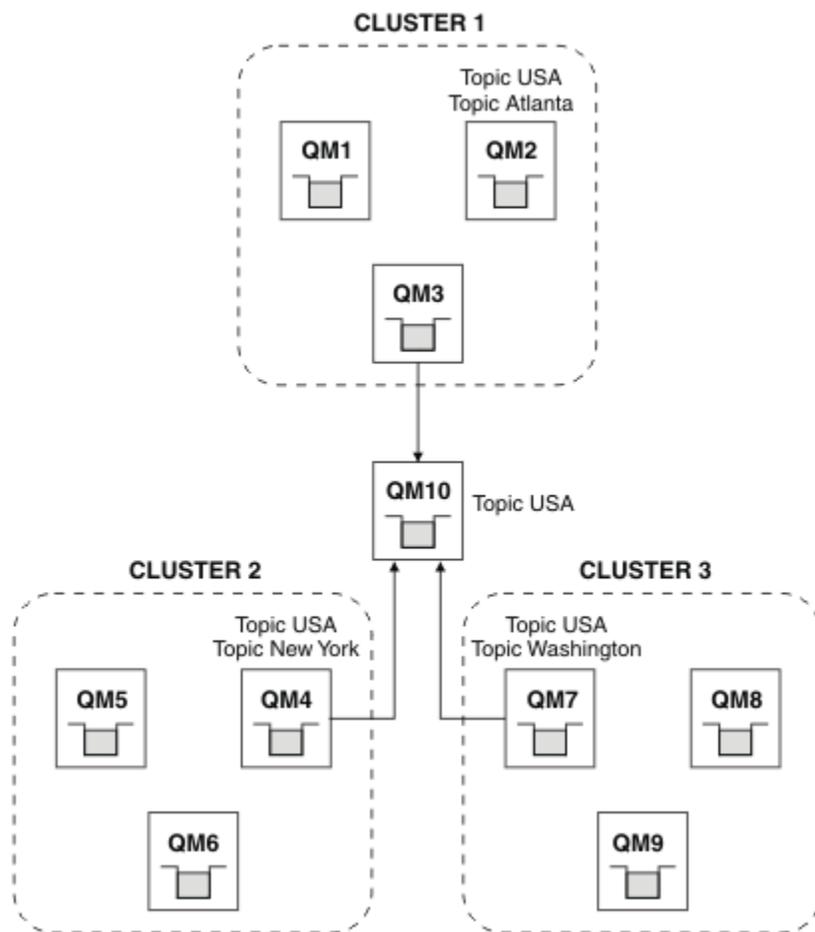


図 67.ブリッジされるクラスター

ブリッジを使用すると、ブリッジを超えて他のクラスターで公開しないクラスター・トピックを分離できます。において 377 ページの図 67、USA は全クラスターで共有されるクラスタートピック、Atlanta、New York および Washington それぞれ 1 つのクラスターでのみ共有されるクラスタートピックである。

以下の手順を使用して、構成をモデル化してみてください。

## 手順

- すべてのキュー・マネージャーで **SUBSCOPE** (QMGR) および **PUBSCOPE** (QMGR) を持つようにすべての SYSTEM.BASE.TOPIC トピック・オブジェクトを変更します。  
クラスター・トピックのルート・トピックに **SUBSCOPE** (ALL) および **PUBSCOPE** (ALL) を明示的に設定しない限り、トピック (クラスター・トピックも含む) は他のキュー・マネージャーに伝搬されません。
- 各クラスターで共有する 3 つのクラスター・トピック・ホスト・キュー・マネージャーのトピックを、属性 **CLUSTER** (*clustername*)、**SUBSCOPE** (ALL)、および **PUBSCOPE** (ALL) を使用して定義します。  
いくつかのクラスター・トピックをすべてのクラスター間で共有するには、それぞれのクラスターで同じトピックを定義します。クラスター属性として、各クラスターのクラスター名を使用します。
- すべてのクラスター間で共有するクラスター・トピックについて、属性 **SUBSCOPE** (ALL) および **PUBSCOPE** (ALL) を使用して、ブリッジ・キュー・マネージャー (QM10) でトピックを再定義します。

## 例

377 ページの図 67 の例では、USA から継承するトピックのみが、三つのクラスターの間で伝搬されます。

## 次のタスク

**SUBSCOPE** (ALL) および **PUBSCOPE** (ALL) を使用してブリッジ・キュー・マネージャーで定義されたトピックのサブスクリプションがクラスター間で伝搬されます。

属性 **CLUSTER** (*clustername*)、**SUBSCOPE** (ALL) および **PUBSCOPE** (ALL) を使用して各クラスター内で定義されたトピックのサブスクリプションが各クラスター内で伝搬されます。

他のサブスクリプションはすべて、キュー・マネージャーにとってローカルです。

### 関連タスク

[パブリッシュ/サブスクライブ・クラスターでの単一のトピック・スペースの作成](#)

複数のキュー・マネージャー上で実行されるように、パブリッシュ/サブスクライブ・システムを拡大します。各パブリッシャーおよびサブスクライバーに単一で同一のトピック・スペースを提供するために、パブリッシュ/サブスクライブ・クラスターを使用します。

[既存の IBM WebSphere MQ 6 トピック・スペースへのバージョン 7 以降のキュー・マネージャーの追加](#)

既存の IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・システムを拡張して、バージョン 7 以降のキュー・マネージャーと同じトピック・スペースを共有して相互運用するようにします。

[複数のクラスターのトピック・スペースの結合](#)

複数のクラスターにまたがるトピック・スペースを作成します。1つのクラスターでトピックにパブリッシュし、別のクラスターでそれに対してサブスクライブします。

[複数のクラスター内のトピック・スペースに対するパブリッシュおよびサブスクライブ](#)

オーバーラップされたクラスターを使用して、複数のクラスター内のトピックに対するパブリッシュおよびサブスクライブを行います。クラスター内でトピック・スペースがオーバーラップしない限り、この手法を使用できます。

### 関連情報

[分散パブリッシュ/サブスクライブのネットワーク](#)

[トピック・スペース](#)

[クラスター・トピックの定義](#)

[パブリケーション有効範囲](#)

[サブスクリプション有効範囲](#)

### 複数のクラスター内のトピック・スペースに対するパブリッシュおよびサブスクライブ

オーバーラップされたクラスターを使用して、複数のクラスター内のトピックに対するパブリッシュおよびサブスクライブを行います。クラスター内でトピック・スペースがオーバーラップしない限り、この手法を使用できます。

### 始める前に

従来のクラスターを複数作成して、クラスター間の交差部分にいくつかのキュー・マネージャーを配置します。

### このタスクについて

さまざまな理由で、クラスターをオーバーラップさせることを選択する場合があります。

1. 高可用性サーバーまたはキュー・マネージャーの数が限られています。クラスター・リポジトリおよびクラスター・トピック・ホストはすべてそれらにデプロイすることになっています。
2. ゲートウェイ・キュー・マネージャーを使用して接続される従来のキュー・マネージャー・クラスターが既に存在します。パブリッシュ/サブスクライブ・アプリケーションを同じクラスター・トポロジーにデプロイする予定です。
3. 自己完結型のパブリッシュ/サブスクライブ・アプリケーションがいくつか存在します。パフォーマンス上の理由で、パブリッシュ/サブスクライブ・クラスターを小さいままにし、従来のクラスターから分離させる方が適切です。それらのアプリケーションを別のクラスターにデプロイすることになっています。ただし、モニタリング・アプリケーションのライセンス交付を受けたコピーが1つだけであるため、1つのキュー・マネージャー上ですべてのパブリッシュ/サブスクライブ・アプリケーションをモニ

ターする必要があります。このキュー・マネージャーは、すべてのクラスター内のクラスター・トピックに対するパブリケーションにアクセスできなければなりません。

オーバーラップしないトピック・スペースでトピックが定義されるようにすることにより、オーバーラップしているパブリッシュ/サブスクライブ・クラスターにそのトピックをデプロイすることができます (379 ページの図 68 を参照)。トピック・スペースがオーバーラップしている場合、オーバーラップするクラスターにデプロイすると問題が発生します。

パブリッシュ/サブスクライブ・クラスターがオーバーラップするため、オーバーラップ内のキュー・マネージャーを使用して、任意のトピック・スペースにパブリッシュおよびサブスクライブすることができます。

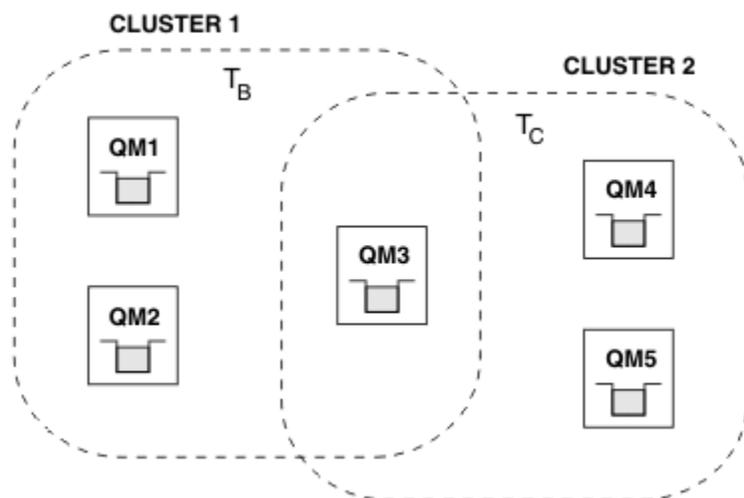


図 68. オーバーラップするクラスター、オーバーラップしないトピック・スペース

## 手順

トピック・スペースがオーバーラップしないようにする方法を作成します。

例えば、トピック・スペースごとに固有のルート・トピックを定義します。そのルート・トピックをクラスター・トピックにします。

a) DEFINE TOPIC(B) TOPICSTR('B') CLUSTER('CLUSTER 1') ...

b) DEFINE TOPIC(C) TOPICSTR('C') CLUSTER('CLUSTER 2') ...

## 例

379 ページの図 68 では、QM3 に接続されたパブリッシャーおよびサブスクライバーは、 $T_B$  または  $T_C$  にパブリッシュまたはサブスクライブできます

## 次のタスク

両方のクラスター内のトピックを使用するパブリッシャーおよびサブスクライバーを、オーバーラップ内のキュー・マネージャーに接続します。

特定のクラスター内のトピックのみを使用する必要があるパブリッシャーおよびサブスクライバーを、オーバーラップ内にはないキュー・マネージャーに接続します。

## 関連タスク

パブリッシュ/サブスクライブ・クラスターでの単一のトピック・スペースの作成

複数のキュー・マネージャー上で実行されるように、パブリッシュ/サブスクライブ・システムを拡大します。各パブリッシャーおよびサブスクライバーに単一で同一のトピック・スペースを提供するために、パブリッシュ/サブスクライブ・クラスターを使用します。

既存の IBM WebSphere MQ 6 トピック・スペースへのバージョン 7 以降のキュー・マネージャーの追加

既存の IBM WebSphere MQ 6 パブリッシュ/サブスクライブ・システムを拡張して、バージョン 7 以降のキュー・マネージャーと同じトピック・スペースを共有して相互運用するようにします。

#### 複数のクラスターのトピック・スペースの結合

複数のクラスターにまたがるトピック・スペースを作成します。1つのクラスターでトピックにパブリッシュし、別のクラスターでそれに対してサブスクライブします。

#### 複数のクラスター内でのトピック・スペースの結合および分離

いくつかのトピック・スペースを特定のクラスターに分離して、他のトピック・スペースを結合し、接続されているすべてのクラスター内でそれらにアクセスできるようにします。

### 関連情報

[分散パブリッシュ/サブスクライブのネットワーク](#)

[トピック・スペース](#)

[クラスター・トピックの定義](#)

## パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続

階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

### 始める前に

1. パブリッシュ/サブスクライブ階層内のキュー・マネージャーには、固有のキュー・マネージャー名が必要です。
2. パブリッシュ/サブスクライブ階層は、"キュー型パブリッシュ/サブスクライブ"のキュー・マネージャー機能に依存します。親と子のキュー・マネージャー両方でこれが有効になっている必要があります。キューに入れられたパブリッシュ/サブスクライブの開始を参照してください。
3. パブリッシュ/サブスクライブ関係は、キュー・マネージャーの送信側チャンネルと受信側チャンネルに依存します。チャンネルを確立する方法は2つあります。
  - 親および子キュー・マネージャーの両方を IBM MQ クラスターに追加します。 [クラスターへのキュー・マネージャーの追加](#)を参照してください。
  - 送信側/受信側チャンネルのペアを、子キュー・マネージャーから親へ、および親キュー・マネージャーから子へ確立します。各チャンネルは、ターゲット・キュー・マネージャーと同じ名前の伝送キューか、ターゲット・キュー・マネージャーと同じ名前のキュー・マネージャー別名のいずれかを使用する必要があります。Point-to-Point チャンネル接続を確立する方法の詳細については、[IBM MQ 分散キューイング技法](#)を参照してください。

チャンネル構成の各タイプにわたる階層を構成する例については、以下の一連のパブリッシュ/サブスクライブ階層シナリオを参照してください。

- [シナリオ 1: キュー・マネージャー別名を使用した Point-to-Point チャンネルの使用](#)
- [シナリオ 2: 伝送キューとリモート・キュー・マネージャーに同じ名前を使用した Point-to-Point チャンネルの使用](#)
- [シナリオ 3: クラスター・チャンネルを使用したキュー・マネージャーの追加](#)

### このタスクについて

ALTER QMGR PARENT (PARENT\_NAME) runmqsc コマンドを使用して、子を親に接続します。この構成は子キュー・マネージャー上で実行されます。PARENT\_NAME は親キュー・マネージャーの名前です。

### 手順

```
ALTER QMGR PARENT(PARENT_NAME)
```

## 例

1つ目の例は、キュー・マネージャー QM2 を QM1 の子として接続し、QM2 に照会して正常に **STATUS** が ACTIVE である子になったことを確認する方法を示しています。

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
  1 : alter qmgr parent(QM1)
AMQ8005: IBM MQ queue manager changed.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
      QMNAME(QM2)                TYPE(LOCAL)
      STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
      QMNAME(QM1)                TYPE(PARENT)
      STATUS(ACTIVE)
```

次の例は、QM1 にその接続について照会した結果を示しています。

```
C:\Documents and Settings\Admin>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
      QMNAME(QM1)                TYPE(LOCAL)
      STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
      QMNAME(QM2)                TYPE(CHILD)
      STATUS(ACTIVE)
```

**STATUS** が ACTIVE と表示されない場合は、子と親の間のチャンネルが正しく構成され実行されているか確認します。両方のキュー・マネージャーのエラー・ログで、考えられるエラーを確認します。

## 次のタスク

1つのキュー・マネージャーでパブリッシャーとサブスクライバーが使用するトピックは、デフォルトで階層内の他のキュー・マネージャー上のパブリッシャーとサブスクライバーによって共有されます。

**SUBSCOPE** および **PUBSCOPE** トピック・プロパティを使用して管理対象のトピックを構成し、共有レベルを制御することができます。[分散パブリッシュ/サブスクライブ・ネットワークの構成](#)を参照してください。

### 関連概念

[パブリケーションとサブスクリプションの有効範囲の結合](#)

IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

[パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合](#)

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

### 関連タスク

[パブリッシュ/サブスクライブ・クラスターの構成](#)

キュー・マネージャーでトピックを定義します。トピックをクラスター・トピックにするには、**CLUSTER** プロパティを設定します。このトピックのパブリケーションとサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティを設定します。

[別のキュー・マネージャーへのクラスター・トピック定義の移動](#)

トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

## トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。この方式を利用して、アベイラビリティとワークロード・バランスを向上させることができます。同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

## パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断

パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

### 関連情報

[ストリームおよびトピック](#)

[DISPLAY PUBSUB](#)

[パブリッシュ/サブスクライブ・メッセージング](#)

## パブリッシュ/サブスクライブ階層からのキュー・マネージャーの切断

パブリッシュ/サブスクライブ階層内の親キュー・マネージャーから子キュー・マネージャーを切断します。

### このタスクについて

**ALTER QMGR** コマンドを使用して、キュー・マネージャーをブローカー階層から切断します。キュー・マネージャーは、いつでも任意の順序で切断することができます。

親の更新に対応する要求は、キュー・マネージャー間の接続が稼働中に送信されます。

### 手順

```
ALTER QMGR PARENT( '')
```

### 例

```
C:\Documents and Settings\Admin>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
  1 : alter qmgr parent('')
AMQ8005: IBM MQ queue manager changed.
  2 : display pubsub type(child)
AMQ8147: IBM MQ object not found.
display pubsub type(parent)
  3 : display pubsub type(parent)
AMQ8147: IBM MQ object not found.
```

### 次のタスク

不要になったすべてのストリーム、キュー、および手動で定義したチャンネルをすべて削除できます。

### 関連概念

[パブリケーションとサブスクリプションの有効範囲の結合](#)

IBM WebSphere MQ 7.0 以降では、パブリケーションおよびサブスクリプションの有効範囲は独立して機能し、キュー・マネージャー間のパブリケーションのフローを決定します。

[パブリッシュ/サブスクライブ・ネットワークでのトピック・スペースの結合](#)

キュー・マネージャーのトピック・スペースを、パブリッシュ/サブスクライブ・クラスターまたは階層内の他のキュー・マネージャーと結合します。パブリッシュ/サブスクライブ・クラスターを結合し、パブリッシュ/サブスクライブ・クラスターと階層を結合します。

### 関連タスク

[パブリッシュ/サブスクライブ・クラスターの構成](#)

キュー・マネージャーでトピックを定義します。トピックをクラスター・トピックにするには、**CLUSTER** プロパティを設定します。このトピックのパブリケーションとサブスクリプションに使用するルーティングを選択するには、**CLROUTE** プロパティを設定します。

### 別のキュー・マネージャーへのクラスター・トピック定義の移動

トピック・ホスト・ルーティング型または直接ルーティング型のいずれかのクラスターについて、キュー・マネージャーを使用廃止するときに、またはクラスター・キュー・マネージャーで障害が発生したか長時間にわたって使用不可であることが理由で、クラスター・トピック定義を移動する必要がある場合があります。

### トピック・ホスト・ルーティング型クラスターへのさらなるトピック・ホストの追加

トピック・ホスト・ルーティング型パブリッシュ/サブスクライブ・クラスターでは、複数のキュー・マネージャーで同じクラスター・トピック・オブジェクトを定義することにより、それらの複数のキュー・マネージャーを使用してパブリケーションをサブスクリプションに送付することが可能です。この方式を利用して、アベイラビリティとワークロード・バランスを向上させることができます。同じクラスター・トピック・オブジェクトにトピック・ホストを追加する場合は、**PUB** パラメーターを使用して、新しいトピック・ホストを介してパブリケーションのルーティングを開始するタイミングを制御できます。

### パブリッシュ/サブスクライブ階層へのキュー・マネージャーの接続

階層内の親キュー・マネージャーに子キュー・マネージャーを接続します。子キュー・マネージャーが既に別の階層またはクラスターのメンバーである場合は、この接続を行うことによって両方の階層が結合してまとめられるか、クラスターが階層に結合します。

## ULW 複数のインストールの構成

同一のシステムで複数のインストールを使用している場合は、それらのインストールとキュー・マネージャーを構成する必要があります。

### このタスクについて

この情報は UNIX, Linux, and Windows に適用されます。

### 手順

- 以下のリンクの情報を使用してインストールを構成してください。
  - [393 ページの『プライマリー・インストールの変更』](#)
  - [395 ページの『キュー・マネージャーとインストールの関連付け』](#)
  - [383 ページの『複数のインストール環境でのアプリケーションの接続』](#)

## ULW 複数のインストール環境でのアプリケーションの接続

UNIX, Linux, and Windows システムに IBM WebSphere MQ 7.1 以降のライブラリーがロードされると、それ以上操作を行わなくても、IBM MQ によって自動的に適切なライブラリーが使用されます。IBM MQ は、アプリケーションが接続されるキュー・マネージャーと関連付けられているインストール済み環境からのライブラリーを使用します。

以下の概念は、アプリケーションを IBM MQ に接続する方法を説明する際に使用されます。

### リンク

アプリケーションは、コンパイル時に IBM MQ ライブラリーにリンクされ、機能エクスポートを取得します。このエクスポートはアプリケーションの実行時に読み込まれます。

### ロード中

アプリケーションの実行時に、IBM MQ ライブラリーの場所が探索され、ロードされます。ライブラリーを見つけるための具体的なメカニズムは、オペレーティング・システムと、アプリケーションの構築方法によって異なります。複数インストール環境でのライブラリーの場所探索方法とロード方法の詳細については、[385 ページの『IBM MQ ライブラリーのロード』](#)を参照してください。

### 接続中

アプリケーションは、MQCONN または MQCONNX 呼び出しなどを使用して、実行中のキュー・マネージャーに接続する際に、読み込まれた IBM MQ ライブラリーを使用して接続を行います。

サーバー・アプリケーションをキュー・マネージャーに接続する場合は、そのキュー・マネージャーに関連付けられているインストール済み環境から読み込まれたライブラリーを使用する必要があります。シス

テム上に複数のインストール済み環境がある場合、この制限により、オペレーティング・システムが、ロードする IBM MQ ライブラリーを見つけるために使用するメカニズムを選択する際に、以下のような新たな課題が生じます。

- **setmqm** コマンドを使用して、キュー・マネージャーに関連付けられているインストール済み環境を変更する場合、読み込む必要があるライブラリーも変わる。
- アプリケーションを、異なるインストール済み環境が所有している複数のキュー・マネージャーに接続する場合、複数のライブラリー・セットを読み込む必要がある。

しかし、IBM WebSphere MQ 7.1 以降のライブラリーの場所が探索されてロードされると、それ以上操作を行わなくても、IBM MQ によって適切なライブラリーがロードされ、使用されます。アプリケーションがキュー・マネージャーに接続されると、IBM MQ は、そのキュー・マネージャーに関連付けられているインストール済み環境からライブラリーを読み込みます。

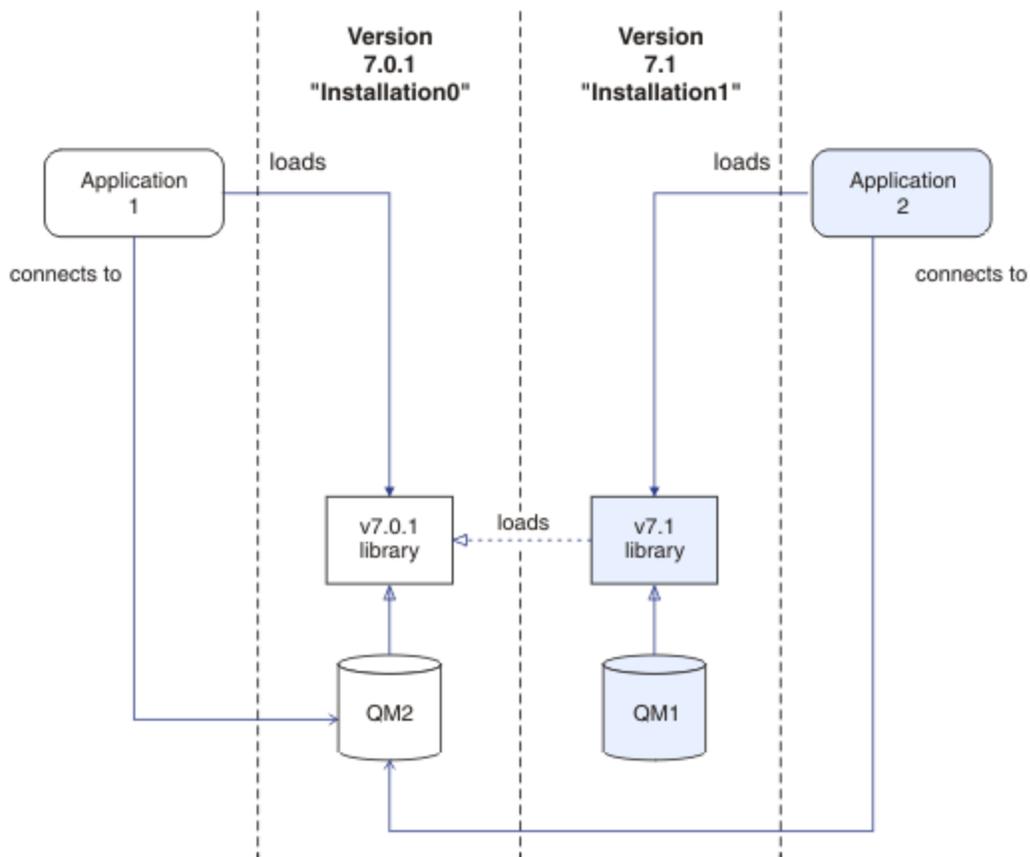


図 69. 複数のインストール環境でのアプリケーションの接続

例えば、384 ページの図 69 は、IBM WebSphere MQ 7.0.1 インストール済み環境 (Installation0) と IBM WebSphere MQ 7.1 インストール済み環境 (Installation1) を含む複数インストール環境を示しています。これらのインストール済み環境には 2 つのアプリケーションが接続されていますが、異なるライブラリー・バージョンをロードします。

Application 1 は、IBM WebSphere MQ 7.0.1 ライブラリーを直接ロードします。application 1 が QM2 に接続すると、IBM WebSphere MQ 7.0.1 ライブラリーが使用されます。application 1 が QM1 に接続しようとした場合、または QM2 が Installation1 に関連付けられている場合、application 1 は **2059 (080B) (RC2059): MQRC\_Q\_MGR\_NOT\_AVAILABLE** エラーで失敗します。アプリケーションの操作の失敗は、IBM WebSphere MQ 7.0.1 ライブラリーが他のライブラリー・バージョンをロードできないことが原因で発生します。つまり、IBM WebSphere MQ 7.0.1 ライブラリーが直接ロードされる場合、それより新しいバージョンの IBM MQ のインストール済み環境に関連付けられているキュー・マネージャーを使用することはできません。

Application 2は、IBM WebSphere MQ 7.1 ライブラリーを直接ロードします。application 2がQM2に接続すると、IBM WebSphere MQ 7.1 ライブラリーはIBM WebSphere MQ 7.0.1 ライブラリーをロードして使用します。application 2がQM1に接続する場合、またはQM2がInstallation1に関連付けられている場合は、IBM WebSphere MQ 7.1 ライブラリーがロードされ、アプリケーションは予期したとおりに動作します。

複数のインストール済み環境でのマイグレーション・シナリオおよびアプリケーションの接続について詳しくは、[UNIX, Linux, and Windows での複数インストール済み環境のキュー・マネージャーの共存](#)を参照してください。

IBM WebSphere MQ 7.1 ライブラリーのロード方法の詳細については、[385 ページの『IBM MQ ライブラリーのロード』](#)を参照してください。

## サポートおよび制約事項

以下の IBM WebSphere MQ 7.1 以降のライブラリーのいずれかが検出されてロードされると、IBM MQ は自動的に適切なライブラリーをロードして使用することができます。

- C サーバー・ライブラリー
- C++ サーバー・ライブラリー
- XA サーバー・ライブラリー
- COBOL サーバー・ライブラリー
- COM+ サーバー・ライブラリー
- 非管理対象モードの .NET

IBM MQ は、バインディング・モードの Java および JMS アプリケーションに適したライブラリーも自動的にロードし、使用します。

複数のインストール済み環境を使用するアプリケーションに関する制約事項がいくつかあります。詳しくは、[389 ページの『複数のインストールを使用するアプリケーションの制限』](#)を参照してください。

### 関連概念

[389 ページの『複数のインストールを使用するアプリケーションの制限』](#)

CICS サーバー・ライブラリー、ファスト・パス接続、メッセージ・ハンドル、および出口を複数インストール環境で使用する場合、いくつかの制限があります。

[385 ページの『IBM MQ ライブラリーのロード』](#)

IBM MQ ライブラリーのロード方法を決定する際には、使用環境、既存のアプリケーションを変更できるかどうか、プライマリー・インストールを行うかどうか、IBM MQ のインストール場所、および IBM MQ の場所を変更する可能性があるかどうかなど、いくつかの要素を考慮する必要があります。

### 関連タスク

[393 ページの『プライマリー・インストールの変更』](#)

**setmqinst** コマンドを使用すると、インストールをプライマリー・インストールとして設定する、あるいはその設定を解除することができます。

[395 ページの『キュー・マネージャーとインストールの関連付け』](#)

キュー・マネージャーを作成すると、そのキュー・マネージャーは、**crtmqm** コマンドを実行したインストールに自動的に関連付けられます。UNIX, Linux, and Windows では、**setmqm** コマンドを使用して、キュー・マネージャーに関連付けられているインストールを変更できます。

### 関連情報

[プライマリー・インストールの選択](#)

## **ULW** IBM MQ ライブラリーのロード

IBM MQ ライブラリーのロード方法を決定する際には、使用環境、既存のアプリケーションを変更できるかどうか、プライマリー・インストールを行うかどうか、IBM MQ のインストール場所、および IBM MQ の場所を変更する可能性があるかどうかなど、いくつかの要素を考慮する必要があります。

この情報は、IBM WebSphere MQ 7.1 以降のバージョンのライブラリーに適用されます。

IBM MQ ライブラリーの場所を探索しロードする方法は、ご使用のインストール済み環境に応じて異なります。

- UNIX and Linux システムでは、IBM WebSphere MQ 7.1 以降のバージョンのコピーをデフォルトの場所にインストールしている場合、既存のアプリケーションは引き続き、前のバージョンと同じように動作します。ただし、アプリケーションが /usr/lib でシンボリック・リンクを必要とする場合は、IBM WebSphere MQ 7.1 以降のバージョンのインストール済み環境をプライマリー・インストールとして選択するか、シンボリック・リンクを手動で作成する必要があります。
- IBM WebSphere MQ 7.1 以降のバージョンがデフォルト以外の場所にインストールされている (IBM WebSphere MQ 7.0.1 もインストールされている場合に該当する) ときには、正しいライブラリーがロードされるように、既存のアプリケーションを変更しなければならない場合があります。

IBM MQ ライブラリーの場所を探索しロードする方法は、既存のアプリケーションがライブラリーをロードするためにどのようにセットアップされたかによっても異なります。ライブラリーのロード方法の詳細については、388 ページの『オペレーティング・システムのライブラリー・ロード・メカニズム』を参照してください。

最善策として、オペレーティング・システムによってロードされる IBM MQ ライブラリーが、キュー・マネージャーと関連付けられたライブラリーとなるようにしてください。

IBM MQ ライブラリーをロードする方法は、プラットフォームによって異なり、それぞれ利点と欠点があります。

表 26. ライブラリーをロードするためのオプションの利点と欠点			
プラットフォーム	オプション	利点	欠点
  UNIX and Linux システム	アプリケーションの埋め込みランタイム検索パス (RPath) を設定または変更する。  このオプションでは、アプリケーションを再コンパイルしてリンクする必要があります。アプリケーションのコンパイルとリンクについて詳しくは、 <a href="#">プロシージャ型アプリケーションの構築</a> を参照してください。	<ul style="list-style-type: none"> <li>• 変更範囲が明確です。</li> </ul>	<ul style="list-style-type: none"> <li>• アプリケーションを再コンパイルしてリンクする必要があります。</li> <li>• IBM MQ の場所が変わった場合に、RPath を変更する必要があります。</li> </ul>

表 26. ライブラリーをロードするためのオプションの利点と欠点 (続き)

プラットフォーム	オプション	利点	欠点
UNIX and Linux システム	<p><b>-k</b> オプションまたは <b>-l</b> オプションを指定して、<a href="#">setmqenv</a> または <a href="#">crtmqenv</a> を使用し、<a href="#">LD_LIBRARY_PATH</a> 環境変数を設定します。(</p> <p><b>AIX</b> AIX では、この環境変数は <a href="#">LIBPATH</a>)</p>	<ul style="list-style-type: none"> <li>既存のアプリケーションを変更する必要がありません。</li> <li>アプリケーションの埋め込み RPath をオーバーライドします。</li> <li>IBM MQ の場所が変わった場合の変数の変更が容易です。</li> </ul>	<ul style="list-style-type: none"> <li><a href="#">setuid</a> アプリケーションおよび <a href="#">setgid</a> アプリケーション、または他の方法で構築されたアプリケーションでは、セキュリティ上の理由で、<a href="#">LD_LIBRARY_PATH</a> が無視される場合があります。</li> <li>環境固有であるため、アプリケーションを実行する環境ごとに設定する必要があります。</li> <li><a href="#">LD_LIBRARY_PATH</a> に依存する他のアプリケーションに影響する可能性があります。</li> <li><b>HP-UX</b> HP-UX: アプリケーションのコンパイル時に使用したオプションにより、<a href="#">LD_LIBRARY_PATH</a> が使用不可になっている場合があります。詳しくは、<a href="#">HP-UX 用のランタイム・リンクに関する考慮事項</a>を参照してください。</li> <li><b>Linux</b> Linux: アプリケーションの構築時に使用したコンパイラーにより、<a href="#">LD_LIBRARY_PATH</a> が使用不可になっている場合があります。詳しくは、<a href="#">Linux 用のランタイム・リンクに関する考慮事項</a>を参照してください。</li> </ul>
<p><b>Windows</b> Windows システム</p>	<p><a href="#">setmqenv</a> または <a href="#">crtmqenv</a> を使用して、<a href="#">PATH</a> 変数を設定する。</p>	<ul style="list-style-type: none"> <li>既存のアプリケーションを変更する必要がありません。</li> <li>IBM MQ の場所が変わった場合の変数の変更が容易です。</li> </ul>	<ul style="list-style-type: none"> <li>環境固有であるため、アプリケーションを実行する環境ごとに設定する必要があります。</li> <li>他のアプリケーションに影響する可能性があります。</li> </ul>

表 26. ライブラリーをロードするためのオプションの利点と欠点 (続き)

プラットフォーム	オプション	利点	欠点
<p><b>ULW</b></p> <p>UNIX, Linux, and Windows システム</p>	<p>プライマリー・インストールを IBM WebSphere MQ 7.1 以降のインストールに設定する。393 ページの『<a href="#">プライマリー・インストールの変更</a>』を参照してください。</p> <p>プライマリー・インストールについて詳しくは、<a href="#">プライマリー・インストールの選択</a>を参照してください。</p>	<ul style="list-style-type: none"> <li>既存のアプリケーションを変更する必要がありません。</li> <li>IBM MQ の場所が変わった場合のプライマリー・インストールの変更が容易です。</li> <li>前のバージョンの IBM MQ と同じような動作になります。</li> </ul>	<ul style="list-style-type: none"> <li>IBM WebSphere MQ 7.0.1 がインストールされている場合は、プライマリー・インストールを IBM WebSphere MQ 7.1 以降に設定することはできません。</li> <li>Linux</li> <li>UNIX</li> <li>UNIX and Linux: /usr/lib がデフォルトの検索パスにない場合は、機能しません。</li> </ul>

## HP-UX のライブラリーのロードに関する考慮事項

### HP-UX

前のバージョンの IBM MQ 製品資料で提供されたコンパイル・コマンドのサンプルには、64 ビット・アプリケーション用の `-W1, +noenvvar` リンク・オプションが組み込まれていました。このオプションにより、共有ライブラリーをロードするための `LD_LIBRARY_PATH` が使用不可になります。ご使用のアプリケーションで、`RPath` に指定されている場所以外から IBM MQ ライブラリーを読み込む場合は、アプリケーションを更新する必要があります。 `-W1, +noenvvar` リンク・オプションを指定せずに再コンパイルおよびリンクするか、または `chatr` コマンドを使用して、アプリケーションを更新することができます。

アプリケーションによるライブラリーの現在のロード方法を調べるには、388 ページの『[オペレーティング・システムのライブラリー・ロード・メカニズム](#)』を参照してください。

## Linux のライブラリーのロードに関する考慮事項

### Linux

バージョン 3.2.x など、一部のバージョンの `gcc` を使用してコンパイルされたアプリケーションには、`LD_LIBRARY_PATH` 環境変数を使用してオーバーライドできない `RPath` を埋め込むことができます。 `readelf -d applicationName` コマンドを使用して、アプリケーションが影響を受けるかどうかを判別できます。 `RPATH` シンボルは存在するが、`RUNPATH` シンボルが存在しない場合は、`RPath` をオーバーライドすることはできません。

## Solaris のライブラリーのロードに関する考慮事項

### Solaris

前のバージョンの IBM MQ 製品資料で提供されたコンパイル・コマンドのサンプルには、`-lmqmcs -lmqmzse` リンク・オプションが組み込まれていました。これらのライブラリーの適切なバージョンが、IBM MQ によって自動的にロードされるようになりました。IBM MQ がデフォルト以外の場所にインストールされている場合、またはシステム上に複数のインストール済み環境がある場合は、アプリケーションを更新する必要があります。アプリケーションを更新するには、`-lmqmcs -lmqmzse` リンク・オプションを指定せずに再コンパイルしてリンクします。

## オペレーティング・システムのライブラリー・ロード・メカニズム

Windows システムでは、ライブラリーを見つける際に以下のいくつかのディレクトリーが検索されます。

- アプリケーションの読み込み元のディレクトリー。

- 現行ディレクトリー。
- `PATH` 環境変数 (グローバル `PATH` 変数および現行ユーザーの `PATH` 変数の両方) のディレクトリー。

**Linux** **UNIX** UNIX and Linux システムでは、以下のように、ロードするライブラリーを見つけるために使用されてきた方法が多数あります。

- `LD` ライブラリー・パス 環境変数を使用する (AIX、および HP-UX 上の `SHLIB` パス では ライブラリー・パス も使用)。この変数が設定されている場合、その変数は、必要な IBM MQ ライブラリーを検索する一連のディレクトリーを定義します。これらのディレクトリーにライブラリーがある場合、他の方法を使用して検出されるどのライブラリーよりもそのライブラリーが優先されます。
- 埋め込み検索パス (RPath) を使用。アプリケーションには、IBM MQ ライブラリーを検索する一連のディレクトリーが含まれている場合があります。 `LD_LIBRARY_PATH` が設定されていない場合、または必要なライブラリーが変数を使用して検出されなかった場合は、ライブラリーを見つけるために RPath が検索されます。既存のアプリケーションで RPath を使用しているが、アプリケーションを再コンパイルしてリンクできない場合は、デフォルトの場所に IBM WebSphere MQ 7.1 をインストールするか、ライブラリーの検出に別の方法を使用する必要があります。
- デフォルトのライブラリー・パスを使用。 `LD_LIBRARY_PATH` 変数と RPath の場所を検索しても、IBM MQ ライブラリーが見つからなかった場合は、デフォルトのライブラリー・パスが検索されます。通常は、このパスに `/usr/lib` または `/usr/lib64` が含まれています。デフォルトのライブラリー・パスを検索してもライブラリーが見つからなかった場合は、依存関係の欠落により、アプリケーションは開始できません。

アプリケーションに埋め込み検索パスがあるかどうかについては、オペレーティング・システム・メカニズムを使用して調べることができます。以下に例を示します。

- **AIX** AIX: `dump`
- **HP-UX** HP-UX: `chatr`
- **Linux** Linux: `readelf`
- **Solaris** Solaris: `elfdump`

## 関連概念

[389 ページの『複数のインストールを使用するアプリケーションの制限』](#)

CICS サーバー・ライブラリー、ファスト・パス接続、メッセージ・ハンドル、および出口を複数インストール環境で使用する場合、いくつかの制限があります。

[383 ページの『複数のインストール環境でのアプリケーションの接続』](#)

UNIX, Linux, and Windows システムに IBM WebSphere MQ 7.1 以降のライブラリーがロードされると、それ以上操作を行わなくても、IBM MQ によって自動的に適切なライブラリーが使用されます。IBM MQ は、アプリケーションが接続されるキュー・マネージャーと関連付けられているインストール済み環境からのライブラリーを使用します。

## 関連タスク

[393 ページの『プライマリー・インストールの変更』](#)

`setmqinst` コマンドを使用すると、インストールをプライマリー・インストールとして設定する、あるいはその設定を解除することができます。

[395 ページの『キュー・マネージャーとインストールの関連付け』](#)

キュー・マネージャーを作成すると、そのキュー・マネージャーは、`crtmqm` コマンドを実行したインストールに自動的に関連付けられます。UNIX, Linux, and Windows では、`setmqm` コマンドを使用して、キュー・マネージャーに関連付けられているインストールを変更できます。

## 関連情報

[プライマリー・インストールの選択](#)

## **ULW** 複数のインストールを使用するアプリケーションの制限

CICS サーバー・ライブラリー、ファスト・パス接続、メッセージ・ハンドル、および出口を複数インストール環境で使用する場合、いくつかの制限があります。

## CICS サーバー・ライブラリー

CICS サーバー・ライブラリーを使用している場合、IBM MQ は正しいライブラリー・レベルを自動的に選択しません。アプリケーションが接続するキュー・マネージャーに対する適切なライブラリー・レベルを使用して、ユーザーがアプリケーションをコンパイルおよびリンクする必要があります。詳しくは、[TXSeries® Multiplatforms バージョン 5 で使用するためのライブラリーの作成](#)を参照してください。

## メッセージ・ハンドル

特殊値 MQHC\_UNASSOCIATED\_HCONN を使用するメッセージ・ハンドルは、プロセスでロードされる最初のインストールでの使用に制限されています。特定のインストールによってこのメッセージ・ハンドルが使用できない場合は、理由コード MQRC\_HMSG\_NOT\_AVAILABLE が返されます。

この制限は、メッセージ・プロパティーに影響を与えます。あるインストール上のキュー・マネージャーから、メッセージ・ハンドルを使用してメッセージ・プロパティーを取得し、それらを別のインストール上のキュー・マネージャーに書き込むことはできません。メッセージ・ハンドルについて詳しくは、[MQCRTMH - メッセージ・ハンドルの作成](#)を参照してください。

## 出口

複数インストール環境では、IBM WebSphere MQ 7.1 (以上) のインストールと連携するために、既存の出口を更新する必要があります。`crtmqcvx` コマンドを使用して生成されたデータ変換出口は、更新されたコマンドを使用して再生成する必要があります。

すべての出口は、MQIEP 構造体を使用して記述される必要があります、埋め込まれた RPATH を使用して IBM MQ ライブラリーを指定できず、IBM MQ ライブラリーにリンクすることはできません。詳しくは、[出口とインストール可能サービスの作成 \(UNIX, Linux, and Windows\)](#)を参照してください。

## ファスト・パス

複数のインストールが存在するサーバー上では、IBM WebSphere MQ 7.1 (以上) との接続にファスト・パス接続を使用するアプリケーションは、以下の規則を守る必要があります。

1. キュー・マネージャーは、アプリケーションが IBM MQ ランタイム・ライブラリーをロードしたときに使用したのと同じインストールに関連付けられている必要があります。アプリケーションは、別のインストールに関連付けられているキュー・マネージャーに対してファスト・パス接続を使用してはなりません。接続を試みるとエラーが発生し、理由コード MQRC\_INSTALLATION\_MISMATCH が出力されます。
2. アプリケーションが IBM MQ ランタイム・ライブラリーをロードしたときに使用したのと同じインストールに関連付けられているキュー・マネージャーに対して非ファスト・パスで接続すると、以下の条件が両方とも真である場合を除いて、このアプリケーションはファスト・パス接続できなくなります。
  - このアプリケーションが、同じインストールに関連付けられているキュー・マネージャーとの最初の接続に、ファスト・パス接続を使用した。
  - 環境変数 AMQ\_SINGLE\_INSTALLATION が設定されている。
3. IBM WebSphere MQ 7.1 (以上) のインストールに関連付けられているキュー・マネージャーと非ファスト・パス接続をする場合は、アプリケーションがファスト・パス接続できるかどうかに対する影響はありません。
4. IBM WebSphere MQ 7.0.1 インストール済み環境に関連付けられたキュー・マネージャーへの接続と、IBM WebSphere MQ 7.1 以降のインストール済み環境に関連付けられたキュー・マネージャーへのファスト・パス接続を組み合わせることはできません。

AMQ\_SINGLE\_INSTALLATION を設定すると、キュー・マネージャーとの任意の接続をファスト・パス接続にできます。それ以外は、以下のように、ほぼ同様の制限が適用されます。

- インストールは、IBM MQ ランタイム・ライブラリーをロードするために使用したのと同じでなければならない。
- 同一プロセス上のすべての接続は、同一インストールに対してでなければならない。別のインストールに関連付けられているキュー・マネージャーに接続しようとする、理由コード MQRC\_INSTALLATION\_MISMATCH で接続が失敗します。AMQ\_SINGLE\_INSTALLATION が設定されて

いと、この制限がファスト・パス接続だけでなく、すべての接続に適用されることに注意してください。

- 1つのキュー・マネージャーに対してのみ、ファスト・パス接続する。

## 関連情報

[MQCONN - キュー・マネージャーの接続 \(拡張\)](#)

[MQIEP 構造体](#)

[2583 \(0A17\) \(RC2583\): MQRC\\_INSTALLATION\\_MISMATCH](#)

[2587 \(0A1B\) \(RC2587\): MQRC\\_HMSG\\_NOT\\_AVAILABLE](#)

[2590 \(0A1E\) \(RC2590\): MQRC\\_FASTPATH\\_NOT\\_AVAILABLE](#)

## ULW

## 複数のインストール環境での .NET アプリケーションの接続

デフォルトでは、プライマリー・インストール済み環境の .NET アセンブリーがアプリケーションで使用されます。プライマリー・インストール済み環境がない場合、またはプライマリー・インストール済み環境のアセンブリーを使用しない場合は、アプリケーションの構成ファイル、または `DEVPATH` 環境変数を更新する必要があります。

システム上にプライマリー・インストール済み環境がある場合、そのインストール済み環境の .NET アセンブリーとポリシー・ファイルはグローバル・アセンブリー・キャッシュ (GAC) に登録されています。他のすべてのインストール済み環境の .NET アセンブリーは、各インストール済み環境のインストール・パスにあります。それらのアセンブリーは GAC には登録されていません。したがって、デフォルトでは、プライマリー・インストール済み環境の .NET アセンブリーを使用して、アプリケーションが実行されます。以下のいずれかの状況に当てはまる場合は、アプリケーション構成ファイルを更新する必要があります。

- プライマリー・インストール済み環境がない。
- アプリケーションでプライマリー・インストール済み環境のアセンブリーを使用しない。
- アプリケーションのコンパイル時に使用したバージョンよりも古いバージョンの IBM MQ が、プライマリー・インストール済み環境として使用されている。

アプリケーション構成ファイルの更新方法については、391 ページの『[アプリケーション構成ファイルを使用する .NET アプリケーションの接続](#)』を参照してください。

次の状況に当てはまる場合は、`DEVPATH` 環境変数を更新する必要があります。

- アプリケーションでプライマリー以外のインストール済み環境のアセンブリーを使用するが、プライマリー・インストールのバージョンがそのプライマリー以外のインストールのバージョンと同じである。

`DEVPATH` 変数の更新方法について詳しくは、392 ページの『[DEVPATH を使用する .NET アプリケーションの接続](#)』を参照してください。

## アプリケーション構成ファイルを使用する .NET アプリケーションの接続

アプリケーション構成ファイル内に、さまざまなタグを設定して、プライマリー・インストール済み環境以外のアセンブリーを使用するようにアプリケーションをリダイレクトする必要があります。

以下の表には、特定のアセンブリーを使用して .NET アプリケーションを接続できるようにするために、アプリケーション構成ファイルに加える必要がある具体的な変更の内容が示されています。

	古いバージョンの IBM MQ でコンパイルされたアプリケーション	新しいバージョンの IBM MQ でコンパイルされたアプリケーション
新しいバージョンの IBM MQ プライマリー・インストール済み環境でアプリケーションを実行する場合 (GAC の新しいバージョンのアセンブリーを使用):	変更の必要はありません。	変更の必要はありません。

表 27. 特定のアセンブリーを使用するためのアプリケーションの構成 (続き)

	古いバージョンの IBM MQ でコンパイルされたアプリケーション	新しいバージョンの IBM MQ でコンパイルされたアプリケーション
古いバージョンの IBM MQ プライマリー・インストール済み環境でアプリケーションを実行する場合 (GAC の古いバージョンのアセンブリーを使用):	変更の必要はありません。	アプリケーション構成ファイルで以下のようになります。 <ul style="list-style-type: none"> <li>• <i>bindingRedirect</i> タグを使用して、GAC にある古いバージョンのアセンブリーを使用するように指示する。</li> </ul>
新しいバージョンの IBM MQ のプライマリー以外のインストール済み環境でアプリケーションを実行する場合 (インストール・フォルダーの新しいバージョンのアセンブリーを使用):	アプリケーション構成ファイルで以下のようになります。 <ul style="list-style-type: none"> <li>• <i>codebase</i> タグを使用して、新しいバージョンのアセンブリーの場所を指すようにする。</li> <li>• <i>bindingRedirect</i> タグを使用して、新しいバージョンのアセンブリーを使用するように指示する。</li> </ul>	アプリケーション構成ファイルで以下のようになります。 <ul style="list-style-type: none"> <li>• <i>codebase</i> タグを使用して、新しいバージョンのアセンブリーの場所を指すようにする。</li> </ul>
古いバージョンの IBM MQ のプライマリー以外のインストール済み環境でアプリケーションを実行する場合 (インストール・フォルダーの古いバージョンのアセンブリーを使用):	アプリケーション構成ファイルで以下のようになります。 <ul style="list-style-type: none"> <li>• <i>codebase</i> タグを使用して、古いバージョンのアセンブリーの場所を指すようにする。</li> <li>• <i>publisherpolicy Apply=no</i> タグを組み込む。</li> </ul>	アプリケーション構成ファイルで以下のようになります。 <ul style="list-style-type: none"> <li>• <i>codebase</i> タグを使用して、古いバージョンのアセンブリーの場所を指すようにする。</li> <li>• <i>bindingRedirect</i> タグを使用して、古いバージョンのアセンブリーを使用するように指示する。</li> <li>• <i>publisherpolicy Apply=no</i> タグを組み込む。</li> </ul>

サンプルのアプリケーション構成ファイル `NonPrimaryRedirect.config` は、`MQ_INSTALLATION_PATH\tools\dotnet\samples\base` フォルダーに入れられて出荷されています。このファイルは、プライマリー以外のインストール済み環境の IBM MQ インストール・パスを使用して変更できます。また、このファイルは、*linkedConfiguration* タグを使用して、他の構成ファイルに直接含めることができます。 `nmqsget.exe.config` および `nmqspout.exe.config` のサンプルが用意されています。両方のサンプルで *linkedConfiguration* タグが使用されます。これらのサンプルには `NonPrimaryRedirect.config` ファイルが含まれています。

## DEVPATH を使用する .NET アプリケーションの接続

`DEVPATH` 環境変数を使用して、アセンブリーを検索することができます。`DEVPATH` 変数で指定されたアセンブリーは、GAC のどのアセンブリーよりも優先して使用されます。この変数をいつ使用するかについて詳しくは、該当する Microsoft の `DEVPATH` に関する資料を参照してください。

`DEVPATH` 環境変数を使用してアセンブリーを検索するには、使用するアセンブリーを含むフォルダーに `DEVPATH` 変数を設定する必要があります。その後、アプリケーション構成ファイルを更新し、以下のランタイム構成情報を追加する必要があります。

```
<configuration>
<runtime>
<developmentMode developerInstallation="true"/>
```

```
</runtime>  
</configuration>
```

## 関連概念

383 ページの『[複数のインストール環境でのアプリケーションの接続](#)』

UNIX, Linux, and Windows システムに IBM WebSphere MQ 7.1 以降のライブラリーがロードされると、それ以上操作を行わなくても、IBM MQ によって自動的に適切なライブラリーが使用されます。IBM MQ は、アプリケーションが接続されるキュー・マネージャーと関連付けられているインストール済み環境からのライブラリーを使用します。

## 関連情報

[プライマリー・インストールの選択](#)

[次を使用して.NET](#)

[複数のインストール](#)

## ULW プライマリー・インストールの変更

**setmqinst** コマンドを使用すると、インストールをプライマリー・インストールとして設定する、あるいはその設定を解除することができます。

## このタスクについて

このタスクは、UNIX, Linux, and Windows に適用されます。

プライマリー・インストールとは、必要とされている、システム全体が関係する場所が参照するインストールです。プライマリー・インストールと、プライマリー・インストールを選択するための考慮事項について詳しくは、[プライマリー・インストールの選択](#)を参照してください。

IBM WebSphere MQ 7.1 以降のインストールが IBM WebSphere MQ 7.0.1 のインストールと共存している場合は、IBM WebSphere MQ 7.0.1 のインストールをプライマリー・インストールにする必要があります。このインストールは、IBM WebSphere MQ 7.1 以降のバージョンがインストールされている場合にはプライマリーとしてフラグが立てられ、IBM WebSphere MQ 7.1 以降をプライマリーにすることはできません。

**Windows** Windows でのインストール・プロセス中に、そのインストールをプライマリー・インストールにすることを指定できます。

**Linux** **UNIX** UNIX and Linux システムの場合は、インストール後に **setmqinst** コマンドを実行して、そのインストールをプライマリー・インストールとして設定する必要があります。

393 ページの『[プライマリー・インストールの設定](#)』。

394 ページの『[プライマリー・インストールの設定解除](#)』。

## プライマリー・インストールの設定

### 手順

インストールをプライマリー・インストールとして設定するには、以下のようにします。

1. 次のコマンドを入力して、インストールが既にプライマリー・インストールになっているかどうかを確認します。

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

ここで、`MQ_INSTALLATION_PATH` は、IBM WebSphere MQ 7.1 以降のインストールのインストール・パスです。

2. 既存の IBM WebSphere MQ 7.1 以降のインストールがプライマリー・インストールとして設定されている場合は、[394 ページの『プライマリー・インストールの設定解除』](#)の手順に従って、その設定を解除します。システムに IBM WebSphere MQ 7.0.1 がインストールされている場合は、プライマリー・インストールを変更することはできません。

3. 適切な権限を使用してログオンしていることを確認します。

- ▶ **UNIX** UNIX and Linux では root として。
- ▶ **Linux** Windows システムでは Administrators グループのメンバーとして。

4. 次のいずれかのコマンドを入力します。

- プライマリー・インストールにするインストールのパスを使用してプライマリー・インストールを設定するには、以下のようになります。

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH
```

- プライマリー・インストールにするインストールの名前を使用してプライマリー・インストールを設定するには、以下のようになります。

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -n installationName
```

5. **Windows**

Windows システムの場合は、システムを再始動します。

## プライマリー・インストールの設定解除

### 手順

プライマリー・インストールとしてのインストールを設定解除するには、以下のようになります。

1. 次のコマンドを入力して、プライマリー・インストールになっているインストールを確認します。

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

ここで、MQ\_INSTALLATION\_PATH は、IBM WebSphere MQ 7.1 以降のインストールのインストール・パスです。

IBM WebSphere MQ 7.0.1 がプライマリー・インストールの場合は、そのプライマリー・インストールを設定解除することはできません。

2. 適切な権限を使用してログオンしていることを確認します。

- ▶ **UNIX** UNIX and Linux では root として。
- ▶ **Linux** Windows システムでは Administrators グループのメンバーとして。

3. 次のいずれかのコマンドを入力します。

- プライマリー・インストールにしておく必要がなくなったインストールのパスを使用してプライマリー・インストールを設定解除するには、以下のようになります。

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -p MQ_INSTALLATION_PATH
```

- プライマリー・インストールにしておく必要がなくなったインストールの名前を使用してプライマリー・インストールを設定解除するには、以下のようになります。

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -n installationName
```

### 関連情報

[Windows 上のプライマリー・インストールでのみ使用できる機能](#)

[UNIX and Linux 上のプライマリー・インストールへの外部ライブラリーおよび制御コマンド・リンク](#)

[プライマリー・インストールのアンインストール、アップグレード、および保守](#)

[インストール名の選択](#)

[setmqinst](#)

キュー・マネージャーを作成すると、そのキュー・マネージャーは、**crtmqm** コマンドを実行したインストールに自動的に関連付けられます。UNIX, Linux, and Windows では、**setmqm** コマンドを使用して、キュー・マネージャーに関連付けられているインストールを変更できます。

## このタスクについて

キュー・マネージャーが関連付けられているインストールは、そのキュー・マネージャーに制限を課します。つまり、そのキュー・マネージャーは、そのインストールからのコマンドでしか管理できないのです。これには、以下に示した重要な3つの例外があります。

- **setmqm** は、キュー・マネージャーに関連付けられているインストールを変更します。このコマンドは、キュー・マネージャーが現在関連付けられているインストールからではなく、キュー・マネージャーに関連付けたいインストールから実行しなければなりません。**setmqm** コマンドによって指定されたインストール名は、このコマンドの実行元であるインストールに一致している必要があります。
- 通常 **strmqm** は、キュー・マネージャーに関連付けられているインストールから実行する必要があります。ただし、IBM WebSphere MQ 7.0.1 またはそれよりも前のキュー・マネージャーが初めて IBM WebSphere MQ 7.1 以降のインストールで始動するときには、**strmqm** を使用できます。この場合は、**strmqm** がキュー・マネージャーを始動し、このコマンドの実行元のインストールにそのキュー・マネージャーを関連付けます。
- **dspmq** は、**dspmq** コマンドと同じインストールに関連付けられているキュー・マネージャーだけでなく、システム上のすべてのキュー・マネージャーに関する情報を表示します。**dspmq -o installation** コマンドは、どのキュー・マネージャーがどのインストールに関連付けられているのかについての情報を表示します。

HA 環境の場合、**addmqinf** コマンドは **addmqinf** コマンドの実行元のインストールにキュー・マネージャーを自動的に関連付けます。**strmqm** コマンドが **addmqinf** コマンドと同じインストールから実行される限り、これ以上のセットアップは必要ありません。別のインストールを使用しているキュー・マネージャーを始動するには、最初に **setmqm** コマンドを使用して、そのキュー・マネージャーに関連付けられているインストールを変更する必要があります。

キュー・マネージャーをインストール済み環境に関連付ける場合は、**setmqm** コマンドを以下の方法で使用できます。

- IBM MQ の等価のバージョン間で個々のキュー・マネージャーを移動する。例えば、テスト・システムから実動システムにキュー・マネージャーを移動します。
- 古いバージョンの IBM MQ から新しいバージョンの IBM MQ に個々のキュー・マネージャーをマイグレーションする。キュー・マネージャーをバージョン間でマイグレーションすることには、留意しなければならぬさまざまな影響があります。マイグレーションの詳細については、[保守とマイグレーション](#) を参照してください。

## 手順

1. 現在キュー・マネージャーに関連付けられているインストールから **endmqm** コマンドを使用してそのキュー・マネージャーを停止します。
2. 別のインストールから **setmqm** コマンドを使用して、キュー・マネージャーをそのインストールに関連付けます。

例えば、Installation2 という名前のインストールに関連付けられるようキュー・マネージャー QMB を設定するには、Installation2 から次のコマンドを入力します。

```
MQ_INSTALLATION_PATH/bin/setmqm -m QMB -n Installation2
```

ここで、**MQ\_INSTALLATION\_PATH** は、Installation2 がインストールされている場所のパスです。

3. 現在キュー・マネージャーに関連付けられているインストールから **strmqm** コマンドを使用して、そのキュー・マネージャーを始動します。

このコマンドは、必要とされるキュー・マネージャーのマイグレーションを実行し、結果として、そのキュー・マネージャーを使用する準備が完了します。

## 次のタスク

キュー・マネージャーが関連付けられているインストールが削除されていた場合、またはキュー・マネージャーの状況情報を使用できない場合、**setmqm** コマンドはキュー・マネージャーと別のインストールとの関連付けに失敗します。この状況では、以下のアクションを実行します。

1. **dspmqinst** コマンドを使用して、システム上の別のインストールを表示します。
2. `mqs.ini` 内の `QueueManager` スタンザの `InstallationName` フィールドを手動で変更し、別のインストールを指定します。
3. そのインストールから **dltmqm** コマンドを使用して、キュー・マネージャーを削除します。

### 関連概念

396 ページの『システム上の IBM MQ のインストールの見つけ方』

システムに複数の IBM MQ インストールがある場合、インストールされているバージョンとそれらの場所を確認できます。

88 ページの『IBM MQ 構成ファイル `mqs.ini`』

IBM MQ 構成ファイル `mqs.ini` には、ノード上のすべてのキュー・マネージャーに関連する情報が入っています。これは、インストール時に自動的に作成されます。

### 関連情報

[プライマリー・インストールの選択](#)

[addmqinf](#)

[dspmq](#)

[dspmqinst](#)

[endmqm](#)

[setmqm](#)

[strmqm](#)

ULW

## システム上の IBM MQ のインストールの見つけ方

システムに複数の IBM MQ インストールがある場合、インストールされているバージョンとそれらの場所を確認できます。

システム上の IBM MQ のインストールを見つけるには、以下の方法を使用します。

- **dspmqver** コマンドを使用する。このコマンドが IBM WebSphere MQ 7.0.1 のインストールから実行された場合は、このコマンドによってシステム上のすべてのインストールの詳細が得られるわけではありません。
- IBM MQ がインストールされた場所を照会するには、プラットフォームのインストール・ツールを使用します。次に、IBM WebSphere MQ 7.1 以降のインストールから **dspmqver** コマンドを使用します。以下のコマンドは、IBM MQ のインストール先を照会するために使用できるコマンドの例です。
  - AIX システムの場合は、**lslpp** コマンドを使用できます。

```
lslpp -R ALL -l mqm.base.runtime
```

- HP-UX システムの場合は、**swlist** コマンドを使用できます。

```
swlist -a location -a revision -l product MQSERIES
```

- Linux システムの場合は、**rpm** コマンドを使用できます。

```
rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
```

– Solaris システムの場合は、**pkginfo** および **pkgparam** コマンドを使用できます。

1. 次のコマンドを入力して、インストール済みパッケージをリストします。

```
pkginfo | grep -w mqm
```

2. リストされたパッケージごとに、次のコマンドを入力します。

```
pkgparam pkgname BASEDIR
```

– Windows システムの場合は、**wmic** コマンドを使用できます。このコマンドでは、wmic クライアントをインストールできます。

```
wmic product where "(Name like '%MQ%') AND (not Name like '%bitSupport%')" get Name, Version, InstallLocation
```

- UNIX and Linux システムでは、以下のコマンドを実行して、IBM MQ がインストールされている場所を見つけます。

```
cat /etc/opt/mqm/mqinst.ini
```

次に、IBM WebSphere MQ 7.1 以降のインストールから **dspmqrver** コマンドを使用します。

- 32 ビット Windows でシステム上のインストール済み環境の詳細を表示するには、次のコマンドを実行します。

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation" /s
```

- 64 ビット Windows では、次のコマンドを実行します。

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\WebSphere MQ\Installation" /s
```

注: **reg.exe** コマンドでは、IBM WebSphere MQ 7.1 以降のインストールに関する情報のみが表示されます。

## 関連情報

[dspmqrver](#)

[dspmqrinst](#)

[複数のインストール](#)

## 高可用性、リカバリー、および再始動の構成

キュー・マネージャーに障害が起きた場合にキューの可用性を維持したり、サーバーやストレージの障害後にメッセージをリカバリーしたりして、アプリケーションの可用性を高めます。

### このタスクについて

**z/OS** z/OS では、高可用性がプラットフォームに組み込まれています。キュー共有グループを使用することにより、サーバー・アプリケーションの可用性を向上させることもできます。[共有キューおよびキュー共有グループ](#)を参照してください。

**Multi** マルチプラットフォームでは、クライアント・アプリケーションの可用性を高めるために、クライアントの再接続機能を使用して、キュー・マネージャーのグループ間でクライアントを自動的に切り替えたり、キュー・マネージャーの障害後に複数インスタンス・キュー・マネージャーの新しいアクティブ・インスタンスに切り替えたりできます。IBM MQ classes for Java は自動クライアント再接続をサポートしていません。複数インスタンスキュー・マネージャーを複数のサーバーで1つのキュー・マネージャーとして実行するように構成します。そのキュー・マネージャーにサーバー・アプリケーションをデプ

ロイしてください。アクティブ・インスタンスを実行しているサーバーに障害が起きた場合、別のサーバー上にある同じキュー・マネージャーのスタンバイ・インスタンスに実行が自動的に切り替えられます。サーバー・アプリケーションをキュー・マネージャー・サービスとして実行するように構成した場合、スタンバイ・インスタンスが、アクティブに実行しているキュー・マネージャー・インスタンスになると、サーバー・アプリケーションは再始動されます。

Multiplatforms でサーバー・アプリケーションの可用性を高めるためのもう 1 つの方法は、キュー・マネージャー・クラスター内の複数のコンピューターにサーバー・アプリケーションをデプロイすることです。IBM WebSphere MQ 7.1 からは、クラスター・エラー・リカバリー機能によって、問題が解決されるまで、問題の原因となった操作が再実行されます。z/OS 以外のサーバーにおけるクラスター・エラー・リカバリーの変更を参照してください。IBM MQ for Multiplatforms をプラットフォーム固有の次のようなクラスターリング・ソリューションの一部として構成することも可能です。

- Microsoft Cluster Server

-  IBM i での HA クラスター

-   PowerHA® for AIX (旧称: HACMP on AIX) およびその他の UNIX and Linux のクラスターリング・ソリューション

メッセージング・システムでは、システムに入ってきたメッセージが確実にその宛先に配布されます。IBM MQ は `dspmqrte` コマンドを使用して、あるキュー・マネージャーから別のキュー・マネージャーに移動するときに、メッセージの経路をトレースできます。システムに障害が起きた場合、障害のタイプやシステムの構成方法によって、様々な方法でメッセージをリカバリーできます。IBM MQ では、メッセージの受信や送信を取り扱うキュー・マネージャーのアクティビティのリカバリー・ログが維持されています。WebSphere MQ は、次の 3 つのタイプのリカバリーに関してログを使用します。

1. 再始動リカバリー。計画どおりに IBM MQ を停止した場合
2. 障害リカバリー。障害によって IBM MQ が停止した場合
3. メディア・リカバリー。損傷したオブジェクトを復元する場合

いずれの場合も、リカバリーによってキュー・マネージャーは、停止した時の状態に復元されます。ただし、不完了トランザクションはロールバックされ、キュー・マネージャーの停止時に未完了であった更新がある場合は、この更新がキューから除去されます。リカバリーによって、持続メッセージはすべて復元されます。このプロセス中に、非持続メッセージが失われることがあります。

## クライアントの自動再接続

追加のコードを作成しなくても、いくつかのコンポーネントを構成することによって、クライアント・アプリケーションが自動的に再接続するようにすることができます。

クライアントの自動再接続はインラインです。接続はクライアント・アプリケーション・プログラムのどのポイントでも自動的に復元され、オブジェクトを開くためのハンドルがすべて復元されます。

対照的に、手動再接続では、クライアント・アプリケーションで MQCONN または MQCONNX を使用して接続を再作成し、オブジェクトを再オープンする必要があります。クライアントの自動再接続は多くのクライアント・アプリケーションに適していますが、すべてのクライアント・アプリケーションに適しているわけではありません。

399 ページの表 28 は、クライアント・ワークステーションにインストールする必要のある、IBM MQ クライアント・サポートの最新のリリースのリストです。アプリケーションがクライアントの自動再接続を使用できるようにするには、示されているレベルの 1 つにクライアント・ワークステーションをアップグレードする必要があります。399 ページの表 29 には、クライアント自動再接続を有効にするためのその他の要件がリストされています。

再接続オプションへのプログラム・アクセスによって、クライアント・アプリケーションは再接続オプションを設定できます。さらに、JMS および XMS クライアント以外では、クライアント・アプリケーションが再接続オプションにアクセスする場合、再接続イベントを扱うためにイベント・ハンドラーを作成することができます。

既存のクライアント・アプリケーションでは、以下のようにして、再コンパイルおよびリンクせずに、再接続のサポートのメリットを受けられます。

- 非 JMS クライアントでは、mqclient.ini 環境変数に DefRecon を設定して再接続オプションを設定します。キュー・マネージャーに接続するには CCDT を使用します。クライアントがマルチ・インスタンスのキュー・マネージャーに接続する場合、CCDT にあるアクティブで待機状態のキュー・マネージャー・インスタンスのネットワーク・アドレスを提供します。
- JMS クライアントでは、接続ファクトリー構成に再接続オプションを設定します。Java EE サーバーの EJB コンテナ内で実行している場合、MDB は、IBM MQ リソース・アダプターのアクティベーション・スペック (WebSphere Application Server で実行している場合はリスナー・ポート) によって提供される再接続メカニズムを使用して、IBM MQ に再接続できます。ただし、アプリケーションが MDB でない (または Web コンテナ内で実行されている) 場合、このシナリオではクライアントの自動再接続がサポートされていないため、アプリケーションは独自の再接続ロジックを実装する必要があります。IBM MQ リソース・アダプターは、メッセージ駆動型 Bean にメッセージを配信するためにこの再接続機能を提供しますが、サーブレットなどのその他の Java EE エlement は、独自の再接続を実装する必要があります。

注：IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

クライアント・インターフェース	クライアント	再接続オプションへのプログラム・アクセス	再接続サポート
API メッセージング	C、C++、COBOL、非管理の Visual Basic、XMS (Windows 上の管理対象でない XMS)	7.0.1	7.0.1
	JMS (JSE、および Java EE クライアント・コンテナと管理対象のコンテナ)	7.0.1.3	7.0.1.3
	IBM MQ classes for Java	サポート対象外	サポート対象外
	管理対象の XMS および管理対象の .NET クライアント：C#、Visual Basic	7.1	7.1
その他の API	Windows Communication Foundation (管理対象でない <sup>1</sup> )	サポート対象外	7.0.1
	Windows Communication Foundation (管理対象 <sup>1</sup> )	サポート対象外	サポート対象外
	Axis 1	サポート対象外	サポート対象外
	Axis 2	サポート対象外	7.0.1.3
	HTTP (Web 2.0)	サポート対象外	7.0.1.3

1. WCF バインディング構成で、管理対象か管理対象でないかを設定します。

自動再接続には、以下の構成要件があります。

コンポーネント	要件	要件に適合しない場合の影響
IBM MQ MQI client インストール	399 ページの表 28 を参照してください。	MQRC_OPTIONS_ERROR
IBM MQ サーバー・インストール	レベル 7.0.1	MQRC_OPTIONS_ERROR
チャンネル	SHARECNV > 0	MQRC_ENVIRONMENT_ERROR
アプリケーション環境	スレッド化必須	MQRC_ENVIRONMENT_ERROR

表 29. 自動再接続の構成要件 (続き)

コンポーネント	要件	要件に適合しない場合の影響
MQI	以下のいずれかです。 <ul style="list-style-type: none"> <li>MQCONNでMQCNO オプションをMQCNO_RECONNECTまたはMQCNO_RECONNECT_Q_MGRに設定。</li> <li>mqclient.iniのDefrecon=YES QMGR</li> <li>JMS内で、接続ファクトリーのCLIENTRECONNECTOPTIONSプロパティを設定します。</li> </ul>	接続での障害発生時、あるいはキュー・マネージャーの終了時または障害発生時のMQCC_FAILED

400 ページの図 70 は、クライアントの再接続に関するコンポーネント間の主な相互作用を示しています。

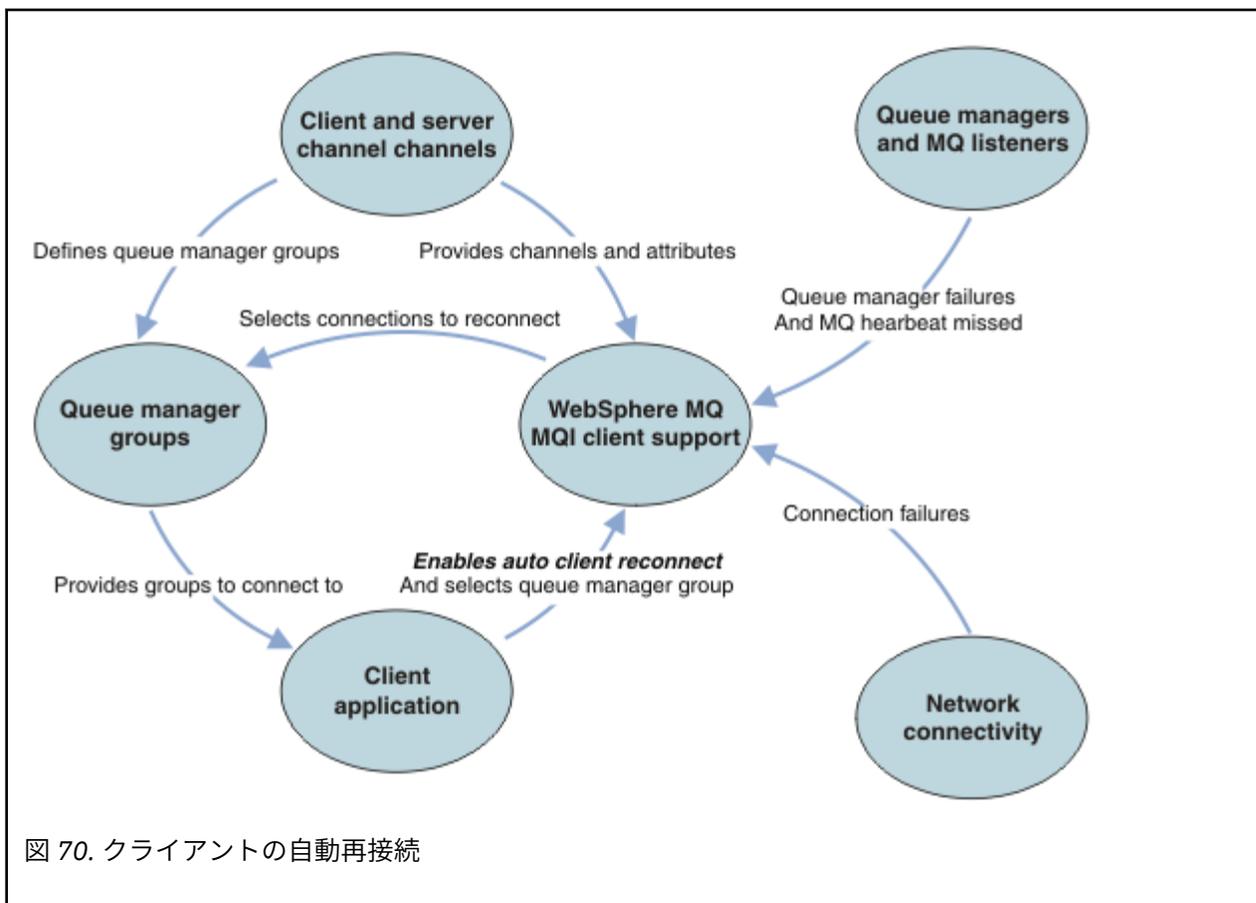


図 70. クライアントの自動再接続

## クライアント・アプリケーション

クライアント・アプリケーションは IBM MQ MQI client です。

- デフォルトでは、クライアントの自動再接続は行われません。クライアントの自動再接続は、MQCONNでMQCNO オプションMQCNO\_RECONNECTまたはMQCNO\_RECONNECT\_Q\_MGRを設定することにより有効になります。
- 追加のコーディングを行わなくても自動再接続を利用できるように、多くのアプリケーションがこのような方法で作成されています。mqclient.ini 構成ファイル内のチャンネル・スタンザのDefRecon 属性を設定することで、コードを変更することなく、既存のプログラムの自動再接続を有効にできます。

- 以下の3つの選択肢のいずれかを使用します。
  1. ロジックが再接続の影響を受けないようにプログラムを変更する。例えば、同期点内でMQI呼び出しを発行し、バックアウトされたトランザクションを再実行依頼しなければならない場合があります。
  2. 再接続を検出するイベント・ハンドラーを追加し、接続が再度確立されたときに、クライアント・アプリケーションの状態を復元する。
  3. 自動再接続を有効にしない。代わりにクライアントを切断し、MQCONN または MQCONNX いずれかのMQI呼び出しを新たに発行して、同じキュー・マネージャー・グループで実行されている別のキュー・マネージャー・インスタンスを見つける。

この3つの選択肢について、491 ページの『[アプリケーションの復旧](#)』のトピックで詳しく説明されています。

- 同じ名前のキュー・マネージャーに再接続しても、それは同じキュー・マネージャーのインスタンスに再接続したという保証にはなりません。

同じキュー・マネージャーのインスタンスに再接続するには、MQCNO オプションの MQCNO\_RECONNECT\_Q\_MGR を使用します。

- クライアントでは、再接続の状態に関する通知を受けられるようにイベント・ハンドラーを登録することができます。イベント・ハンドラーで渡される MQHCONN は使用できません。以下の理由コードが用意されています。

#### **MQRC\_RECONNECTING**

接続に障害が起きたため、システムが再接続を試行しています。再接続が複数回試行された場合には、複数の MQRC\_RECONNECTING イベントを受け取ります。

#### **MQRC\_RECONNECTED**

再接続が実行され、すべてのハンドルが正常に再設定されました。

#### **MQRC\_RECONNECT\_FAILED**

再接続は成功しませんでした。

#### **MQRC\_RECONNECT\_QMID\_MISMATCH**

再接続可能な接続が、MQCNO\_RECONNECT\_Q\_MGR を指定し、接続は別のキュー・マネージャーに対して再接続を試みました。

#### **MQRC\_RECONNECT\_Q\_MGR\_REQD**

MQGET 呼び出しの MQMO\_MATCH\_MSG\_TOKEN といったオプションが、同じキュー・マネージャーへの再接続を必要とするクライアント・プログラムで指定されました。

- 再接続可能クライアントは、接続後に初めて自動再接続可能になります。すなわち、MQCONNX 呼び出し自体は、失敗した場合は再試行されません。例えば、MQCONNX から戻りコード 2543 - MQRC\_STANDBY\_Q\_MGR を受け取った場合は、少し経ってから呼び出しを再発行する必要があります。

#### **MQRC\_RECONNECT\_INCOMPATIBLE**

この理由コードは、アプリケーションが、再接続オプションを設定する際に、MQPMO\_LOGICAL\_ORDER (MQPUT および MQPUT1 で)、または MQGMO\_LOGICAL\_ORDER (MQGET で) の使用を試みた場合に返されます。理由コードが戻される理由は、今後同様なケースにおいてアプリケーションが二度と再接続を使用しないようにするためです。

#### **MQRC\_CALL\_INTERRUPTED**

この理由コードは、Commit 呼び出しの実行およびクライアント再接続の最中に接続が中断された場合に返されます。同期点の外側の持続メッセージの MQPUT によっても、アプリケーションに同じ理由コードが戻される結果になります。

## 複数インスタンス・キュー・マネージャー

複数インスタンスのキュー・マネージャーがスタンバイ・インスタンスをアクティブ化した後で、クライアント自動再接続を使って、単純に IBM MQ MQI client ・アプリケーションを再始動します。

複数インスタンス・キュー・マネージャーのスタンバイ・インスタンスは、通常、アクティブ・インスタンスとは異なるネットワーク・アドレス上に置かれます。クライアント接続定義表 (CCDT) には、両方のイ

インスタンスのネットワーク・アドレスが含まれます。 **CONNAME** パラメーターのネットワーク・アドレスのリストを提供するか、もしくは **CCDT** でキュー・マネージャー用に複数の行を定義します。

一般的に、IBM MQ MQI clients はキュー・マネージャー・グループ内の任意のキュー・マネージャーに再接続します。 IBM MQ MQI client が、同じキュー・マネージャーにだけ再接続するようにしたい場合があります。 1つのキュー・マネージャーに対して、親和性を持つ可能性があります。 クライアントが異なるキュー・マネージャーに再接続しないようにすることができます。 **MQCNO** オプションに **MQCNO\_RECONNECT\_Q\_MGR** を設定します。 IBM MQ MQI client は、異なるキュー・マネージャーに再接続された場合、失敗します。 **MQCNO** オプションに **MQCNO\_RECONNECT\_Q\_MGR** を設定する場合、同じキュー・マネージャー・グループに他のキュー・マネージャーを含めないようにしてください。 クライアントが再接続するキュー・マネージャーが以前接続したのと同じでない場合、クライアントはエラーを戻します。

## キュー・マネージャー・グループ

クライアント・アプリケーションの接続先および再接続先が、常に同じ名前のキュー・マネージャーか、同じキュー・マネージャーか、またはクライアント接続テーブルで同じ **QMNAME** で定義された一連のキュー・マネージャーのうちのいずれかであるかを選択できます。

- クライアント・チャンネル定義にあるキュー・マネージャーの名前属性 **QMNAME** は、キュー・マネージャー・グループの名前です。
- クライアント・アプリケーションで、**MQCONN** または **MQCONNX** の **QmgrName** パラメーターの値をキュー・マネージャー名に設定した場合、クライアントはその名前のキュー・マネージャーにのみ接続します。 キュー・マネージャー名にアスタリスク (\*) の接頭部を付けた場合、クライアントは、同じ **QMNAME** 値を持つキュー・マネージャー・グループ内の任意のキュー・マネージャーに接続します。 詳しい説明については、[CCDT のキュー・マネージャー・グループ](#)を参照してください。

## キュー共有グループ

**z/OS** z/OS キュー共有グループへのクライアント自動再接続は、他の任意の環境における再接続と同じ仕組みを使用します。 クライアントは、元の接続のために構成したのと同じキュー・マネージャーのセットに対して再接続します。 例えば、クライアント・チャンネル定義テーブルを使用する場合、管理者は、テーブルのすべての項目が、同じ z/OS キュー共有グループに解決されるようにする必要があります。

## クライアントおよびサーバーのチャンネル定義

クライアントおよびサーバーのチャンネル定義では、クライアント・アプリケーションが再接続可能なキュー・マネージャーのグループを定義します。 この定義は、再接続の選択内容やタイミング、およびセキュリティなどの他の要因を管理します (関連トピックを参照)。 再接続の面で最も関係のあるチャンネル属性を、以下の2つのグループにリストします。

### クライアント接続の属性

#### **Connection affinity (AFFINITY) AFFINITY**

接続のアフィニティ。

#### **Client channel weight (CLNTWGHT) CLNTWGHT**

クライアント・チャンネル・ウェイト。

#### **Connection name (CONNAME) CONNAME**

接続情報。

#### **Heartbeat interval (HBINT) HBINT**

ハートビート間隔。 サーバー接続チャンネルのハートビート間隔を設定します。

#### **Keepalive Interval (KAINT) KAINT**

キープアライブ間隔。 サーバー接続チャンネルのキープアライブ間隔を設定します。

**z/OS** KAINT は z/OS にのみ適用されることに注意してください。

#### **Queue manager name (QMNAME) QMNAME**

キュー・マネージャー名。

## サーバー接続の属性

### Heartbeat interval (HBINT) HBINT

ハートビート間隔。クライアント接続チャンネルのハートビート間隔を設定します。

### Keepalive Interval (KAINT) KAINT

キープアライブ間隔。クライアント接続チャンネルのキープアライブ間隔を設定します。

 KAINT は z/OS にのみ適用されることに注意してください。

KAINT はネットワーク層のハートビートで、HBINT はクライアントとキュー・マネージャーの間の IBM MQ ハートビートです。これらのハートビートをより短い時間に設定すると、以下の 2 つの効果があります。

1. 接続上のアクティビティーがシミュレートされるので、非アクティブな接続のクローズを受け持つネットワーク層ソフトウェアが接続をシャットダウンする可能性が低くなります。
2. 接続がシャットダウンされた場合に、障害の起きた接続を検出するまでの遅延時間が短くなります。

デフォルトの TCP/IP キープアライブ間隔は 2 時間です。KAINT および HBINT 属性の設定時間を短くすることを検討してください。ネットワークの通常動作が自動再接続の要件に適しているとは限りません。例えば、ファイアウォールによっては、非アクティブの TCP/IP 接続をシャットダウンするまでに、少なくとも 10 分を要します。

## ネットワーク 接続

ネットワークから IBM MQ MQI client に渡されるネットワーク障害のみが、クライアントの自動再接続機能によって処理されます。

- トランSPORTによって自動的に実行される再接続は、IBM MQ には認識されません。
- HBINT を設定すると、IBM MQ では認識されないネットワーク障害を処理する上で役立ちます。

## キュー・マネージャーおよび IBM MQ リスナー

クライアントの再接続は、サーバーの障害、キュー・マネージャーの障害、ネットワーク接続の障害、さらに管理者による別のキュー・マネージャー・インスタンスへの切り替えによって起動されます。

- 複数インスタンス・キュー・マネージャーを使用している場合は、アクティブなキュー・マネージャー・インスタンスからスタンバイ・インスタンスに制御を切り替えるときにも、クライアントの再接続が発生します。
- デフォルトの `endmqm` コマンドを使用してキュー・マネージャーを終了しても、クライアントの自動再接続はトリガーされません。`endmqm` コマンドに、クライアント自動再接続を要求する `-r` オプション、またはシャットダウン後にスタンバイ・キュー・マネージャー・インスタンスに転送する `-s` オプションを追加してください。

## IBM MQ MQI client の自動再接続サポート

IBM MQ MQI client でクライアント自動再接続サポートを使用すると、キュー・マネージャーに再接続するための `MQCONN MQI` 呼び出しまたは `MQCONNX MQI` 呼び出しを発行しなくても、クライアント・アプリケーションの再接続が自動的に実行され、クライアント・アプリケーションが処理を続行します。

- クライアント自動再接続は、以下の状況のいずれかに該当する場合にトリガーされます。
  - キュー・マネージャーに障害が発生した
  - `endmqm` コマンドに `-r` (再接続) オプションを指定してキュー・マネージャーが終了した
- `MQCONNX MQCNO` オプションは、クライアント自動再接続を有効にするかどうかを制御します。オプションについては、[再接続オプション](#)で説明されています。
- クライアント自動再接続は、アプリケーションの代わりに MQI 呼び出しを発行して、接続ハンドルおよび他の開いているオブジェクトへのハンドルを復元します。それにより、プログラムは、接続の障害により発生した MQI エラーを処理した後で、通常の処理を再開することができます。[493 ページの『自動再接続されるクライアントの復旧』](#)を参照してください。

- 接続に対してチャンネル出口プログラムを作成した場合は、その出口がこれらの追加の MQI 呼び出しを受け取ります。
- 再接続の開始時および終了時にトリガーされる再接続イベント・ハンドラーを登録することができます。

再接続にかかる時間は 1 分ほどですが、キュー・マネージャーに大量の管理対象リソースがあるためにそれ以上の時間がかかることがあります。この間、クライアント・アプリケーションが IBM MQ リソースに属さないロックを保持し続ける可能性があります。クライアントの再接続の待ち時間を制限するために構成可能なタイムアウト値が 1 つあります。この値 (秒単位) は、mqclient.ini ファイルで設定します。

```
Channels:
MQReconnectTimeout = 1800
```

タイムアウト後、再接続は試行されなくなります。システムは、タイムアウトになったことを検出すると、MQRC\_RECONNECT\_FAILED エラーを返します。

## z/OS コンソール・メッセージのモニター

IBM MQ for z/OS では、キュー・マネージャーまたはチャンネル・イニシエーターから送信される情報メッセージの中に、特に重要と見なすべきメッセージが多数存在します。そうしたメッセージは、問題があることを示しているわけではありませんが、対応が必要になる可能性のある問題が潜んでいることを示しているので、トラッキングしておく役立ちます。

これらのコンソール・メッセージが出されたということは、ユーザー・アプリケーションが多数のメッセージをそのページ・セットに入れたことを示している場合もあり、以下のような大きな問題の徴候である可能性があります。

- メッセージを PUT するユーザー・アプリケーションに、制御不能なループなどの問題がある。
- キューからメッセージを GET するユーザー・アプリケーションが機能しなくなっている。

### モニターするコンソール・メッセージ

以下のリストは、大きな問題を示している可能性があるメッセージの概略を示しています。システムの自動化を使用してこれらのメッセージをトラッキングする必要があるかどうかを判別し、適切な文書を用意して、すべての潜在的な問題を効果的に追跡できるようにしてください。

**CSQI004I: csect-name connection-type の接続 connection-name のために、キュー queue-name に索引タイプ index-type の索引を作成することを検討してください。num-msgs 個のメッセージがスキップされました**

- アプリケーションが、索引が定義されていないキューからメッセージ ID または関連 ID を使用してメッセージを受け取ったことを、キュー・マネージャーが検出しました。
- 特定されたキューに索引を設定するために、ローカル・キュー・オブジェクト queue-name の INDXTYPE 属性の値が index-type となるように変更することを検討してください。

**CSQI031I: csect-name ページ・セット psid の新規エクステン트는正常にフォーマットされました。**

- このページ・セットに割り振られているキューの curdepth を確認します。
- メッセージを処理するために、障害の原因を調べます。

**CSQI041I: csect-name ジョブ jobname ユーザー userid がページ・セット psid へのアクセス中にエラー。**

- ページ・セットがキュー・マネージャーに割り振られているかどうかを判別します。
- **DISPLAY USAGE** コマンドを発行して、ページ・セットの状態を判別します。
- キュー・マネージャーのジョブ・ログに追加のエラー・メッセージがあるかどうかを確認します。

**CSQI045I: csect-name ログ RBA が rba に達しました。ログのリセットを計画します。**

- 適当なときにキュー・マネージャーを停止し、ログをリセットする。

- ・キュー・マネージャーが6バイトのログRBAを使用している場合は、8バイトのログRBAを使用するようキュー・マネージャーを変換することを検討してください。

**CSQI046E: csect-name ログ RBA が rba に達しました。ログのリセットを実行します。**

- ・適当なときにキュー・マネージャーを停止し、ログをリセットする。
- ・キュー・マネージャーが6バイトのログRBAを使用している場合は、8バイトのログRBAを使用するようキュー・マネージャーを変換することを検討してください。

**CSQI047E: csect-name ログ RBA が rba に達しました。キュー・マネージャーを停止しログをリセットします。**

- ・即座にキュー・マネージャーを停止し、ログをリセットする。
- ・キュー・マネージャーが6バイトのログRBAを使用している場合は、8バイトのログRBAを使用するようキュー・マネージャーを変換することを検討してください。

**CSQJ004I: アクティブ・ログのコピー n は非活動で、単一モードです。ENDRBA=ttt**

- ・キュー・マネージャーによって「single」ロギング・モードがアクティブ化されました。これは通常、ログのオフロードの問題を示しています。
- ・**DISPLAY LOG** コマンドを発行して、二重のアクティブ・ログおよびアーカイブ・ログに関する設定を判別します。この表示は、オフロード処理を必要とするアクティブ・ログの数も示します。
- ・キュー・マネージャーのジョブ・ログに追加のエラー・メッセージがあるかどうかを確認します。

**CSQJ031D: csect-name、ログ RBA 範囲をリセットする必要があります。起動を続行するには「Y」を、シャットダウンするには「N」を入力してください**

- ・できるだけ早くキュー・マネージャーを停止し、ログをリセットします。
- ・キュー・マネージャーが6バイトのログRBAを使用している場合は、8バイトのログRBAを使用するようキュー・マネージャーを変換することを検討してください。

**CSQJ032E: csect-name alert-lvl - ログ RBA 範囲の終端 max-rba に近づきました。現行のログ RBA は current-rba です。**

- ・できるだけ早くキュー・マネージャーを停止してログをリセットする計画を立ててください。
- ・キュー・マネージャーが6バイトのログRBAを使用している場合は、8バイトのログRBAを使用するようキュー・マネージャーを変換することを検討してください。

**CSQJ110E: 活動ログ・データ・セットの最新コピー n は nnn パーセントまできています**

- ・ログのオフロード処理に関連する未解決の要求を判別するための表示要求を実行して、他の待ちオフロード・タスクを完了する手順を実行する。要求を満たすのに必要な処置を取り、オフロード処理を続行できるようにする。
- ・十分なアクティブ・ログ・データ・セットがあるかどうかを確認する。必要に応じて、**DEFINE LOG** コマンドを使用して、動的にログ・データ・セットを追加できる。

**CSQJ111A: 活動ログ・データ・セットにスペースがありません**

- ・表示要求を実行し、ログのオフロード処理に関連して未解決の要求がないかどうかを確かめる。要求を満たすのに必要な処置を取り、オフロード処理を続行できるようにする。
- ・十分なアクティブ・ログ・データ・セットがあるかどうかを確認する。必要に応じて、**DEFINE LOG** コマンドを使用して、動的にログ・データ・セットを追加できる。
- ・オフロードに必要な資源が不足しているために遅延が生じた場合は、必要な資源を使用可能にして、オフロードを完了させ、ロギングを続行できるようにしなければならない。この状態から回復する方法については、アーカイブ・ログの問題を参照。

**CSQJ114I: 保存データ・セットにエラーが発生し、オフロードは生成中の保存データ・セットでだけ継続します。**

- ・キュー・マネージャーのジョブ・ログに追加のエラー・メッセージがあるかどうかを確認します。
- ・アーカイブ・ログのもう1つのコピーを作成して、ご使用のBSDSを手動で更新します。

**CSQJ115E: オフロードが失敗しました。保存データ・セットを割り振ることができませんでした。**

メッセージ CSQJ103E または CSQJ073E のエラー状況情報を調べる。データ・セット割り振りエラーの原因となった状況を訂正し、再試行時にオフロードが行われるようにする。

**CSQJ136I: 磁気テープ装置を接続 ID=xxxx 相関 ID=yyyyyy に割り振れません。m 個割り振られ、n 個許可されました。**

- ・キュー・マネージャーのジョブ・ログに追加のエラー・メッセージがあるかどうかを確認します。

**CSQJ151I: csect-name RBA rrr の読み取り中エラー。CONNECTION-ID=xxxx CORRELATION-ID=yyyyyy REASON CODE=ccc**

- ・キュー・マネージャーのジョブ・ログに追加のメッセージがあるかどうかを確認します。
- ・ **DISPLAY CONN** コマンドを発行して、活動をコミットしていない接続を判別します。
- ・アプリケーションが更新をコミットできることを確認します。

**CSQJ160I: 長時間実行されている UOW が見つかりました。URID=urid CONNECTION NAME=name**

- ・キュー・マネージャーのジョブ・ログに追加のメッセージがあるかどうかを確認します。
- ・ **DISPLAY CONN** コマンドを発行して、活動をコミットしていない接続を判別します。
- ・アプリケーションが更新をコミットできることを確認します。

**CSQJ161I: n オフロードの後、UOW が解決されませんでした。URID=urid CONNECTION NAME=name**

- ・ページ・セットがキュー・マネージャーに割り振られているかどうかを判別します。
- ・ **DISPLAY USAGE** コマンドを発行して、ページ・セットの状態を判別します。
- ・キュー・マネージャーのジョブ・ログに追加のメッセージがあるかどうかを確認します。

**CSQP011E: ページ・セット psid の接続エラー状況 ret-code**

- ・このページ・セットに割り振られているキューの curdepth を確認します。
- ・メッセージを処理するために、障害の原因を調べます。

**CSQP013I: csect-name ページ・セット psid に作成された新規エクステントです。新規エクステントは今フォーマットされます。**

- ・このページ・セットに割り振られているキューの curdepth を確認します。
- ・メッセージを処理するために、障害の原因を調べます。
- ・キューを別のページ・セットに再配置する必要があるかどうかを判別します。
- ・ボリュームが満杯の場合には、ページ・セットをマルチボリュームのデータ・セットにする必要があるかどうかを判別します。ページ・セットがすでにマルチボリュームになっている場合は、使用しているストレージ・グループにさらにボリュームを追加することを検討します。使用できるスペースが増えたら、ページ・セットの **EXPAND** メソッドを **SYSTEM** に設定して、拡張を再試行します。再試行が必要な場合は、**EXPAND** を **SYSTEM** に切り替えて、通常の設定に戻します。

**CSQP014E: csect-name ページ・セット psid の拡張が失敗しました。これ以降の拡張要求は拒否されます。**

- ・このページ・セットに割り振られているキューの curdepth を確認します。
- ・メッセージを処理するために、障害の原因を調べます。
- ・キューを別のページ・セットに再配置する必要があるかどうかを判別します。

**CSQP016E: csect-name ページ・セット psid は範囲の最大数に達しました。もう一度拡張はできません。**

- ・このページ・セットに割り振られているキューの curdepth を確認します。
- ・メッセージを処理するために、障害の原因を調べます。

**CSQP017I: csect-name ページ・セット psid の拡張を開始しました。**

**DISPLAY THREAD** コマンドを発行して、IBM MQ 内の作業単位の状態を判別します。

**CSQP047E: ページ・セットが使用不可である場合、問題が発生する可能性があります。この状態を訂正するアクションをとってください。**

- ・システム・プログラマー応答に従ってください。

**CSQQ008I: nn の回復単位がキュー・マネージャー qqqq において、まだ未確定です。**

- ・送達不能キューの状態を調べます。送達不能キューの PUT が使用不可に設定されていないことを確認します。
- ・送達不能キューが MAXMSG の限度に達していないことを確認します。

**CSQQ113I: psb-name region-id このメッセージを処理できません。**

- ・CSQOUTX データ・セットを確認して、CSQINPX 障害の原因を判別します。
- ・一部のコマンドは処理されない可能性があります。

**CSQX035I: csect-name キュー・マネージャー qmgr-name への接続が停止中、または壊れています。MQCC=mqcc MQRC=mqrc (mqrc-text)。**

- ・MQRCを確認して、障害の原因を判別します。
- ・これらのコードについては、IBM MQ for z/OS のメッセージ、完了コード、および理由コードに記載されています。

**CSQX032I: csect-name 初期化コマンド・ハンドラーが終了しました。**

- ・MQRCを確認して、障害の原因を判別します。
- ・これらのコードについては、IBM MQ for z/OS のメッセージ、完了コード、および理由コードに記載されています。

**CSQX048I: csect-name name のメッセージを変換できません。MQCC=mqcc MQRC=mqrc (mqrc-text)。**

- ・ジョブ・ログを確認して、TCP/IP 障害の原因を判別します。
- ・TCP/IP アドレス・スペースのエラーを確認します。

**CSQX234I: csect-name リスナーが停止しました。TRPTYPE=trptype INDISP=disposition**

- ・リスナーが停止しない場合は、**STOP** コマンドに続いて、TCP/IP アドレス・スペースのエラーを確認してください。
- ・システム・プログラマー応答に従ってください。

**CSQX407I: csect-name クラスター・キュー q-name の定義が矛盾しています。**

- ・クラスター内の複数のクラスター・キューの値が矛盾しています。相違を調べて解決してください。

**CSQX411I: csect-name リポジトリ・マネージャーが停止しました。**

- ・エラーが原因でリポジトリ・マネージャーが停止した場合は、ジョブ・ログでメッセージを確認してください。

**CSQX417I: csect-name 除去されたキュー・マネージャー qmgr-name のクラスター送信者が残っています。**

- ・システム・プログラマー応答に従ってください。

**CSQX418I: csect-name クラスター cluster\_name にリポジトリが1つしかありません。**

- ・高可用性をさらに高めるには、2つの完全リポジトリを使用するようにクラスターを構成する必要があります。

**CSQX419I: csect-name クラスター cluster\_name にクラスター受信側がありません。**

- ・システム・プログラマー応答に従ってください。

**CSQX420I: csect-name クラスター cluster\_name にリポジトリがありません。**

- ・システム・プログラマー応答に従ってください。

**CSQX448E: csect-name** リポジトリ・マネージャーがエラーのため停止しています。n 秒後に再始動します。

- ・システム・プログラマー応答に従ってください。

このメッセージは、以下のコマンドを使用して SYSTEM.CLUSTER.COMMAND.QUEUE を有効にするまで 600 秒 (10 分) ごとに書き出されます。

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
```

キューを有効にする前に、最初の CSQX448E メッセージが発行される前にリポジトリ・マネージャーが終了する原因となった問題を解決するために、手操作による介入が必要になる場合があります。

## 高可用性の構成

IBM MQ キューマネージャを高可用性 (HA) 構成で運用する場合、PowerHAforAIX (旧 HACMP) または Microsoft クラスタサービス (MSCS) などの高可用性マネージャ、または IBM MQ マルチインスタンスのキューマネージャで動作するように設定することができます。 **V 9.0.4** Linux システムでは、クォーラム・ベースのグループを使用して高可用性を提供する複製データ・キュー・マネージャー (RDQM) をデプロイすることもできます。

以下の構成定義に留意する必要があります。

### キュー・マネージャー・クラスター

1 つ以上のコンピューター上にある 2 つ以上のキュー・マネージャーのグループ。自動相互接続を提供し、グループ間でロード・バランシングと冗長度が適切になるようにキューを共有できます。IBM WebSphere MQ 7.1 からは、クラスター・エラー・リカバリー機能によって、問題が解決されるまで、問題の原因となった操作が再実行されます。

### HA クラスター

HA クラスターは、複数のコンピューターと、ディスクやネットワークなどのリソースで構成されたグループです。クラスター間は相互に接続されていて、そのいずれかに障害が生じた場合に HACMP (UNIX) や MSCS (Windows) などの HA マネージャーによってフェイルオーバーが実行されるように構成されます。フェイルオーバーは、障害が起きたコンピューターからアプリケーションの状態データをクラスター内の別のコンピューターに転送して、そこで操作が再開されるようにします。これにより、HA クラスター内で実行されるサービスに対する高可用性が提供されます。IBM MQ クラスターと HA クラスターとの間の関係は、[409 ページの『HA クラスターとキュー・マネージャー・クラスターとの関係』](#)で説明されています。

### 複数インスタンス・キュー・マネージャー

複数のコンピューター上で構成されている同じキュー・マネージャーのインスタンス。複数のインスタンスを開始すると、あるインスタンスがアクティブ・インスタンスになり、それ以外のインスタンスがスタンバイになります。アクティブ・インスタンスが失敗すると、別のコンピューター上で実行しているスタンバイ・インスタンスが自動的に引き継ぎます。複数インスタンス・キュー・マネージャーを使用すれば、HACMP や MSCS などのクラスター・テクノロジーを必要としないで、IBM MQ に基づく独自の可用性の高いメッセージング・システムを構成することができます。HA クラスターおよび複数インスタンス・キュー・マネージャーは、キュー・マネージャーの可用性を高めるための代替方法です。複数インスタンス・キュー・マネージャーを HA クラスター内に入れてそれらを結合しないでください。

### **V 9.0.4** 高可用性複製データ・キュー・マネージャー (HA RDQM)

3 つの Linux サーバーのグループ内の各ノード上で構成されている同じキュー・マネージャーのインスタンス。3 つのインスタンスのいずれかがアクティブ・インスタンスです。アクティブ・キュー・マネージャーからのデータは、他の 2 つのインスタンスに同期的に複製されるため、障害が発生した場合はこれらのインスタンスのいずれかが処理を引き継ぐことができます。サーバーのグループ化は Pacemaker によって制御され、複製は DRBD によって制御されます。

### **V 9.0.5** 災害復旧複製データ・キュー・マネージャー (DR RDQM)

1 つのサイトの 1 次ノードで 1 つのキュー・マネージャーを実行し、別のサイトのリカバリー・ノードでそのキュー・マネージャーの 2 次インスタンスを実行します。1 次インスタンスと 2 次インスタンスの間でデータを複製し、何かの理由で 1 次ノードが失われた場合は、2 次インスタンスを 1 次イン

タンスにして開始します。どちらのノードも Linux サーバーでなければなりません。複製は DRBD によって制御します。

## 複数インスタンス・キュー・マネージャーと HA クラスターとの相違

複数インスタンス・キュー・マネージャーおよび HA クラスターは、キュー・マネージャーの高可用性を達成するための代替方法です。以下は、2つのアプローチの間の相違点を明白にするいくつかのポイントです。

複数インスタンス・キュー・マネージャーには、以下の機能が含まれます。

- IBM MQ に統合された基本的なフェイルオーバー・サポート
- HA クラスターより高速のフェイルオーバー
- 簡単な構成および操作
- IBM MQ Explorer との統合

複数インスタンス・キュー・マネージャーには、以下の制約があります。

- 可用性の高い、ハイパフォーマンスのネットワーク・ストレージが必要とされる
- キュー・マネージャーがフェイルオーバーするときに IP アドレスが変更されるので、ネットワーク構成がより複雑になる

HA クラスターには、以下の機能が含まれます。

- アプリケーション・サーバーやデータベースなど、複数のリソースを調整する機能
- 2つよりも多いノードから成るクラスターを含む、より柔軟な構成オプション
- オペレーターの介入なしで複数回フェイルオーバーできる機能
- フェイルオーバーの一部としてキュー・マネージャーの IP アドレスを引き継ぐ機能

HA クラスターには、以下の制約があります。

- 追加の製品購入およびスキルが必要となる
- クラスターのノード間で切り替え可能なディスクが必要となる
- HA クラスターの構成が比較的複雑
- 従来はフェイルオーバーが比較的到低速でしたが、最近の HA クラスター製品では改善されています
- キュー・マネージャーなどのリソースのモニターに使用されるスクリプトに不備がある場合、不必要なフェイルオーバーが発生することがあります

## HA クラスターとキュー・マネージャー・クラスターとの関係

キュー・マネージャー・クラスターは、キュー・マネージャー・クラスター・キューの使用可能な複数インスタンスに渡るメッセージのロード・バランシングを提供します。この機能は、単一のキュー・マネージャーよりも高い可用性を提供します。なぜなら、キュー・マネージャーに障害が生じた後に、メッセージング・アプリケーションがキュー・マネージャー・クラスター・キューの残りの正常なインスタンスに引き続きメッセージを送ったり、アクセスしたりできるためです。ただし、キュー・マネージャー・クラスターはクラスター内の使用可能なキュー・マネージャーに新規メッセージを自動的にルーティングしますが、使用できないキュー・マネージャーのキューに現時点で入れられているメッセージは、そのキュー・マネージャーを再始動するまで使用できません。そのため、キュー・マネージャー・クラスター自体は、すべてのメッセージ・データの可用性は提供せず、キュー・マネージャーの障害の自動検知、およびキュー・マネージャーの再始動やフェイルオーバーの自動起動の機能もありません。高可用性 (HA) クラスターには、それらの機能があります。これら 2つのタイプのクラスターは、効果的に共有することができます。キュー・マネージャー・クラスターの概要については、[クラスターの設計](#)を参照してください。

Linux

UNIX

## UNIX and Linux での HA クラスター

UNIX and Linux プラットフォーム上の高可用性 (HA) クラスターで IBM MQ を使用できます。例えば、PowerHA for AIX (旧称 HACMP)、Veritas Cluster Server、HP Serviceguard、Red Hat Cluster Suite を備えた Red Hat Enterprise Linux クラスターなどです。

IBM WebSphere MQ 7.0.1 より前には、HA クラスターの構成を支援するために SupportPac MC91 が提供されていました。IBM WebSphere MQ 7.0.1 は、キュー・マネージャーがどこにデータを保管するかに関して、以前のバージョンよりも強力な制御を提供しました。これにより、HA クラスター内でキュー・マネージャーを構成することがより容易になります。SupportPac MC91 で提供されたスクリプトのほとんどは必要なくなり、SupportPac はなくなりました。

このセクションでは、[410 ページの『HA クラスターの構成』](#)、[HA クラスターとキュー・マネージャー・クラスターとの関係](#)、[410 ページの『IBM MQ クライアント』](#)、および [411 ページの『HA クラスター内で作動する IBM MQ』](#) について紹介し、キュー・マネージャーを HA クラスターと共に構成するための手順を示して、改造可能なスクリプトの例を提供します。

このセクションで説明される構成の手順に関する支援として、使用する環境に応じた HA クラスターの文書を参照してください。

## HA クラスターの構成

このセクションでノードという語は、オペレーティング・システムおよび HA ソフトウェアを実行しているエンティティを指して使用されます。このような使用法での同義語としては、「コンピューター」、「システム」、「マシン」、「パーティション」、「ブレード」などがあります。IBM MQ を使用すると、スタンバイ構成またはテークオーバー構成のいずれか (すべてのクラスター・ノードが IBM MQ ワークロードを実行する相互テークオーバーを含む) をセットアップするのに役立ちます。

スタンバイ構成は最も基本的な HA クラスター構成で、1つのノードが作業を実行している間、もう1つのノードはスタンバイとして機能します。スタンバイ・ノードは作業を実行しないので、アイドルと呼ばれます。この構成は、コールド・スタンバイと呼ばれることもあります。そのような構成では、高度のハードウェア冗長度が必要となります。ハードウェアに関して節約するために、この構成を拡張して、単一のスタンバイ・ノードに対して複数の作業ノードを組み合わせることも可能です。この方法のポイントは、スタンバイ・ノードが他のいずれの作業ノードによる作業も引き継ぐことができることです。この構成もスタンバイ構成と呼ばれ、「N+1」構成と呼ばれることもあります。

テークオーバー構成は、すべてのノードが何らかの作業を実行し、ノード障害の際には重要な作業を引き継ぐことが可能な、より高度な構成です。

片側テークオーバー構成は、スタンバイ・ノードが重要ではない、そして移動できない何らかの作業を追加で実行する構成です。この構成はスタンバイ構成と似ていますが、(重要ではない) 作業がスタンバイ・ノードによって実行される点が異なります。

相互テークオーバー構成は、すべてのノードが可用性の高い (移動可能な) 作業を実行する構成です。このタイプの HA クラスター構成は、すべてのノードが重要なワークロードをアクティブに処理していることを示すために、「アクティブ/アクティブ」と呼ばれることもあります。

拡張されたスタンバイ構成またはいずれかのテークオーバー構成では、他のノードの作業をテークオーバーできるノードに負わせる可能性のあるピーク・ロード量を検討することは重要です。そのようなノードは、受容可能なレベルのパフォーマンスを保つために十分な能力を持つ必要があります。

## HA クラスターとキュー・マネージャー・クラスターとの関係

キュー・マネージャー・クラスターは管理作業を減らし、キュー・マネージャー・クラスター・キューの複数インスタンスに渡るメッセージのロード・バランシングを提供します。それらはまた、単一のキュー・マネージャーよりも高い可用性を提供します。キュー・マネージャーに障害が生じた後に、メッセージング・アプリケーションがキュー・マネージャー・クラスター・キューの機能しているインスタンスに引き続きアクセスできるためです。ただし、キュー・マネージャー・クラスター自体には、キュー・マネージャーの障害の自動検知、およびキュー・マネージャーの再始動やフェイルオーバーの自動起動の機能はありません。HA クラスターには、それらの機能があります。これら2つのタイプのクラスターは、効果的に共有することができます。

## IBM MQ クライアント

再始動またはテークオーバーに依存する可能性のあるキュー・マネージャーと通信している IBM MQ クライアントは、接続の失敗に対応できるように、そして再接続を繰り返し試行するように記述する必要があります。IBM WebSphere MQ 7 では、接続の可用性およびワークロード・バランシングを支援するクライ

アント・チャンネル定義テーブル (CCDT) の処理に関する機能が導入されました。ただし、これらはフェイルオーバー・システムに関する作業には直接関係しません。

トランザクション機能により、IBM MQ MQI client は、同じキュー・マネージャーに接続されている限り、2 フェーズ・トランザクションに参加できます。トランザクション機能で、IP ロード・バランサーなどの手法を使用してキュー・マネージャーのリストから選択することはできません。HA 製品を使用する場合、キュー・マネージャーはそれが実行されているノードの ID (名前およびアドレス) を保持するので、HA 制御の下にあるキュー・マネージャーと共にトランザクション機能を使用できます。

## HA クラスター内で作動する IBM MQ

すべての HA クラスターには、フェイルオーバーの単位があります。これは、可用性の高いサービスを形成するすべてのリソースを含む定義のセットです。フェイルオーバーの単位には、サービス自体と、それが依存する他のすべてのリソースが含まれます。

HA ソリューションは、フェイルオーバーの単位に対して異なる用語を使用します。

- PowerHA for AIX では、フェイルオーバーの単位はリソース・グループと呼ばれます。
- Veritas Cluster Server では、サービス・グループと呼ばれます。
- Serviceguard では、パッケージと呼ばれます。

このトピックでは、フェイルオーバーの単位を意味する用語として、リソース・グループを使用します。

IBM MQ のフェイルオーバーの単位として最も小さいものは、キュー・マネージャーです。通常、キュー・マネージャーを含むリソース・グループには、リソース・グループによる使用のために専用予約されている、ボリューム・グループまたはディスク・グループ内の共有ディスクも含まれます。また、キュー・マネージャーへの接続に使用される IP アドレスが含まれます。さらに、同じリソース・グループ内のリスナーまたはトリガー・モニターなど、別のリソースとして、またはキュー・マネージャー自体の制御下で、他の IBM MQ リソースを含めることも可能です。

HA クラスターで使用されるキュー・マネージャーでは、データおよびログがクラスター内のノード間で共有されるディスク上に存在する必要があります。HA クラスターは、一度にクラスター内の 1 つのノードだけがディスクに書き込むことができるようにします。HA クラスターはモニター・スクリプトを使用して、キュー・マネージャーの状態をモニターできます。

キュー・マネージャーに関連したデータおよびログの両方のために、単一の共有ディスクを使用することが可能です。ただし通常は、別々の共有ファイル・システムを使用して、それらを個別にサイズ変更および調整できるようにします。

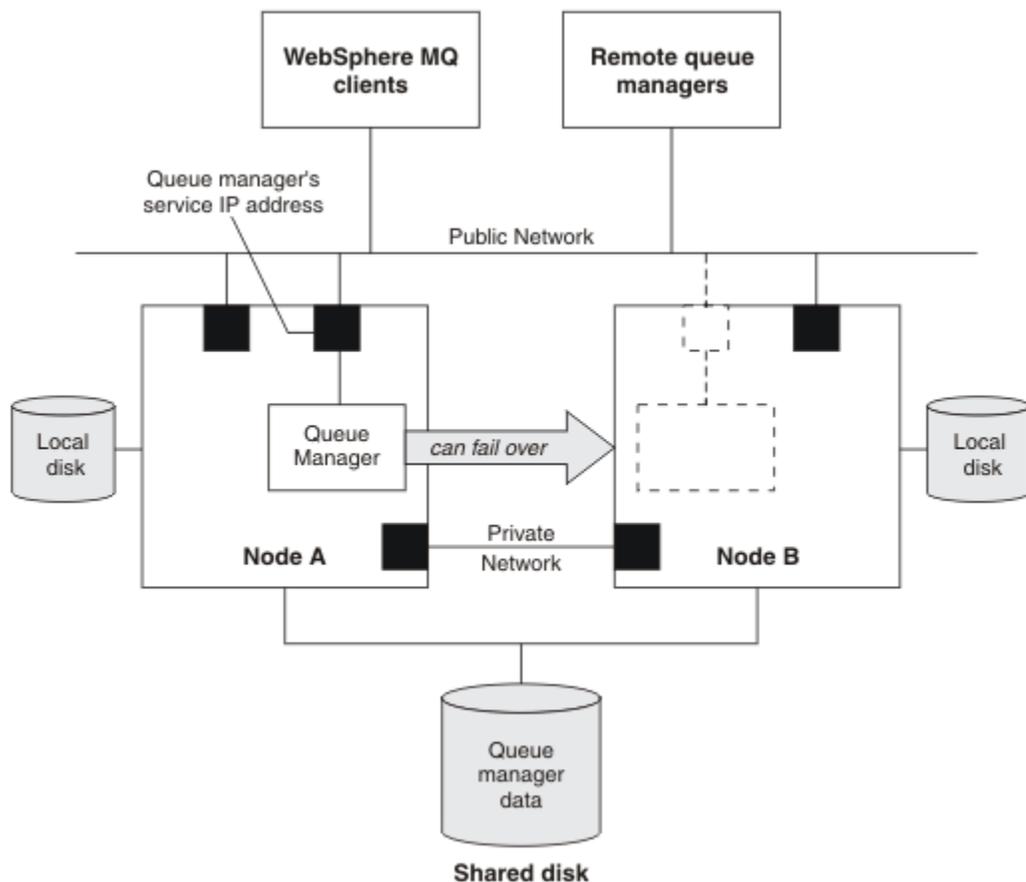


図 71. HA クラスタ

図 1 は、2つのノードのある HA クラスタを示しています。HA クラスタは、リソース・グループ内で定義されたキュー・マネージャーの可用性を管理しています。1つのノード(ノード A)だけが現在キュー・マネージャーを実行しているので、これはアクティブ/パッシブまたはコールド・スタンバイ構成です。キュー・マネージャーは、共有ディスク上のデータおよびログ・ファイルと共に作成されました。キュー・マネージャーには、同じく HA クラスタによって管理されるサービス IP アドレスがあります。キュー・マネージャーは、共有ディスクおよびそのサービス IP アドレスに依存します。HA クラスタがノード A からノード B にキュー・マネージャーをフェイルオーバーするときは、最初にキュー・マネージャーの従属リソースをノード B に移動してから、キュー・マネージャーを始動します。

HA クラスタに複数のキュー・マネージャーが含まれる場合は、その HA クラスタ構成のために、フェイルオーバーの後に同じノード上に複数のキュー・マネージャーが実行する結果になることがあります。HA クラスタ内の各キュー・マネージャーには、独自のポート番号が割り当てられます。これは、いずれかの特定の時点でアクティブになっているクラスタ・ノードで使用されます。

通常、HA クラスタは root ユーザーとして実行されます。IBM MQ は、mqm ユーザーとして実行されます。IBM MQ の管理は、mqm グループのメンバーに認可されます。mqm ユーザーおよびグループの両方が、すべての HA クラスタ・ノード上に存在することを確認してください。ユーザー ID およびグループ ID は、クラスタ全体で一貫性のあるものでなければなりません。root ユーザーが IBM MQ を管理することは許可されていません。スクリプトを開始、停止、またはモニターするスクリプトは、mqm ユーザーに切り替える必要があります。

注: IBM MQ は、すべてのノードに正しくインストールする必要があります。製品の実行可能ファイルを共有することはできません。

### Linux → UNIX 共有ディスクの構成 (UNIX and Linux)

HA クラスタ内の IBM MQ キュー・マネージャーでは、データ・ファイルおよびログ・ファイルが、共有ディスク上で指定された共通のリモート・ファイル・システム内にあることが必要です。

## このタスクについて

図 1 は、HA クラスター内のキュー・マネージャー用に可能なレイアウトを示しています。キュー・マネージャーのデータおよびログ・ディレクトリーは、どちらも /MQHA/QM1 にマウントされる共有ディスク上にあります。フェイルオーバーが生じるとき、このディスクは HA クラスターのノード間で切り替えられるので、キュー・マネージャーをどこで再始動してもデータが使用可能になります。mqm.ini ファイルには、QM1 キュー・マネージャー用のスタanzasがあります。qm.ini ファイルの Log スタanzasには、LogPath の値があります。

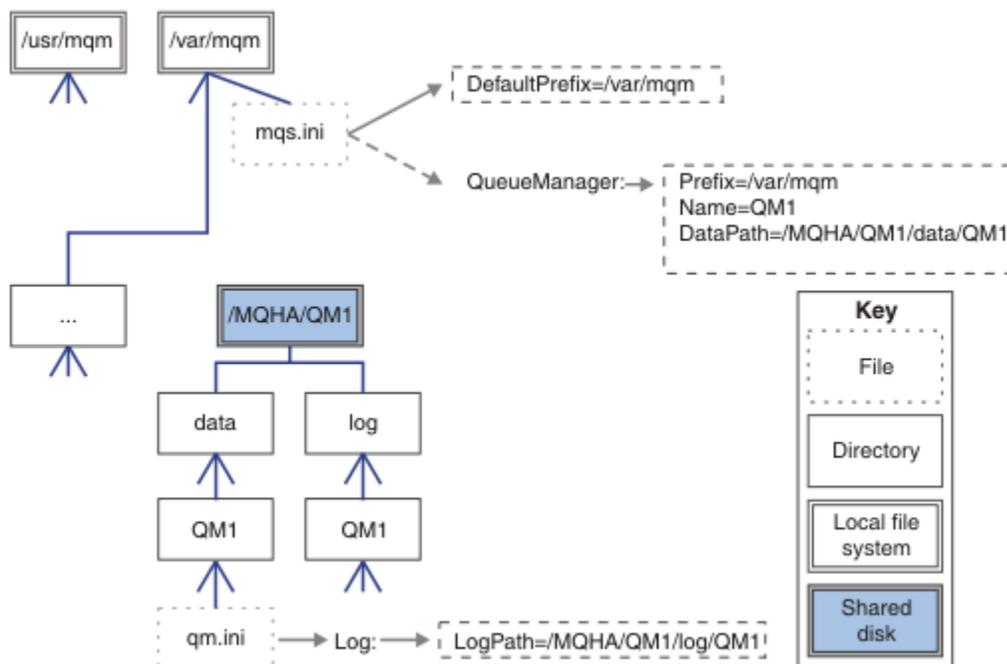


図 72. 共有に指定された data および log ディレクトリー

## 手順

1. キュー・マネージャーのファイル・システム用のマウント・ポイントの名前を決めます。  
例えば、キュー・マネージャーのデータ・ファイル用には /MQHA/qmgrname/data、ログ・ファイル用には /MQHA/qmgrname/log とします。
2. キュー・マネージャーのデータ・ファイルおよびログ・ファイルを入れるためのボリューム・グループ (またはディスク・グループ) を作成します。  
このボリューム・グループは、キュー・マネージャーと同じリソース・グループ内の高可用性 (HA) クラスターによって管理されます。
3. キュー・マネージャーのデータ・ファイルおよびログ・ファイル用のファイル・システムをボリューム・グループ内に作成します。
4. 次にノードごとに、ファイル・システム用のマウント・ポイントを作成して、ファイル・システムがマウント可能であることを確認します。  
mqm ユーザーはマウント・ポイントを所有する必要があります。

Linux

UNIX

## HA クラスター・キュー・マネージャーの作成 (UNIX and Linux)

キュー・マネージャーを高可用性クラスターで使用するための最初のステップは、ノードの 1 つにキュー・マネージャーを作成することです。

## このタスクについて

HA クラスターで使用するためのキュー・マネージャーを作成するには、まずキュー・マネージャーを作成するクラスター内のノードを 1 つ選択し、次にそのノードで以下の手順を実行する必要があります。

## 手順

1. キュー・マネージャーのファイル・システムをノードにマウントします。
2. **crtmqm** コマンドを使用してキュー・マネージャーを作成します。  
以下に例を示します。  

```
crtmqm -md /MQHA/qmgrname/data -ld /MQHA/qmgrname/log qmgrname
```
3. **strmqm** コマンドを使用して、手動でキュー・マネージャーを開始します。
4. キューとチャンネルの作成、およびキュー・マネージャーの始動時にリスナーが自動的に開始するようにキュー・マネージャーを設定するなど、キュー・マネージャーの初期構成を完了します。
5. **endmqm** コマンドを使用してキュー・マネージャーを停止させます。
6. 以下のように **dspmqinf** コマンドを使用して、**addmqinf** コマンドを表示します。

```
dspmqinf -o command qmgrname
```

ここで、qmgrname は、キュー・マネージャーの名前です。

**addmqinf** コマンドの使用について詳しくは、[414 ページの『他の HA クラスタ・ノードへのキュー・マネージャー構成の追加 \(UNIX and Linux\)』](#)を参照してください。

**addmqinf** コマンドは、以下の例のように表示されます。

```
addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \  
-vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
```

7. 表示されるコマンドを注意深く記録してください。
8. キュー・マネージャーのファイル・システムをアンマウントします。

## 次のタスク

これで、[414 ページの『他の HA クラスタ・ノードへのキュー・マネージャー構成の追加 \(UNIX and Linux\)』](#)で説明されているステップを完了する準備ができました。

### Linux → UNIX 別の HA クラスタ・ノードへのキュー・マネージャー構成の追加 (UNIX and Linux)

キュー・マネージャーの構成情報を HA クラスタ内の他のノードに追加する必要があります。

## 始める前に

この作業を完了する前に、[413 ページの『HA クラスタ・キュー・マネージャーの作成 \(UNIX and Linux\)』](#)のステップを完了する必要があります。キュー・マネージャーを作成したら、他の各ノードで以下の手順を実行して、HA クラスタ内の他のノードのそれぞれにキュー・マネージャーの構成情報を追加する必要があります。

## このタスクについて

HA クラスタで使用するためのキュー・マネージャーを作成する場合は、まず、キュー・マネージャーを作成するクラスタ内のノードを 1 つ選択する必要があります (詳しくは、[413 ページの『HA クラスタ・キュー・マネージャーの作成 \(UNIX and Linux\)』](#)を参照)。

## 手順

1. キュー・マネージャーのファイル・システムをマウントする。
2. ノードにキュー・マネージャー構成情報を追加する。  
構成情報を追加するには、以下の 2 つの方法があります。
  - /var/mqm/mqs.ini を直接編集する。

- 413 ページの『HA クラスター・キュー・マネージャーの作成 (UNIX and Linux)』の手順 6 の **dspmqinf** コマンドで表示された **addmqinf** コマンドを実行する。
3. キュー・マネージャーを開始してから停止して、構成を検査する。  
キュー・マネージャーの開始および停止に使用されるコマンドは、**addmqinf** コマンドと同じ IBM MQ のインストール済み環境から発行される必要があります。キュー・マネージャーに現在関連付けられているものとは別のインストール済み環境からキュー・マネージャーを開始および停止するには、まず、**setmqm** コマンドを使用して、インストール済み環境にキュー・マネージャーとの関連付けを設定する必要があります。詳細については、[setmqm](#) を参照してください。
  4. キュー・マネージャーのファイル・システムをアンマウントする。

## Linux → UNIX HA クラスター・キュー・マネージャーの開始 (UNIX and Linux)

キュー・マネージャーは、HA クラスター内でリソースとして表わされます。HA クラスターは、キュー・マネージャーを始動および停止できることが必要です。ほとんどの場合、キュー・マネージャーを始動するためにシェル・スクリプトを使用できます。これらのスクリプトは、ネットワーク・ファイル・システムを使用するかまたは各ローカル・ディスクにコピーすることにより、クラスター内のすべてのノードの同じ場所で使用可能にする必要があります。

**注:** 失敗したキュー・マネージャーを再始動する前に、アプリケーションをキュー・マネージャーのそのインスタンスから切断する必要があります。そうしない場合、キュー・マネージャーは正しく再始動されない場合があります。

適切なシェル・スクリプトの例をここに示します。これらを必要に合うように作り替えて、HA クラスターの制御下でキュー・マネージャーを始動するために使用できます。

以下のシェル・スクリプトは、HA クラスター・ユーザーから mqm ユーザーに切り替えて、キュー・マネージャーを正常に始動できるようにする方法を示す例です。

```
#!/bin/ksh
# A simple wrapper script to switch to the mqm user.
su mqm -c name_of_your_script $*
```

以下のシェル・スクリプトは、キュー・マネージャーの現在の状態について何も想定せずに、キュー・マネージャーを始動する方法を示す例です。キュー・マネージャーに属するすべてのプロセスを突然に終了させる方法が使用されていることに注意してください。

```
#!/bin/ksh
#
# This script robustly starts the queue manager.
#
# The script must be run by the mqm user.
#
# The only argument is the queue manager name. Save it as QM variable
QM=$1

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

# End any queue manager processes which might be running.

srchstr="(|-m)$QM *.*$"
for process in amqzmuc0 amqzma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqslr amqcrista amqirmfa amqrmppa \
               amqzfuma amqzmuf0 amqzmur0 amqzmgr0
do
  ps -ef | tr "\t" " " | grep $process | grep -v grep | \
  egrep "$srchstr" | awk '{print $2}' | \
  xargs kill -9 > /dev/null 2>&1
done

# It is now safe to start the queue manager.
```

```
# The strmqm command does not use the -x flag.
strmqm ${QM}
```

このスクリプトを変更して、関連する他のプログラムを開始することができます。

## Linux → UNIX HA クラスタ・キュー・マネージャの停止 (UNIX and Linux)

ほとんどの場合、キュー・マネージャを停止するためにシェル・スクリプトを使用できます。適切なシェル・スクリプトの例をここに示します。これらを必要に合うように作り替えて、HA クラスタの制御下でキュー・マネージャを停止するために使用できます。

以下のスクリプトは、キュー・マネージャの現在の状態について何も想定せずに、キュー・マネージャを即時に停止する方法を示す例です。スクリプトは mqm ユーザーが実行する必要があります。そのため、このスクリプトをシェル・スクリプト内にラップして、ユーザーを HA クラスタ・ユーザーから mqm に切り替えることが必要な場合があります (シェル・スクリプトの例は [415 ページの『HA クラスタ・キュー・マネージャの開始 \(UNIX and Linux\)』](#) に記載されています)。

```
#!/bin/ksh
#
# The script ends the QM by using two phases, initially trying an immediate
# end with a time-out and escalating to a forced stop of remaining
# processes.
#
# The script must be run by the mqm user.
#
# There are two arguments: the queue manager name and a timeout value.
QM=$1
TIMEOUT=$2

if [ -z "$QM" ]
then
    echo "ERROR! No queue manager name supplied"
    exit 1
fi

if [ -z "$TIMEOUT" ]
then
    echo "ERROR! No timeout specified"
    exit 1
fi

for severity in immediate brutal
do
    # End the queue manager in the background to avoid
    # it blocking indefinitely. Run the TIMEOUT timer
    # at the same time to interrupt the attempt, and try a
    # more forceful version. If the brutal version fails,
    # nothing more can be done here.

    echo "Attempting ${severity} end of queue manager '${QM}'"
    case $severity in

    immediate)
        # Minimum severity of endmqm is immediate which severs connections.
        # HA cluster should not be delayed by clients
        endmqm -i ${QM} &
        ;;

    brutal)
        # This is a forced means of stopping queue manager processes.

        srchstr="(|-m)$QM *.*$"
        for process in amqzmuc0 amqzma0 amqfcxba amqfqpud amqpcsea amqzlaa0 \
            amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmf0 amqrmppa \
            amqzfuma amqzmuf0 amqzmur0 amqzmgr0
        do
            ps -ef | tr "\t" " " | grep $process | grep -v grep | \
                egrep "$srchstr" | awk '{print $2}' | \
                xargs kill -9 > /dev/null 2>&1
        done

        esac

        TIMED_OUT=yes
        SECONDS=0
        while (( $SECONDS < ${TIMEOUT} ))
```

```

do
  TIMED_OUT=yes
  i=0
  while [ $i -lt 5 ]
  do
    # Check for execution controller termination
    srchstr="( |-m)$QM *.*$"
    cnt=`ps -ef | tr "\t" " " | grep amqzma0 | grep -v grep | \
    egrep "$srchstr" | awk '{print $2}' | wc -l`
    i=`expr $i + 1`
    sleep 1
    if [ $cnt -eq 0 ]
    then
      TIMED_OUT=no
      break
    fi
  done

  if [ ${TIMED_OUT} = "no" ]
  then
    break
  fi

  echo "Waiting for ${severity} end of queue manager '${QM}'"
  sleep 1
done # timeout loop

if [ ${TIMED_OUT} = "yes" ]
then
  continue      # to next level of urgency
else
  break         # queue manager is ended, job is done
fi

done # next phase

```

注: 特定のキュー・マネージャーに対して実行されているプロセスによっては、このスクリプトに含まれるキュー・マネージャー・プロセスのリストが完全ではなかったり、対象キュー・マネージャーで実行されているプロセス以外のプロセスが含まれたりする可能性があります。

```

for process in amqzmuc0 amqzma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrsta amqirmfa amqrmppa \
               amqzfuma amqmuf0 amqzmur0 amqzmgr0

```

構成されている機能と、特定のキュー・マネージャーで実行されているプロセスに基づいて、プロセスをこのリストに包含したり、このリストから除外したりできます。プロセスの完全な一覧と、特定の順序でプロセスを停止する方法については、[UNIX および Linux におけるキュー・マネージャーの手動停止](#)を参照してください。

Linux

UNIX

## HA クラスター・キュー・マネージャーのモニター (UNIX and

### Linux)

通常、高可用性 (HA) クラスターがキュー・マネージャーの状態を定期的にモニターするための手段を提供します。ほとんどの場合、このためにシェル・スクリプトを使用できます。適切なシェル・スクリプトの例をここに示します。これらのスクリプトを必要に合うように作り替えて、環境に特定の追加のモニター検査を行うために使用できます。

IBM WebSphere MQ 7.1 から、IBM MQ の複数のインストール環境が 1 つのシステムで共存できるようになりました。複数のインストール環境について詳しくは、[複数のインストール](#)を参照してください。IBM WebSphere MQ 7.1 以上のインストール環境も含めて、複数のインストール環境を対象にしてモニター・スクリプトを使用する場合は、追加の手順が必要になる可能性があります。プライマリー・インストール環境がある場合や、IBM WebSphere MQ 7.1 より前のバージョンでスクリプトを使用する場合は、`MQ_INSTALLATION_PATH` を指定してスクリプトを使用する必要はありません。そうでない場合は、`MQ_INSTALLATION_PATH` を正しく指定するために以下の手順を実行してください。

1. IBM WebSphere MQ 7.1 のインストール環境から `crtmqenv` コマンドを使用して、キュー・マネージャーの正しい `MQ_INSTALLATION_PATH` を確認します。

```

crtmqenv -m qmname

```

このコマンドによって、*qmname* で指定したキュー・マネージャーの正しい *MQ\_INSTALLATION\_PATH* の値が返されます。

2. 該当する *qmname* パラメーターと *MQ\_INSTALLATION\_PATH* パラメーターを使用して、モニター・スクリプトを実行します。

**注:** PowerHA for AIX では、モニター・プログラムにキュー・マネージャー用のパラメーターを指定する手段が提供されていません。キュー・マネージャーごとに、キュー・マネージャー名をカプセル化する別個のモニター・プログラムを作成する必要があります。キュー・マネージャー名をカプセル化するための、AIX 上で使用するスクリプトを以下に示します。

```
#!/bin/ksh
su mqm -c name_of_monitoring_script qmname MQ_INSTALLATION_PATH
```

*MQ\_INSTALLATION\_PATH* は、キュー・マネージャー *qmname* が関連付けられている IBM MQ のインストール環境のパスを指定するオプション・パラメーターです。

下記のスクリプトは、**runmqsc** がハングする可能性に対して耐久力がありません。通常、HA クラスタはハングしたモニター・スクリプトを障害として処理するので、それ自体はこの可能性に対して耐久力があります。

しかしこのスクリプトは、キュー・マネージャーが開始状態にあることに関しては容認します。それは、HA クラスタがキュー・マネージャーを開始した直後に、そのモニターを開始することが一般的であるためです。一部の HA クラスタは、リソースの開始フェーズと実行フェーズとを区別しますが、開始フェーズの所要時間を構成することが必要です。キュー・マネージャーの開始に必要な時間はそれが行う作業の量に依存するので、キュー・マネージャーの開始に必要な最大時間を選択することは困難です。小さ過ぎる値を選択すると、キュー・マネージャーが開始処理をまだ完了していないだけの場合でも、障害が発生したと、HA クラスタが間違って判断してしまいます。これにより、フェイルオーバーが無限に続く結果になる可能性があります。

スクリプトは、mqm ユーザーによって実行される必要があります。そのため、このスクリプトをシェル・スクリプト内にラップして、ユーザーを HA クラスタ・ユーザーから mqm に切り替えることが必要な場合があります (シェル・スクリプトの例は、[415 ページ](#)の『[HA クラスタ・キュー・マネージャーの開始 \(UNIX and Linux\)](#)』にあります)。

```
#!/bin/ksh
#
# This script tests the operation of the queue manager.
#
# An exit code is generated by the runmqsc command:
# 0 => Either the queue manager is starting or the queue manager is running and responds.
#     Either is OK.
# >0 => The queue manager is not responding and not starting.
#
# This script must be run by the mqm user.
QM=$1
MQ_INSTALLATION_PATH=$2

if [ -z "$QM" ]
then
    echo "ERROR! No queue manager name supplied"
    exit 1
fi

if [ -z "$MQ_INSTALLATION_PATH" ]
then
    # No path specified, assume system primary install or MQ level < 7.1.0.0
    echo "INFO: Using shell default value for MQ_INSTALLATION_PATH"
else
    echo "INFO: Prefixing shell PATH variable with $MQ_INSTALLATION_PATH/bin"
    PATH=$MQ_INSTALLATION_PATH/bin:$PATH
fi

# Test the operation of the queue manager. Result is 0 on success, non-zero on error.
echo "ping qmgr" | runmqsc ${QM} > /dev/null 2>&1
pingresult=$?

if [ $pingresult -eq 0 ]
then # ping succeeded
```

```

echo "Queue manager '${QM}' is responsive"
result=0

else # ping failed

# Don't condemn the queue manager immediately, it might be starting.
srchstr="(|-m)$QM *.*$"
cnt=`ps -ef | tr "\t" " " | grep strmqm | grep "$srchstr" | grep -v grep \
| awk '{print $2}' | wc -l`
if [ $cnt -gt 0 ]
then
# It appears that the queue manager is still starting up, tolerate
echo "Queue manager '${QM}' is starting"
result=0
else
# There is no sign of the queue manager starting
echo "Queue manager '${QM}' is not responsive"
result=$pingresult
fi

fi

exit $result

```

## Linux → UNIX キュー・マネージャーを HA クラスターの制御下に置く方法

### (UNIX and Linux)

キュー・マネージャーを、HA クラスターの制御下で、キュー・マネージャーの IP アドレスおよび共有ディスクと共に構成する必要があります。

### このタスクについて

キュー・マネージャーを HA クラスターの制御下に置くには、キュー・マネージャーおよび関連付けられたすべてのリソースを含めるためのリソース・グループを定義する必要があります。

### 手順

1. キュー・マネージャー、キュー・マネージャーのボリュームまたはディスク・グループ、およびキュー・マネージャーの IP アドレスを含むリソース・グループを作成します。  
IP アドレスは仮想 IP アドレスであり、コンピューターの IP アドレスではありません。
2. HA クラスターがクラスター・ノード間でリソースを適正に切り替えて、キュー・マネージャーを制御する準備ができたことを確認してください。

## Linux → UNIX HA クラスター・キュー・マネージャーの削除 (UNIX and Linux)

キュー・マネージャーの実行が必要ではなくなったノードからキュー・マネージャーを除去できます。

### このタスクについて

HA クラスター内のノードからキュー・マネージャーを除去するには、その構成情報を除去する必要があります。

### 手順

1. HA クラスターからノードを除去して、HA クラスターがこのノードでのキュー・マネージャーのアクティブ化を試行しなくなるようにします。
2. 以下の **rmvmqinf** コマンドを使用して、キュー・マネージャーの構成情報を除去します。

```
rmvmqinf qmgrname
```

3. オプション: キュー・マネージャーを完全に削除するには、**dltmqm** コマンドを使用します。

**重要:** **dltmqm** コマンドを使用してキュー・マネージャーを削除すると、そのキュー・マネージャーのデータおよびログ・ファイルが完全に削除されることに注意してください。

キュー・マネージャーを削除した後に、`rmvmqinf` コマンドを使用して他のノードから残りの構成情報を除去することができます。

## Windows Microsoft Cluster Service (MSCS) のサポート

仮想サーバーのフェイルオーバーをサポートする MSCS の紹介およびセットアップ。

この情報は IBM MQ for Windows のみに適用されます。

Microsoft Cluster Service (MSCS) を使用すると、サーバーをクラスターに接続して、データおよびアプリケーションにさらに高い可用性を提供し、システムの管理を容易にすることができます。MSCS は、サーバーまたはアプリケーション障害を自動的に検出し、リカバリーすることができます。

MSCS は仮想サーバーのフェイルオーバーをサポートします。この仮想サーバーは、アプリケーション、Web サイト、印刷キュー、またはファイル共有 (例えば、ディスク・スピンドル、ファイル、IP アドレスを含む) に対応します。

フェイルオーバーとは、MSCS がクラスター内のあるコンピューターのアプリケーションで障害を検出したら、障害を起こしたアプリケーションを通常の方法でシャットダウンし、その状態データを他のコンピューターに転送し、そこでアプリケーションを再開するというプロセスです。

このセクションでは、以下の各項目で MSCS クラスターを紹介し、MSCS サポートのセットアップについて説明します。

- [420 ページの『MSCS クラスターの紹介』](#)
- [421 ページの『MSCS クラスター化用の IBM MQ のセットアップ』](#)

続いて、以下の各セクションで MSCS クラスター化用に IBM MQ を構成する方法を説明します。

- [423 ページの『MSCS で使用するようキュー・マネージャーを作成する』](#)
- [424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#)
- [426 ページの『MSCS 制御下にキュー・マネージャーを置く』](#)
- [432 ページの『MSCS 制御からキュー・マネージャーを除去する』](#)

さらに、以下の各項目で、IBM MQ で MSCS を使用するうえで役立つヒントを挙げ、IBM MQ の MSCS サポート・ユーティリティー・プログラムについて詳しく説明します。

- [433 ページの『MSCS の使用に関するヒント』](#)
- [436 ページの『MSCS ユーティリティー・プログラムのサポート』](#)

## Windows MSCS クラスターの紹介

MSCS クラスターは、相互接続された複数のコンピューターのグループのことで、いずれかのコンピューターに障害が起きた時に、MSCS によってフェイルオーバーが実行されて、アプリケーションの状態データが、障害が起きたコンピューターからクラスター内の別のコンピューターへ転送され、そこで操作が再開されるように構成されています。

[408 ページの『高可用性の構成』](#)には、MSCS クラスター、複数インスタンス・キュー・マネージャー、および IBM MQ クラスターの間の比較が含まれています。

このセクションおよび従属するトピックで、用語クラスターが単独で使用されるときには、常に MSCS クラスターを意味しています。これは、このガイドの他の場所で説明されている IBM MQ クラスターとは異なります。

2 台のマシンのクラスターは、仮想 IP アドレスを使用して、クライアント・アクセス用のネットワークと一緒に接続される 2 台のコンピューター (例えば、A と B) で構成されます。さらに、1 つ以上の専用ネットワークによって相互に接続されていることもあります。A および B は、使用するそれぞれのコンピューター上で、サーバー・アプリケーション用に少なくとも 1 つのディスクを共有します。また、MSCS が排他的に使用するための別の共有ディスクがあります。これは独立ディスク (RAID) レベル 1 の冗長配列でなければなりません。このディスクを *quorum* ディスクと言います。MSCS は、両方のコンピューターをモニターして、ハードウェアとソフトウェアが正しく実行していることを確認します。

このような単純なセットアップでは、どちらのコンピューターにもすべてのアプリケーションがインストールされますが、ライブ・アプリケーションはコンピューター A でのみ実行され、コンピューター B は稼

働しても待ち状態になります。コンピューター A に何らかの問題が発生すると、障害が起きたアプリケーションは MSCS によって通常の方法でシャットダウンされ、他のコンピューターにその状態データが転送され、アプリケーションはそこで再開されます。これがフェイルオーバーです。アプリケーションをクラスター指向にしておく、MSCS との対話が十分に行われ、フェイルオーバーの機能が高まります。

2 台のコンピューターのクラスター用の一般的なセットアップを [421 ページの図 73](#) に示します。

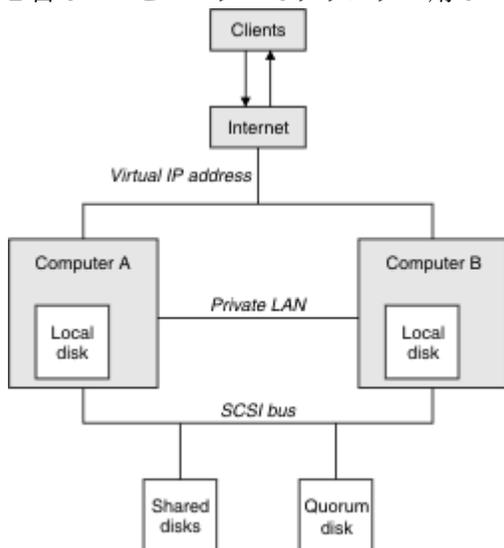


図 73. 2 台のコンピューターにおける MSCS クラスター

それぞれのコンピューターが共有ディスクにアクセスできますが、MSCS 制御の下では、共有ディスクにアクセスできるのは一度に 1 台のコンピューターだけです。フェイルオーバーの際、MSCS はアクセスを他のコンピューターに切り替えます。共有ディスク自体は通常 RAID ですが、それ以外でも構いません。

各コンピューターは、クライアント・アクセス用に外部ネットワークに接続され、それぞれが IP アドレスを持っています。ただし、このクラスターと通信する外部クライアントは、1 つの仮想 IP アドレスだけを認識し、MSCS がクラスター内で IP トラフィックの経路を適切に定めます。

また、MSCS は、1 つ以上の専用接続、または公衆ネットワークを介して 2 台のコンピューター間で独自の通信を実行し、例えば、ハートビートを使用して 2 台のコンピューターの状態をモニターしたり、データベースを同期したりします。

### Windows MSCS クラスター化用の IBM MQ のセットアップ

キュー・マネージャーを MSCS へのフェイルオーバーの単位にすることにより、クラスター化用に IBM MQ を構成します。キュー・マネージャーを MSCS へのリソースとして定義すると、キュー・マネージャーをモニターし、問題がある場合にはクラスター内の別のコンピューターにキュー・マネージャーを転送できます。

システムに合わせて設定するには、クラスター内の各コンピューター上に IBM MQ をインストールすることから始めます。

キュー・マネージャーは IBM MQ インストール名と関連付けられるため、クラスター内のすべてのコンピューター上で IBM MQ インストール名が同じでなければなりません。 [インストールおよびアンインストール](#) を参照してください。

キュー・マネージャー自体は、それらが作成されるコンピューター上にのみ存在する必要があります。フェイルオーバーの際、MSCS は他のコンピューターでキュー・マネージャーを始動します。ただし、キュー・マネージャーは、ローカル・ドライブ上ではなくクラスター共有ディスク上にログおよびデータ・ファイルを持つ必要があります。キュー・マネージャーが既にローカル・ドライブにインストールされている場合、IBM MQ で提供されているツールを使用してこれを移行することができます。 [424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#) を参照してください。MSCS で使用する新しいキュー・マネージャーを作成するには、 [423 ページの『MSCS で使用するよう](#)

インストールおよび移行後、MSCS クラスター・アドミニストレーターを使用して MSCS にキュー・マネージャーを認識させます。[426 ページの『MSCS 制御下にキュー・マネージャーを置く』](#)を参照してください。

MSCS 制御からキュー・マネージャーを除去する場合、[432 ページの『MSCS 制御からキュー・マネージャーを除去する』](#)で説明されている手順に従ってください。

#### **Windows** セットアップの対称性と MSCS

アプリケーションがあるノードから別のノードに切り替わる時、ノードに関係なく同じ方法で動作する必要があります。これを確実にする最善の方法は、環境を同一にすることです。

可能であれば、各コンピューター上で同一のハードウェア、オペレーティング・システム・ソフトウェア、製品ソフトウェア、および構成でクラスターをセットアップします。特に、2 台のコンピューターにインストールされた必須ソフトウェアがすべて、バージョン、保守レベル、SupportPacs、パスおよび出口の点で同一であるように、さらに [422 ページの『MSCS のセキュリティ』](#)で説明されているとおり共通のネーム・スペース (セキュリティ環境) が存在するようにしてください。

#### **Windows** MSCS のセキュリティ

MSCS セキュリティを確実にするために、次のガイドラインに従ってください。

ガイドラインは次のとおりです。

- クラスター内の各コンピューターに同一のソフトウェアがインストールされていることを確認してください。
- クラスター間で共通のネーム・スペース (セキュリティ環境) を作成します。
- MSCS クラスターのノードをドメインのメンバーにし、そのドメイン内ではクラスター所有者のユーザー・アカウントをそのドメイン・アカウントにします。
- クラスターのその他のユーザー・アカウントもドメイン・アカウントにして、それらのユーザー・アカウントを両方のノードで使用できるようにします。既にドメインがあり、IBM MQ に関するアカウントがドメイン・アカウントになっている場合は、これが自動的に当てはまります。現在ドメインがない場合、クラスター・ノードおよび関連するアカウントを提供するために、小さなドメインをセットアップすることを検討してください。その目的は、2 台のコンピューターのクラスターを単一のコンピューター・リソースのようにすることです。

あるコンピューターに対してローカルのアカウントが、もう 1 台のコンピューター上には存在しないということを覚えておってください。もう 1 台のコンピューター上に同じ名前で作成しても、そのセキュリティ ID (SID) が異なるため、アプリケーションが別のノードに移動したとき、そのノードでの許可はありません。

フェイルオーバーまたは移動時に、IBM MQ MSCS サポートにより、キュー・マネージャー・オブジェクトを収めたどのファイルも必ず宛先ノードに対して同等の許可を持つようになります。明示的には、コードにより、Administrators グループと mqm グループ、および SYSTEM アカウントが完全な制御を持っていること、また古いノードですべてのユーザーに読み取りアクセス権があった場合には、宛先ノードで許可が追加されることが確認されます。

ドメイン・アカウントを使用して IBM MQ サービスを実行できます。クラスター内の各コンピューター上のローカル mqm グループにそのサービスがあることを確認してください。

#### **Windows** 複数のキュー・マネージャーを MSCS と共に使用する

1 台のコンピューター上で複数のキュー・マネージャーを実行する場合、以下のセットアップのいずれかを選択できます。

セットアップは次のとおりです。

- すべてのキュー・マネージャーが 1 つのグループで。この構成では、いずれかのキュー・マネージャーで問題が発生した場合、グループ内のすべてのキュー・マネージャーがグループとして他のコンピューターにフェイルオーバーします。
- 各グループのキュー・マネージャーが単独で。この構成では、あるキュー・マネージャーに問題が発生した場合、他のキュー・マネージャーに影響を与えずに、そのキュー・マネージャーだけが他のコンピューターにフェイルオーバーします。

- 上記2つのセットアップの混合。

## Windows クラスタ・モードと MSCS

Windows 上の IBM MQ でクラスタ・システムを実行する場合は、アクティブ/パッシブとアクティブ/アクティブという2つのモードがあります。

注：MSCS を Microsoft Transaction Server (COM+) で使用する場合は、Active/Active モードは使用できません。

### アクティブ/パッシブ・モード

アクティブ/パッシブ・モードでは、コンピューター A 上でアプリケーションが実行され、コンピューター B はバックアップ専用で、MSCS が障害を検出した場合のみに使用されます。

このモードは、1つの共有ディスクでのみ使用できますが、いずれかのアプリケーションがフェイルオーバーを引き起こした場合、すべてのアプリケーションをグループとして転送する必要があります (共有ディスクにアクセスできるのは一度に1つのコンピューターのみであるためです)。

MSCS を構成する際に、A を優先コンピューターとすることができます。これを行うと、コンピューター A が修理されるかまたは取り替えられて、再び正常に機能すると、MSCS がこれを検出してアプリケーションを自動的にコンピューター A に切り替えます。

複数のキュー・マネージャーを実行する場合、キュー・マネージャーごとに個別の共有ディスクを持つことを考慮してください。さらに、各キュー・マネージャーを MSCS 内の別々のグループに設置してください。これで、あるキュー・マネージャーが他のキュー・マネージャーに影響を与えずに、別のコンピューターにフェイルオーバーすることができます。

### アクティブ/アクティブ・モード

アクティブ/アクティブ・モードでは、コンピューター A とコンピューター B の両方でアプリケーションが実行され、各コンピューター上のグループは一方のコンピューターをバックアップとして使用するよう設定されます。コンピューター A 上で障害が検出された場合、MSCS は状態データをコンピューター B に転送し、そこでアプリケーションを再開します。コンピューター B は独自のアプリケーションと A のアプリケーションを実行します。

このセットアップでは、少なくとも2つの共用ディスクが必要です。MSCS を構成する際に、A を A のアプリケーションの優先コンピューター、B を B のアプリケーションの優先コンピューターとして設定できます。フェイルオーバーおよび修理後に、各アプリケーションは自動的に元のコンピューターに戻ります。

IBM MQ では、これは、例えば2つのキュー・マネージャー (コンピューター A 上とコンピューター B 上でそれぞれ1つずつ) が実行されており、それぞれのキュー・マネージャーが自分のコンピューターの完全な能力を活用できることを意味します。コンピューター A で障害が発生すると、両方のキュー・マネージャーがコンピューター B で稼働します。これは、一台のコンピューターの能力を共有することで、大量のデータを高速で処理する能力を低下させます。ただし、A の障害を見つけて修理する間も、重要なアプリケーションは依然として使用可能になっています。

## Windows MSCS で使用するようにキュー・マネージャーを作成する

この手順では、キュー・マネージャーを MSCS の制御下に置くための準備をし、それに適した方法で新しいキュー・マネージャーを作成します。

まず、キュー・マネージャーとそのすべてのリソースをローカル・ドライブ上に作成してから、ログ・ファイルとデータ・ファイルを共有ディスクに移行します。(この操作は逆にしても構いません。) キュー・マネージャーとそのリソースを共有ドライブ上に作成しないでください。

MSCS で使用するようにキュー・マネージャーを作成するには、コマンド・プロンプトから作成するか、または IBM MQ Explorer で作成するかの2つの方法があります。コマンド・プロンプトを使用する利点は、キュー・マネージャーが停止状態で作成され、手動による始動に設定されて MSCS の準備が整う点です。(IBM MQ Explorer の場合、自動的に新しいキュー・マネージャーが始動し、作成後は自動的に始動するように設定します。これを変更する必要があります。)

## コマンド・プロンプトからキュー・マネージャーを作成する

MSCS で使用するよう、コマンド・プロンプトからキュー・マネージャーを作成するには、以下のステップに従います。

1. 環境変数 MQSPREFIX がローカル・ドライブ (例えば、C:\IBM MQ) を参照するように設定されていることを確認します。これを変更した場合、システム・アカウントが変更を適用するよう、マシンをリブートしてください。変数を設定しない場合、キュー・マネージャーは、キュー・マネージャーの IBM MQ デフォルト・ディレクトリで作成されます。
2. **crtmqm** コマンドを使用してキュー・マネージャーを作成します。例えば、デフォルト・ディレクトリに `mscs_test` というキュー・マネージャーを作成するには、次のように入力します。

```
crtmqm mscs_test
```

3. [424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#)に進みます。

## IBM MQ Explorer を使用してキュー・マネージャーを作成する

MSCS で使用するよう、IBM MQ Explorer を使用してキュー・マネージャーを作成するには、以下のステップに従います。

1. 「スタート」メニューから IBM MQ Explorer を始動します。
2. ナビゲーター・ビューで、ツリー・ノードを展開して、「Queue Managers (キュー・マネージャー)」ツリー・ノードを探します。
3. 「キュー・マネージャー」ツリー・ノードを右クリックし、「新規」>「キュー・マネージャー」を選択します。「Create Queue Manager (キュー・マネージャーの作成)」パネルが表示されます。
4. ダイアログを完成 (ステップ 1) してから、「次へ>」をクリックします。
5. ダイアログを完成 (ステップ 2) してから、「次へ>」をクリックします。
6. ダイアログを完成 (ステップ 3) してから、「キュー・マネージャーの開始」および「サーバー接続チャネルの作成」が選択されていないことを確認し、「次へ>」をクリックします。
7. ダイアログを完成 (ステップ 4) してから、「Finish (完了)」をクリックします。
8. [424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#)に進みます。

### Windows キュー・マネージャーを MSCS ストレージに移動する

この手順に従えば、既存のキュー・マネージャーが、MSCS 制御下に置くのに適した状態になるように構成されます。

この操作を行うには、ログ・ファイルとデータ・ファイルを共有ディスクに移動して、障害が起きた場合に他のコンピューターで使用できるようにしておく必要があります。例えば、既存のキュー・マネージャーには、C:\WebSphere MQ\log\QMname や C:\WebSphere MQ\qmgrs\QMname などのパスが存在する場合があります。



**重要:** ファイルは手動で移動せずに、このトピックで説明されているように、IBM MQ MSCS サポートの一部として提供されるユーティリティー・プログラムを使用してください。

移動するキュー・マネージャーが TLS 接続を使用し、その TLS のキー・リポジトリがローカル・マシン上のキュー・マネージャーのデータ・ディレクトリ内にある場合は、そのキー・リポジトリは、他のキュー・マネージャーによって共有ディスクに移動されます。デフォルトでは、TLS 鍵リポジトリの場所を指定するキュー・マネージャー属性 SSLKEYR は、キュー・マネージャー・データ・ディレクトリの下にある `MQ_INSTALLATION_PATH\qmgrs\QMGRNAME\ssl\key` に設定されます。

`MQ_INSTALLATION_PATH` は、IBM MQ がインストールされている上位ディレクトリを表します。

`hamvmqm` コマンドは、このキュー・マネージャー属性を変更しません。この状況では、IBM MQ Explorer または `MQSC` コマンド `ALTER QMGR` を使用して、新しい TLS キー・リポジトリ・ファイルを指すようにキュー・マネージャー属性 `SSLKEYR` を変更する必要があります。

手順は次のとおりです。

1. キュー・マネージャーをシャットダウンし、エラーがないことを確認します。

2. キュー・マネージャーのログ・ファイルまたはキュー・ファイルがすでに共有ディスクに保管されている場合は、この手順の以下の部分を省略し、[426 ページの『MSCS 制御下にキュー・マネージャーを置く』](#)へ直接進みます。
3. キュー・ファイルおよびログ・ファイルの完全メディア・バックアップを作成し、バックアップを安全な位置に保管します (この重要性については [435 ページの『キュー・マネージャーのログ・ファイル』](#)を参照)。
4. 適切な共有ディスク・リソースを既に持っている場合は、ステップ 6 に進みます。そうでない場合は、MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーのログ・ファイルおよびデータ(キュー)ファイルを保管するための十分な容量を持つタイプ共有ディスクのリソースを作成します。
5. 共有ディスクをテストするため、MSCS クラスター・アドミニストレーターを使用して共有ディスクをクラスター・ノードから別のノードへ移動し、また元に戻します。
6. 共有ディスクが、キュー・マネージャーのログ・ファイルとデータ・ファイルをローカルで保管しているクラスター・ノード上でオンラインになっていることを確認します。
7. ユーティリティー・プログラムを実行して、次のようにキュー・マネージャーを移動します。

```
hamvmqm /m qmname /dd " e: \
IBM MQ " /ld " e: \
IBM MQ \log"
```

*qmname* は該当するキュー・マネージャー名に、*e* は共有ディスク・ドライブ文字に、*IBM MQ* は選択したディレクトリーに、それぞれ置き換えて入力します。ディレクトリーは、なければ作成されます。

8. IBM MQ Explorer を使用し、キュー・マネージャーをテストして、機能することを確認します。以下に例を示します。
  - a. キュー・マネージャーのツリー・ノードを右マウス・ボタン・クリックして、「**Start (開始)**」を選択します。キュー・マネージャーが開始します。
  - b. 「キュー」 ツリー・ノードを右クリックし、「**新規**」 > 「**ローカル・キュー ...**」を選択します。キューに名前を付けます。
  - c. 「**完了**」をクリックします。
  - d. キューを右マウス・ボタン・クリックして、「**Put Test Message... (テスト・メッセージの書き込み...)**」を選択します。「Put Test Message (テスト・メッセージの書き込み)」パネルが表示されます。
  - e. メッセージ・テキストを入力してから、「**Put Test Message (テスト・メッセージの書き込み)**」をクリックし、パネルをクローズします。
  - f. キューを右クリックして、「**メッセージの参照...**」を選択します。「メッセージ・ブラウザー」パネルが表示されます。
  - g. メッセージがキュー上にあることを確認して、「**Close (クローズ)**」をクリックします。「メッセージ・ブラウザー」パネルが閉じます。
  - h. キューを右クリックして、「**Clear Messages... (メッセージの消去...)**」を選択します。キューにあるメッセージがクリアされます。
  - i. キューを右クリックして、「**削除...**」を選択します。確認パネルが表示されるため、「**OK (了解)**」をクリックします。キューが削除されます。
  - j. キュー・マネージャーのツリー・ノードを右マウス・ボタン・クリックして、「**Stop... (停止...)**」を選択します。「キュー・マネージャーの終了」パネルが表示されます。
  - k. 「**OK**」をクリックします。キュー・マネージャーが停止します。
9. IBM MQ のように、キュー・マネージャーの始動属性が手動に設定されていることを確認します。IBM MQ Explorer で、キュー・マネージャー・プロパティー・パネルの「Startup (始動)」フィールドを `manual` に設定します。
10. [426 ページの『MSCS 制御下にキュー・マネージャーを置く』](#)に進みます。

## Windows MSCS 制御下にキュー・マネージャーを置く

MSCS 制御下にキュー・マネージャーを置くことに関連したタスク。前提条件のタスクを含みます。

### MSCS 制御下にキュー・マネージャーを置く前に

MSCS 制御下にキュー・マネージャーを置く前に、以下の作業を行ってください。

1. IBM MQ およびその MSCS サポートが、クラスター内の両方のマシンにインストールされていること、[421 ページの『MSCS クラスター化用の IBM MQ のセットアップ』](#)に説明されているように各コンピューター上のソフトウェアがまったく同じであることを確認してください。
2. **haregtyp** ユーティリティー・プログラムを使用し、IBM MQ を MSCS リソース・タイプとして、すべてのクラスター・ノードに登録してください。追加情報については、[436 ページの『MSCS ユーティリティー・プログラムのサポート』](#)を参照してください。
3. キュー・マネージャーをまだ作成していない場合は、[423 ページの『MSCS で使用するようキュー・マネージャーを作成する』](#)を参照してください。
4. キュー・マネージャーをすでに作成してある場合、またはキュー・マネージャーがすでに存在している場合は、[424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#)にある手順を実行したかどうかを確認してください。
5. キュー・マネージャーを実行中の場合は、コマンド・プロンプトまたは IBM MQ エクスプローラーを使用して停止します。
6. このトピックの以下の Windows の手順のどちらかに進む前に、共有ドライブの MSCS に対する試験操作を行います。

### Windows Server 2012

Windows Server 2012 でキュー・マネージャーを MSCS 制御下に置くには、以下の手順に従います。

1. キュー・マネージャーのホストであるクラスター・ノード・コンピューターにログインします。または、クラスター管理権限を持つユーザーとしてリモート・ワークステーションにログインし、キュー・マネージャーのホストであるクラスター・ノードに接続します。
2. フェイルオーバー・クラスター管理ツールを開始します。
3. 「フェイルオーバー・クラスター管理」>「クラスターの接続...」を右クリックします。クラスターへの接続を開きます。
4. Windows の以前のバージョン上の MSCS Cluster Administrator で使用されたグループ方式とは異なり、フェイルオーバー・クラスター管理ツールはサービスおよびアプリケーションの概念を使用します。構成されたサービスまたはアプリケーションには、1つのアプリケーションをクラスター化するために必要なすべてのリソースが含まれています。以下のようにして、MSCS の下でキュー・マネージャーを構成できます。
  - a. クラスターを右クリックし、「役割の構成」を選択して構成ウィザードを開始します。
  - b. 「サービスまたはアプリケーションの選択」パネルで、「他のサーバー」を選択します。
  - c. クライアント・アクセス・ポイントとして、適切な IP アドレスを選択します。

このアドレスは未使用の IP アドレスである必要があり、仮想 キュー・マネージャーへ接続するためにクライアントと他のキュー・マネージャーによって使用されます。この IP アドレスはどちらかのノードの通常の (静的な) アドレスではなく、両方の間を浮動する追加のアドレスです。MSCS ではこのアドレスのルーティングは処理されますが、アドレスが到達可能であることは確認されません。

- d. キュー・マネージャー専用のストレージ・デバイスを割り当てます。このデバイスは、リソース・インスタンスとして作成してからでなければ割り当てることができません。

1つのドライブにログとキュー・ファイルの両方を保管することもできますし、複数のドライブに分割することもできます。いずれの場合も、各キュー・マネージャーが独自の共用ディスクを持っている場合は、キュー・マネージャーが使用するすべてのドライブが、キュー・マネージャー(つまり、ドライブに依存していないもの)に対して、排他的なものであることを確認してください。また、キュー・マネージャーによって使用されるすべてのドライブ用のリソース・インスタンスを作成する必要があります。

ドライブのリソース・タイプは、使用する SCSI サポートに応じて異なります。SCSI アダプターの説明を参照してください。各共有ドライブにすでにグループとリソースが存在する可能性もあります。その場合は、各ドライブのリソース・インスタンスを作成する必要はありません。そのインスタンスを現在のグループからキュー・マネージャー用に作成したグループへ移動します。

各ドライブ・リソースごとに、両方のノードの所有者を設定します。従属リソースを「none (なし)」に設定します。

- e. 「リソース・タイプの選択」 パネルで、「**MQSeries MSCS**」リソースを選択します。
  - f. ウィザードでの残りのステップを完了します。
5. リソースをオンラインにする前に、MQSeries® MSCS リソースは以下の追加の構成を必要とします。
- a. 「新しい MQSeries MSCS」という名前のリソースを含む、新しく定義されたサービスを選択します。
  - b. MQ リソースで、「**プロパティ**」を右クリックします。
  - c. 以下のように、リソースを構成します。
    - Name 対象のキュー・マネージャーを識別しやすくする名前を選択します。
    - Run in a separate Resource Monitor より良い隔離のため
    - Possible owners 両方のノードの設定
    - Dependencies このキュー・マネージャーのドライブと IP アドレスを追加します。

**警告:** これらの依存関係を追加できないことは、IBM MQ がフェイルオーバー時にキュー・マネージャーの状況を誤ったクラスターのディスクに書き込もうとしていることを意味しています。多くのプロセスが同時にこのディスクに書き込もうとしているため、IBM MQ プロセスの実行がブロックされる場合があります。

    - Parameters 以下のとおりです。
      - QueueManagerName (必須)。このリソースを制御するキュー・マネージャーの名前です。このキュー・マネージャーはローカル・コンピューター上になければなりません。
      - PostOnlineCommand (オプション)。キュー・マネージャー・リソースの状態がオフラインからオンラインに変わった時に必ず実行されるプログラムを指定できます。詳細については、435 ページの『[MSCS の PostOnlineCommand と PreOfflineCommand](#)』を参照してください。
      - PreOfflineCommand (オプション)。キュー・マネージャー・リソースの状態がオンラインからオフラインに変わった時に必ず実行されるプログラムを指定できます。詳細については、435 ページの『[MSCS の PostOnlineCommand と PreOfflineCommand](#)』を参照してください。

**注:** looksAlive ポーリング間隔は、デフォルト値の 5000 ミリ秒に設定されます。isAlive ポーリング間隔は、デフォルト値の 60000 ミリ秒に設定されます。これらのデフォルトは、リソース定義が完了した後にのみ変更できます。詳細については、432 ページの『[MSCS での looksAlive と isAlive のポーリング](#)』を参照してください。
  - d. 任意で、優先ノードを設定します(ただし、[436 ページの『MSCS の優先ノードの使用』](#)の内容に注意してください)。
  - e. フェイルオーバー・ポリシー は、デフォルトで適切な値に設定されますが、リソース・フェイルオーバー とグループ・フェイルオーバー を制御するしきい値および期間を調整して、その値をキュー・マネージャーにかかる負荷に合わせることもできます。
6. MSCS クラスター・アドミニストレーターでキュー・マネージャーをオンラインにし、テスト・ワークロードをかけることによってキュー・マネージャーをテストします。テスト用のキュー・マネージャーで実験する場合は、IBM MQ エクスプローラーを使用します。以下に例を示します。
- a. 「キュー」 ツリー・ノードを右クリックし、「**新規**」 > 「**ローカル・キュー ...**」を選択します。キューに名前を付けます。
  - b. 「完了」をクリックします。キューが作成され、コンテンツ・ビューに表示されます。
  - c. キューを右マウス・ボタン・クリックして、「**Put Test Message... (テスト・メッセージの書き込み...)**」を選択します。「Put Test Message (テスト・メッセージの書き込み)」パネルが表示されます。

- d. メッセージ・テキストを入力してから、「**Put Test Message (テスト・メッセージの書き込み)**」をクリックし、パネルをクローズします。
  - e. キューを右クリックして、「**メッセージの参照...**」を選択します。「メッセージ・ブラウザー」パネルが表示されます。
  - f. メッセージがキュー上にあることを確認して、「**Close (クローズ)**」をクリックします。「メッセージ・ブラウザー」パネルが閉じます。
  - g. キューを右クリックして、「**Clear Messages... (メッセージの消去...)**」を選択します。キューにあるメッセージがクリアされます。
  - h. キューを右クリックして、「**削除...**」を選択します。確認パネルが表示されるため、「**OK (了解)**」をクリックします。キューが削除されます。
7. MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーをオフラインにできるかテストし、さらにオンラインに戻せるかテストします。
  8. フェイルオーバーをシミュレートします。

MSCS クラスター・アドミニストレーターで、キュー・マネージャーを含むグループを右クリックして、**Move Group** を選択します。これが完了するには数分かかることがあります。(キュー・マネージャーをもう一方のノードへ移動する操作を別の機会に行うことによって時間を節約する場合は、424 ページの『[キュー・マネージャーを MSCS ストレージに移動する](#)』の手順に従ってください。) 右クリックして **Initiate Failure** を選択することもできます。アクション (ローカル再始動またはフェイルオーバー) は、現在の状態と構成設定によって異なります。

## Windows Server 2008

Windows Server 2008 でキュー・マネージャーを MSCS 制御下に置くには、以下の手順に従います。

1. キュー・マネージャーのホストであるクラスター・ノード・コンピューターにログインします。または、クラスター管理権限を持つユーザーとしてリモート・ワークステーションにログインし、キュー・マネージャーのホストであるクラスター・ノードに接続します。
2. フェイルオーバー・クラスター管理ツールを開始します。
3. 「**フェイルオーバー・クラスター管理**」 > 「**クラスターの管理 ...**」 を右クリックします。クラスターへの接続を開きます。
4. Windows の以前のバージョン上の MSCS Cluster Administrator で使用されたグループ方式とは異なり、フェイルオーバー・クラスター管理ツールはサービスおよびアプリケーションの概念を使用します。構成されたサービスまたはアプリケーションには、1つのアプリケーションをクラスター化するために必要なすべてのリソースが含まれています。以下のようにして、MSCS の下でキュー・マネージャーを構成できます。
  - a. 「**サービスとアプリケーション**」 > 「**サービスまたはアプリケーションの構成 ...**」 を右クリックします。構成ウィザードを開始します。
  - b. 「**サービスまたはアプリケーションの選択**」 パネルで「**その他のサーバー**」を選択します。
  - c. クライアント・アクセス・ポイントとして、適切な IP アドレスを選択します。

このアドレスは未使用の IP アドレスである必要があり、仮想 キュー・マネージャーへ接続するためにクライアントと他のキュー・マネージャーによって使用されます。この IP アドレスはどちらかのノードの通常の (静的な) アドレスではなく、両方の間を浮動する追加のアドレスです。MSCS ではこのアドレスのルーティングは処理されますが、アドレスが到達可能であることは確認されません。
  - d. キュー・マネージャー専用のストレージ・デバイスを割り当てます。このデバイスは、リソース・インスタンスとして作成してからでなければ割り当てることができません。

1つのドライブにログとキュー・ファイルの両方を保管することもできますし、複数のドライブに分割することもできます。いずれの場合も、各キュー・マネージャーが独自の共用ディスクを持っている場合は、キュー・マネージャーが使用するすべてのドライブが、キュー・マネージャー (つまり、ドライブに依存していないもの) に対して、排他的なものであることを確認してください。また、キュー・マネージャーによって使用されるすべてのドライブ用のリソース・インスタンスを作成する必要があります。

ドライブのリソース・タイプは、使用する SCSI サポートに応じて異なります。SCSI アダプターの説明を参照してください。各共有ドライブにすでにグループとリソースが存在する可能性もあります。その場合は、各ドライブのリソース・インスタンスを作成する必要はありません。そのインスタンスを現在のグループからキュー・マネージャー用に作成したグループへ移動します。

各ドライブ・リソースごとに、両方のノードの所有者を設定します。従属リソースを「none (なし)」に設定します。

- e. 「リソース・タイプの選択」 パネルで、「MQSeries MSCS」リソースを選択します。
  - f. ウィザードでの残りのステップを完了します。
5. リソースをオンラインにする前に、MQSeries MSCS リソースは以下の追加の構成を必要とします。
- a. 「新しい MQSeries MSCS」という名前のリソースを含む、新しく定義されたサービスを選択します。
  - b. MQ リソースで、「プロパティ」を右クリックします。
  - c. 以下のように、リソースを構成します。
    - Name 対象のキュー・マネージャーを識別しやすくする名前を選択します。
    - Run in a separate Resource Monitor より良い隔離のため
    - Possible owners 両方のノードの設定
    - Dependencies このキュー・マネージャーのドライブと IP アドレスを追加します。

**警告:** これらの依存関係を追加できないことは、IBM MQ がフェイルオーバー時にキュー・マネージャーの状況を誤ったクラスターのディスクに書き込もうとしていることを意味しています。多くのプロセスが同時にこのディスクに書き込もうとしているため、IBM MQ プロセスの実行がブロックされる場合があります。

    - Parameters 以下のとおりです。
      - QueueManagerName (必須)。このリソースを制御するキュー・マネージャーの名前です。このキュー・マネージャーはローカル・コンピューター上になければなりません。
      - PostOnlineCommand (オプション)。キュー・マネージャー・リソースの状態がオフラインからオンラインに変わった時に必ず実行されるプログラムを指定できます。詳細については、[435 ページの『MSCS の PostOnlineCommand と PreOfflineCommand』](#)を参照してください。
      - PreOfflineCommand (オプション)。キュー・マネージャー・リソースの状態がオンラインからオフラインに変わった時に必ず実行されるプログラムを指定できます。詳細については、[435 ページの『MSCS の PostOnlineCommand と PreOfflineCommand』](#)を参照してください。

**注:** looksAlive ポーリング間隔は、デフォルト値の 5000 ミリ秒に設定されます。isAlive ポーリング間隔は、デフォルト値の 60000 ミリ秒に設定されます。これらのデフォルトは、リソース定義が完了した後のみ変更できます。詳細については、[432 ページの『MSCS での looksAlive と isAlive のポーリング』](#)を参照してください。
  - d. 任意で、優先ノードを設定します(ただし、[436 ページの『MSCS の優先ノードの使用』](#)の内容に注意してください)。
  - e. フェイルオーバー・ポリシー は、デフォルトで適切な値に設定されますが、リソース・フェイルオーバー とグループ・フェイルオーバー を制御するしきい値および期間を調整して、その値をキュー・マネージャーにかかる負荷に合わせることもできます。
6. MSCS クラスター・アドミニストレーターでキュー・マネージャーをオンラインにし、テスト・ワークロードをかけることによってキュー・マネージャーをテストします。テスト用のキュー・マネージャーで実験する場合は、IBM MQ エクスプローラーを使用します。以下に例を示します。
- a. 「キュー」 ツリー・ノードを右クリックし、「新規」 > 「ローカル・キュー ...」を選択します。キューに名前を付けます。
  - b. 「完了」をクリックします。キューが作成され、コンテンツ・ビューに表示されます。
  - c. キューを右マウス・ボタン・クリックして、「Put Test Message... (テスト・メッセージの書き込み...)」を選択します。「Put Test Message (テスト・メッセージの書き込み)」パネルが表示されます。

- d. メッセージ・テキストを入力してから、「**Put Test Message (テスト・メッセージの書き込み)**」をクリックし、パネルをクローズします。
  - e. キューを右クリックして、「**メッセージの参照...**」を選択します。「**メッセージ・ブラウザー**」パネルが表示されます。
  - f. メッセージがキュー上にあることを確認して、「**Close (クローズ)**」をクリックします。「**メッセージ・ブラウザー**」パネルが閉じます。
  - g. キューを右クリックして、「**Clear Messages... (メッセージの消去...)**」を選択します。キューにあるメッセージがクリアされます。
  - h. キューを右クリックして、「**削除...**」を選択します。確認パネルが表示されるため、「**OK (了解)**」をクリックします。キューが削除されます。
7. MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーをオフラインにできるかテストし、さらにオンラインに戻せるかテストします。
  8. フェイルオーバーをシミュレートします。

MSCS クラスター・アドミニストレーターで、キュー・マネージャーを含むグループを右クリックして、**Move Group** を選択します。これが完了するには数分かかることがあります。(キュー・マネージャーをもう一方のノードへ移動する操作を別の機会に行うことによって時間を節約する場合は、424 ページの『[キュー・マネージャーを MSCS ストレージに移動する](#)』の手順に従ってください。) 右クリックして **Initiate Failure** を選択することもできます。アクション (ローカル再始動またはフェイルオーバー) は、現在の状態と構成設定によって異なります。

## Windows 2003

Windows 2003 でキュー・マネージャーを MSCS 制御下に置くには、以下の手順に従います。

1. キュー・マネージャーのホストであるクラスター・ノード・コンピューターにログインします。または、クラスター管理権限を持つユーザーとしてリモート・ワークステーションにログインし、キュー・マネージャーのホストであるクラスター・ノードに接続します。
2. MSCS クラスター・アドミニストレーターを始動します。
3. クラスターへの接続をオープンします。
4. キュー・マネージャーのリソースをまとめるための MSCS グループを作成します。このグループには、どのキュー・マネージャーが関係しているのか簡単にわかるような名前を付けます。各グループには複数のキュー・マネージャーを含めることができます (422 ページの『[複数のキュー・マネージャーを MSCS と共に使用する](#)』を参照)。

以下の作業はすべてこのグループを使用して行います。

5. キュー・マネージャーが使用する各 SCSI 論理ドライブのリソース・インスタンスを作成します。
  - 1 つのドライブにログとキュー・ファイルの両方を保管することもできますし、複数のドライブに分割することもできます。いずれの場合も、各キュー・マネージャーが独自の共用ディスクを持っている場合は、キュー・マネージャーが使用するすべてのドライブが、キュー・マネージャー (つまり、ドライブに依存していないもの) に対して、排他的なものであることを確認してください。また、キュー・マネージャーによって使用されるすべてのドライブ用のリソース・インスタンスを作成する必要があります。

ドライブのリソース・タイプは、使用する SCSI サポートに応じて異なります。SCSI アダプターの説明を参照してください。各共有ドライブにすでにグループとリソースが存在する可能性もあります。その場合は、各ドライブのリソース・インスタンスを作成する必要はありません。そのインスタンスを現在のグループからキュー・マネージャー用に作成したグループへ移動します。

各ドライブ・リソースごとに、両方のノードの所有者を設定します。従属リソースを「none (なし)」に設定します。

6. IP アドレスのリソース・インスタンスを作成します。
  - IP アドレス・リソース (リソース・タイプ *IP アドレス*) を作成します。このアドレスは未使用の IP アドレスである必要があり、仮想 キュー・マネージャーへ接続するためにクライアントと他のキュー・マネージャーによって使用されます。この IP アドレスはどちらかのノードの通常の (静的な) アドレ

スではなく、両方の間を浮動する追加のアドレスです。MSCSではこのアドレスのルーティングは処理されますが、アドレスが到達可能であることは確認されません。

#### 7. キュー・マネージャーにリソース・インスタンスを作成します。

タイプ *IBM MQ MSCS* のリソースを作成します。ウィザードでは、以下の項目を含んで、さまざまな項目の入力をします:

- Name 対象のキュー・マネージャーを識別しやすくする名前を選択します。
- Add to group 作成したグループを使用する
- Run in a separate Resource Monitor より良い隔離のため
- Possible owners 両方のノードの設定
- Dependencies このキュー・マネージャーのドライブと IP アドレスを追加します。

**警告:** これらの依存関係を追加できないことは、IBM MQ がフェイルオーバー時にキュー・マネージャーの状況を誤ったクラスターのディスクに書き込もうとしていることを意味しています。多くのプロセスが同時にこのディスクに書き込もうとしているため、IBM MQ プロセスの実行がブロックされる場合があります。

#### • Parameters 以下のとおりです。

- QueueManagerName (必須)。このリソースを制御するキュー・マネージャーの名前です。このキュー・マネージャーはローカル・コンピューター上になければなりません。
- PostOnlineCommand (オプション)。キュー・マネージャー・リソースの状態がオフラインからオンラインに変わった時に必ず実行されるプログラムを指定できます。詳細については、[435 ページの『MSCS の PostOnlineCommand と PreOfflineCommand』](#)を参照してください。
- PreOfflineCommand (オプション)。キュー・マネージャー・リソースの状態がオンラインからオフラインに変わった時に必ず実行されるプログラムを指定できます。詳細については、[435 ページの『MSCS の PostOnlineCommand と PreOfflineCommand』](#)を参照してください。

**注:** *looksAlive* ポーリング間隔は、デフォルト値の 5000 ミリ秒に設定されます。*isAlive* ポーリング間隔は、デフォルト値の 30000 ミリ秒に設定されます。これらのデフォルトは、リソース定義が完了した後にのみ変更できます。詳細については、[432 ページの『MSCS での looksAlive と isAlive のポーリング』](#)を参照してください。

8. 任意で、優先ノードを設定します (ただし、[436 ページの『MSCS の優先ノードの使用』](#)の内容に注意してください)。
9. フェイルオーバー・ポリシー (グループのプロパティの定義を参照) は、デフォルトでは適切な値に設定されますが、リソース・フェイルオーバーとグループ・フェイルオーバーを制御するしきい値と期間を調整して、その値をキュー・マネージャーにかかる負荷に合わせることもできます。
10. MSCS クラスター・アドミニストレーターでキュー・マネージャーをオンラインにし、テスト・ワークロードをかけることによってキュー・マネージャーをテストします。テスト用のキュー・マネージャーで実験する場合は、IBM MQ エクスプローラーを使用します。以下に例を示します。
  - a. 「キュー」 ツリー・ノードを右クリックし、「新規」 > 「ローカル・キュー ...」を選択します。キューに名前を付けます。
  - b. 「完了」をクリックします。キューが作成され、コンテンツ・ビューに表示されます。
  - c. キューを右マウス・ボタン・クリックして、「Put Test Message... (テスト・メッセージの書き込み...)」を選択します。「Put Test Message (テスト・メッセージの書き込み)」パネルが表示されます。
  - d. メッセージ・テキストを入力してから、「Put Test Message (テスト・メッセージの書き込み)」をクリックし、パネルをクローズします。
  - e. キューを右クリックして、「メッセージの参照...」を選択します。「メッセージ・ブラウザー」パネルが表示されます。
  - f. メッセージがキュー上にあることを確認して、「Close (クローズ)」をクリックします。「メッセージ・ブラウザー」パネルが閉じます。

- g. キューを右クリックして、「**Clear Messages... (メッセージの消去...)**」を選択します。キューにあるメッセージがクリアされます。
  - h. キューを右クリックして、「**削除...**」を選択します。確認パネルが表示されるため、「**OK (了解)**」をクリックします。キューが削除されます。
11. MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーをオフラインにできるかテストし、さらにオンラインに戻せるかテストします。
  12. フェイルオーバーをシミュレートします。

MSCS クラスター・アドミニストレーターで、キュー・マネージャーを含むグループを右クリックして、**Move Group** を選択します。これが完了するには数分かかることがあります。(キュー・マネージャーをもう一方のノードへ移動する操作を別の機会に行うことによって時間を節約する場合は、[424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#)の手順に従ってください。) 右クリックして **Initiate Failure** を選択することもできます。アクション (ローカル再始動またはフェイルオーバー) は、現在の状態と構成設定によって異なります。

### Windows MSCS での *looksAlive* と *isAlive* のポーリング

*looksAlive* および *isAlive* は、MSCS がリソース・タイプで提供されているライブラリー・コードにコールバックを行い、そのリソースがそれ自体の作業状況を判別するための検査を実行するように要求する間隔です。これは最終的には、MSCS がリソースのフェイルオーバーを行うかどうかを判別します。

*looksAlive* 間隔が経過すると (デフォルトは 5000 ms)、キュー・マネージャー・リソースが呼び出され、その状況が適切なものであるかどうかを判別する独自の検査を実行します。

*isAlive* 間隔が経過すると (デフォルトは 30000 ms)、キュー・マネージャー・リソースへの別の呼び出しが行われ、リソースが正しく機能しているかどうかを判別する別の検査が実行されます。これにより、2つのレベルのリソース・タイプ検査を行うことができます。

1. *looksAlive* 状況検査では、リソースが機能しているかどうか明らかにされます。
2. *isAlive* はより重要な検査であり、キュー・マネージャー・リソースがアクティブであるかどうか判別されます。

キュー・マネージャー・リソースがアクティブではないと判別された場合、MSCS は他の拡張 MSCS オプションに基づいて、そのリソースおよび関連する従属リソースの、クラスター内の別のノードへのフェイルオーバーをトリガーします。詳細については、[MSCS の資料](#)を参照してください。

### Windows MSCS 制御からキュー・マネージャーを除去する

MSCS 制御からキュー・マネージャーを除去し、それらのキュー・マネージャーを手動管理に戻すことができます。

保守操作のために、MSCS 制御からキュー・マネージャーを除去する必要はありません。この除去は、MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーを一時的にオフラインにすることによって行えます。MSCS 制御下からのキュー・マネージャーの除去は、より永続的な状態への変更であり、MSCS によるキュー・マネージャーの制御がこれ以上必要なくなったと判断できた時にだけ行います。

除去するキュー・マネージャーが TSL 接続を使用する場合は、IBM MQ エクスプローラーまたは MQSC コマンド **ALTER QMGR** を使用して、キュー・マネージャー属性 **SSLKEYR** がローカル・ディレクトリー上の TLS キー・リポジトリー・ファイルを指すように変更する必要があります。

手順は次のとおりです。

1. [433 ページの『MSCS からキュー・マネージャーをオフラインにする』](#)で説明されているように、MSCS クラスター・アドミニストレーターを使用してキュー・マネージャー・リソースをオフラインにします。
2. リソース・インスタンスを破棄します。この操作では、キュー・マネージャーは破棄されません。
3. 必要に応じて、共用ドライブからローカル・ドライブへキュー・マネージャー・ファイルを移行します。その方法については、[433 ページの『キュー・マネージャーを MSCS ストレージから戻す』](#)を参照してください。
4. キュー・マネージャーをテストします。

## MSCS からキュー・マネージャーをオフラインにする

MSCS からキュー・マネージャーをオフラインにするには、以下のステップを実行します。

1. MSCS クラスター・アドミニストレーターを始動します。
2. クラスターへの接続をオープンします。
3. Groups または Role (Windows 2012 を使用している場合) を選択し、移動するキュー・マネージャーが含まれているグループを開きます。
4. キュー・マネージャーのリソースを選択します。
5. それを右クリックして、Offline を選択します。
6. アクションが完了するまで待ちます。

## キュー・マネージャーを MSCS ストレージから戻す

この手順に従うと、キュー・マネージャーがコンピューターのローカル・ドライブに戻るように構成されます。つまり、通常の IBM MQ キュー・マネージャーになります。この操作を行うには、ログ・ファイルとデータ・ファイルを共有ディスクから移動します。例えば、既存のキュー・マネージャーには、E:\WebSphere MQ\log\QMname や E:\WebSphere MQ\qmgrs\QMname などのパスが存在する場合があります。ファイルは手動で移動せずに、以下のように IBM MQ MSCS サポートの一部として提供される **hamvmqm** ユーティリティ・プログラムを使用してください。

1. キュー・ファイルおよびログ・ファイルの完全メディア・バックアップを作成し、バックアップを安全な位置に保管します (この重要性については [435 ページの『キュー・マネージャーのログ・ファイル』](#) を参照)。
2. 使用するローカル・ドライブを決めて、キュー・マネージャーのログ・ファイルと データ (キュー) ファイルを保管するのに十分な容量があるか確認します。
3. ファイルが現在収められている共有ディスクが、キュー・マネージャーのログ・ファイルと データ・ファイルを移動する先のクラスター・ノード上でオンラインになっていることを確認します。
4. ユーティリティ・プログラムを実行して、次のようにキュー・マネージャーを移動します。

```
hamvmqm /m qmname /dd " c:\
IBM MQ " /ld "c:\
IBM MQ \log"
```

*qmname* は該当するキュー・マネージャー名に、*c* はローカル・ディスク・ドライブ文字に、*IBM MQ* は選択したディレクトリー (選択したディレクトリーが存在していない場合は作成される) に、それぞれ置き換えて入力します。

5. キュー・マネージャーをテストして、機能することを確認します ([424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#) を参照してください)。

### Windows MSCS の使用に関するヒント

このセクションには、MSCS の IBM MQ サポートを効果的に使用するために役立つ一般情報が記載されています。

このセクションには、MSCS の IBM MQ サポートを効果的に使用するために役立つ一般情報が記載されています。

あるマシンから別のマシンにキュー・マネージャーをフェイルオーバーするには、どれくらいの時間がかかるでしょうか。これは、キュー・マネージャーのワークロード量、およびトラフィックの構成 (例えば、トラフィックがどれくらい持続的か、同期点内のトラフィック、障害の前にトラフィックがどれくらいコミットされていたか) によってかなり異なります。IBM の実験では、フェイルオーバーおよびフェイルバック時間はおよそ 1 分でした。この実験は、非常に負荷の少ないキュー・マネージャーで行われましたが、実際の時間は負荷によってかなり違いがあります。

### Windows MSCS の動作確認

以下のステップに従って、稼働中の MSCS クラスターがあることを確認してください。

423 ページの『[MSCS で使用するよう](#)にキュー・マネージャーを作成する』の手順で始まるタスクの説明では、リソースの作成、移動、および破棄が可能な MSCS クラスターが稼働中であることが前提になります。そのようなクラスターがあることを確認する方法は次のとおりです。

1. MSCS クラスター・アドミニストレーターを使用して、グループを作成します。
2. そのグループ内で汎用アプリケーション・リソースのインスタンスを作成し、システム・クロック (パス名 C:&#xa5;winnt&#xa5;system32&#xa5;clock.exe で、C:&#xa5; の作業ディレクトリー) を指定します。
3. リソースをオンラインにできること、そのリソースを含むグループを他のノードに移動できること、およびリソースをオフラインにできることを確認します。

#### Windows 手動始動と MSCS

MSCS が管理するキュー・マネージャーでは、必ず始動属性を手動に設定しなければなりません。これによって、IBM MQ MSCS サポートは、即時にキュー・マネージャーを開始しなくても MQSeries サービスを再開できます。

IBM MQ MSCS サポートは、モニターおよび制御を行えるように、サービスを再始動できなければなりません。どのキュー・マネージャーがどのマシンで実行されるかを自身で管理し続ける必要があります。詳しくは、[424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#)を参照してください。

#### Windows MSCS およびキュー・マネージャー

MSCS を使用しているときのキュー・マネージャーに関する考慮事項。

### 他のノードに対応するキュー・マネージャーを作成する

クラスタリングが IBM MQ で機能するためには、ノード A 上のキュー・マネージャーとノード B 上のキュー・マネージャーを同一にする必要があります。ただし、二番目のものを明示的に作成する必要はありません。1つのノードでキュー・マネージャーを作成または準備し、それを [424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#)で説明されている方法で他のノードに移動すれば、そのキュー・マネージャーは完全にそのノード上で複製されます。

### デフォルト・キュー・マネージャー

MSCS 制御下では、デフォルト・キュー・マネージャーは使用しないでください。キュー・マネージャーには、そのキュー・マネージャーをデフォルトにするためのプロパティがありません。IBM MQ には独自のレコードが個別に存在します。デフォルトに設定されたキュー・マネージャーをフェイルオーバー時に他のコンピューターへ移動しても、移動先ではデフォルトになりません。すべてのアプリケーションが特定のキュー・マネージャーをその名前で参照するように設定してください。

### キュー・マネージャーの削除

キュー・マネージャーをノード間で移動すると、その詳細が両方のコンピューターのレジストリーに記録されます。これを削除する場合は、一方のコンピューターで削除し、[436 ページの『MSCS ユーティリティー・プログラムのサポート』](#)で説明しているようにユーティリティーを実行し、もう一方のコンピューターのレジストリーをクリーンアップします。

### 既存キュー・マネージャーのサポート

キュー・マネージャーのログ・ファイルとキュー・ファイルを、2つのマシン間の共有 SCSI バス上のディスクに書き込むことが可能な場合は、既存のキュー・マネージャーを MSCS 制御下に置くことができます ([421 ページの図 73](#)を参照してください)。MSCS リソースの作成中は、キュー・マネージャーを一時的にオフラインにする必要があります。

キュー・マネージャーを新規作成する場合は、MSCS から独立して作成し、テストしてから、MSCS 制御下に置きます。以下を参照してください。

- [423 ページの『MSCS で使用するよう](#)にキュー・マネージャーを作成する』
- [424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#)

## 管理するキュー・マネージャーを MSCS に指示する

MSCS 制御の下にどのキュー・マネージャーを置くかは、MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーごとにリソース・インスタンスを作成することによって選択できます。このプロセスで、そのインスタンスを管理するキュー・マネージャーを選択できる、リソースのリストが表示されます。

## キュー・マネージャーのログ・ファイル

キュー・マネージャーを MSCS ストレージへ移動すると、そのログ・ファイルとデータ・ファイルは共有ディスクへ移動されます (この例については、[424 ページの『キュー・マネージャーを MSCS ストレージに移動する』](#)を参照してください)。

移動する前に、キュー・マネージャーを完全にシャットダウンし、データ・ファイルとログ・ファイルの全バックアップを取ることをお勧めします。

## 複数のキュー・マネージャー

IBM MQ MSCS サポートを使用すると、各マシンで複数のキュー・マネージャーを実行することができ、個々のキュー・マネージャーを MSCS 制御下に置くことができます。

### **Windows** 常に MSCS を使用してクラスターを管理する

制御コマンドか、または IBM MQ Explorer を使用して、MSCS の制御下にあるキュー・マネージャーに対して直接始動および停止操作を実行しないでください。その代わりに、MSCS クラスター・アドミニストレーターを使用して、キュー・マネージャーをオンラインまたはオフラインにします。

MSCS クラスター・アドミニストレーターの使用は一部には、実際にはキュー・マネージャーを MSCS の制御外で始動したにもかかわらず、キュー・マネージャーがオフラインであることを MSCS が報告するために引き起こされる混乱を避けるためです。さらに重要なことに、MSCS を使用せずにキュー・マネージャーを停止すると、MSCS はこれを障害として検出し、他のノードへのフェイルオーバーが開始されます。

### **Windows** MSCS のアクティブ/アクティブ・モードでの作業

アクティブ/アクティブ・モードでは、MSCS クラスター内にある 2 つのコンピューターの両方がキュー・マネージャーを実行していることがあります。完全に使用されていないマシンを待機用として用意する必要はありません (その必要があれば、これはアクティブ/パッシブ・モードで可能です)。

両方のマシンでワークロードを実行する場合、十分なパフォーマンス・レベルでクラスター・ワークロード全体を実行するには、それぞれのマシンに十分な容量 (プロセッサ、メモリー、2 次ストレージ) が必要です。

**注:** MSCS を Microsoft Transaction Server (COM+) で使用する場合、Active/Active モードは使用できません。これは、IBM MQ を MSCS および COM+ と共に使用して、次のことを行うためです。

- IBM MQ の COM+ サポートを使用するアプリケーション・コンポーネントは、COM+ の一部である分散トランザクション調整プログラム (Distributed Transaction Coordinator (DTC)) と同じコンピューター上で実行する必要があります。
- キュー・マネージャーも、同じコンピューター上で実行する必要があります。
- DTC は MSCS リソースとして構成する必要があります。そのように構成しておくと、いつでもクラスター内のコンピューターの 1 つでのみ実行できます。

### **Windows** MSCS の *PostOnlineCommand* と *PreOfflineCommand*

これらのコマンドを使用して、IBM MQ MSCS サポートをその他のシステムと統合することができます。これらを使用して IBM MQ コマンドを発行できますが、いくつかの制限があります。

これらのコマンドは、タイプ IBM MQ MSCS のリソースに対するパラメーターで指定します。これらのコマンドを使用して、IBM MQ MSCS サポートをその他のシステムまたはプロシージャーと統合することができます。例えば、メール・メッセージの送信、ページャーのアクティブ化、または別のモニター・システ

ムによってキャプチャーされる別の形式のアラートの生成を行うプログラムの名前を指定することが可能です。

PostOnlineCommand は、リソースがオフラインからオンラインに変更されると起動され、PreOfflineCommand は、リソースがオンラインからオフラインに変更されると起動されます。起動時、これらのコマンドはデフォルトで Windows システム・ディレクトリーから実行されます。IBM MQ は 32 ビット・リソース・モニター・プロセスを使用するため、Windows64 ビット・システムでは、\Windows\system32 ディレクトリーではなく、\Windows\SysWOW64 ディレクトリーになります。詳しくは、Windows x64 環境でのファイル・リダイレクトに関する Microsoft の資料を参照してください。どちらのコマンドも、MSCS クラスター・サービスを実行するのに使用されるユーザー・アカウントの下で実行され、また非同期で起動されます。IBM MQ MSCS サポートは、それらのコマンドの完了を待たずに処理を続行します。このため、この後のクラスター操作を妨害したり、遅らせたりする危険はありません。

また、これらのコマンドを使用して IBM MQ コマンドを実行し、例えばリクエスター・チャンネルを再始動できます。ただし、コマンドはキュー・マネージャーの状態変更の時点で実行されたため、長時間実行される機能を実行するには意図されていません。また、キュー・マネージャーの現在の状態を憶測することはできません。キュー・マネージャーがオンラインになった直後に、アドミニストレーターがオフライン・コマンドを発行する可能性があります。

キュー・マネージャーの状態に依存するプログラムを実行する場合は、MSCS Generic Application リソース・タイプのインスタンスを作成し、それらをキュー・マネージャー・リソースと同じ MSCS グループに配置し、それらをキュー・マネージャー・リソースに依存させることを検討してください。

#### Windows MSCS の優先ノードの使用

MSCS でアクティブ/アクティブ・モードを使用する場合は、各キュー・マネージャーの優先ノードを構成すると便利です。しかし、一般的には優先ノードを設定しないで、手動フェイルバックのみを使用することをお勧めします。

その他の比較的ステータスなリソースとは異なり、キュー・マネージャーでは、あるノードから別のノードへのフェイルオーバー (またはフェイルバック) に時間がかかることがあります。不必要な停止を避けるため、キュー・マネージャーをリカバリー済みノードにフェイルバックする前にそのノードをテストしてください。これを行うと、即時フェイルバック設定は使用できません。1日のうちのある時間にフェイルバックが行われるように構成できます。

ノードが完全にリカバリーされたことを確認できたときに、キュー・マネージャーを手動で必要なノードに移動する方法が最も安全です。この方法では、優先ノード・オプションは使用できません。

#### Windows MSCS へのインストール時の COM+ エラー

新しくインストールされた MSCS クラスターに IBM MQ をインストールする場合、アプリケーション・イベント・ログに、ソース COM+ のエラーおよびイベント ID 4691 が報告されることがあります。

これは、Microsoft Distributed Transaction Coordinator (MSDTC) が Microsoft Cluster Server (MSCS) 環境で実行するように構成されていない状態で、IBM MQ をこの環境で実行しようとしたことを意味します。クラスター環境で MSDTC を構成する方法については、Microsoft の資料を参照してください。

#### Windows MSCS ユーティリティー・プログラムのサポート

コマンド・プロンプトで実行できる MSCS の IBM MQ サポートのユーティリティー・プログラムのリスト。

MSCS の IBM MQ サポートには、以下のユーティリティー・プログラムが含まれています。

##### リソース・タイプの登録/登録解除 haregtyp.exe

IBM MQ MSCS リソース・タイプを登録解除すると、それ以降、そのタイプのリソースは作成できなくなります。MSCS では、クラスター内にまだ存在しているインスタンスのタイプとして指定されているリソース・タイプを登録解除することはできません。

1. MSCS 制御下で実行されているキュー・マネージャーを停止するため、MSCS クラスター・アドミニストレーターを使用して、[433 ページの『MSCS からキュー・マネージャーをオフラインにする』](#)の手順に従ってキュー・マネージャーをオフラインにします。
2. MSCS クラスター・アドミニストレーターを使用して、リソース・インスタンスを削除します。

3. リソース・タイプを登録解除するため、コマンド・プロンプトで次のコマンドを入力します。

```
haregtyp /u
```

タイプを後から登録(または後から再登録)する場合は、コマンド・プロンプトで次のコマンドを入力します。

```
haregtyp /r
```

MSCS ライブラリーを正常に登録した後、IBM MQ のインストール後まだシステムをリブートしていない場合は、リブートする必要があります。

#### キュー・マネージャーを MSCS ストレージに移動する

hamvmqm.exe

424 ページの『[キュー・マネージャーを MSCS ストレージに移動する](#)』を参照してください。

#### キュー・マネージャーをノードから削除する

hadltmqm.exe

クラスターにキュー・マネージャーがあり、そのキュー・マネージャーがあるノードから別のノードに移動し、今はそれを破棄したいという場合を考えてください。現在キュー・マネージャーが存在するノードで、そのキュー・マネージャーを削除するには、IBM MQ エクスプローラーを使用します。削除後も、キュー・マネージャーのレジストリー項目は別のコンピューター上に引き続き存在します。これらを削除するには、そのコンピューター上でプロンプトに次のコマンドを入力してください。

```
hadltmqm /m qmname
```

qmname は、除去されるキュー・マネージャーの名前です。

#### セットアップの詳細を確認および保管する

amqmsysn.exe

このユーティリティは、ご使用の IBM MQ MSCS サポートのセットアップについての詳細(これは IBM サポートに連絡する場合に必要な可能性があります)が示されるダイアログを表示します。詳細をファイルに保管するためのオプションがあります。

## Multi 複数インスタンス・キュー・マネージャー

複数インスタンス・キュー・マネージャーは、異なる複数のサーバー上で構成されている同じキュー・マネージャーのインスタンスです。キュー・マネージャーの1つのインスタンスはアクティブ・インスタンスとして定義され、もう1つのインスタンスはスタンバイ・インスタンスとして定義されます。アクティブ・インスタンスで障害が発生すると、複数インスタンス・キュー・マネージャーは、スタンバイ・サーバーで自動的に再始動します。

### 複数インスタンス・キュー・マネージャーの構成の例

438 ページの図 74 は、キュー・マネージャー QM1 の複数インスタンスの構成の例を示しています。2つのサーバーに IBM MQ がインストールされ、そのうちの1つは予備です。1つのキュー・マネージャー QM1 は作成済みです。QM1 の1つのインスタンスはアクティブで、1サーバー上で実行しています。QM1 の他方のインスタンスは、他方のサーバー上でスタンバイとして実行しており、アクティブな処理は実行していませんが、QM1 のアクティブ・インスタンスに障害が起きた場合には、そのアクティブ・インスタンスを引き継ぐ準備ができています。

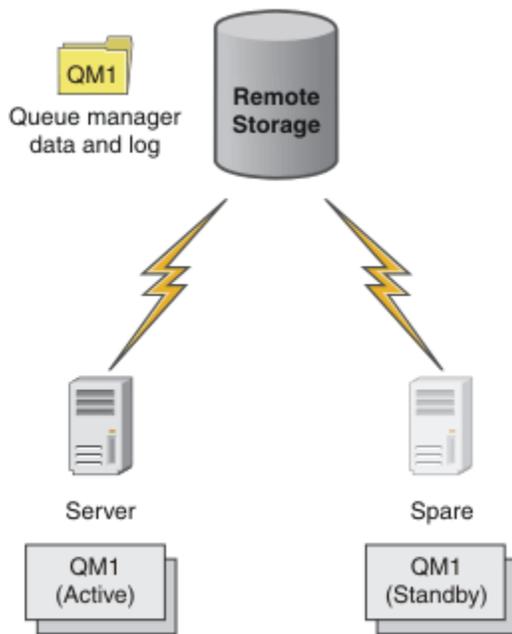


図 74. 複数インスタンス・キュー・マネージャー

キュー・マネージャーを複数インスタンス・キュー・マネージャーとして使用しようとしている場合は、**crtmqm** コマンドを使用して一方のサーバー上で単一のキュー・マネージャーを作成し、そのキュー・マネージャーのデータおよびログを共有ネットワーク・ストレージに配置します。もう一方のサーバー上では、キュー・マネージャーを再び作成するのではなく、**addmqinf** コマンドを使用して、ネットワーク・ストレージ上のキュー・マネージャーのデータおよびログへの参照を作成します。

これで、キュー・マネージャーはどちらのサーバーからでも実行できるようになりました。それぞれのサーバーは、同じキュー・マネージャーのデータおよびログを参照します。キュー・マネージャーは、1つしかなく、一度に1サーバー上でのみアクティブになります。

キュー・マネージャーは、単一インスタンス・キュー・マネージャーまたは複数インスタンス・キュー・マネージャーのいずれかとして実行できます。どちらの場合も、1インスタンスのキュー・マネージャーだけが実行し、要求を処理します。相違点となっているのは、複数インスタンス・キュー・マネージャーとして実行している場合には、キュー・マネージャーのアクティブ・インスタンスを実行していないサーバーはスタンバイ・インスタンスとして実行し、アクティブ・サーバーに障害が起きた場合に、アクティブ・インスタンスから自動的に引き継ぐ準備ができています。

どちらのインスタンスが最初にアクティブになるかを制御するのは、2つのサーバーでキュー・マネージャーを開始する順序だけです。キュー・マネージャーのデータに対する読み取り/書き込みロックを最初に獲得したインスタンスが、アクティブ・インスタンスになります。

もう一方のサーバーが開始しているのであれば、切り替えオプションを使用してアクティブ・インスタンスを停止し、制御をスタンバイに転送することで、アクティブ・インスタンスをそのもう一方のサーバーと交換できます。

QM1 のアクティブ・インスタンスは、実行時に共有キュー・マネージャーのデータおよびログ・フォルダーに対して排他的アクセスを行います。アクティブ・インスタンスに障害が起ると、QM1 のスタンバイ・インスタンスがこれを検出し、アクティブ・インスタンスになります。アクティブ・インスタンスが残したままの状態の QM1 のデータおよびログを引き継いで、クライアントおよびチャネルからの再接続を受け入れます。

アクティブ・インスタンスでは、以下のさまざまな理由から障害が起こる可能性があり、その結果としてスタンバイが引き継ぐことになります。

- アクティブ・キュー・マネージャー・インスタンスをホストするサーバーの障害。
- アクティブ・キュー・マネージャー・インスタンスをホストするサーバーとファイル・システム間の接続障害。

- 無応答のキュー・マネージャー・プロセス。IBM MQ はこれを検出すると、キュー・マネージャーをシャットダウンします。

キュー・マネージャーの構成情報を複数のサーバーに追加し、アクティブ/スタンバイのペアとして実行する任意の2つのサーバーを選択することができます。インスタンスの数は合計2つまでという制限があります。2つのスタンバイ・インスタンスと1つのアクティブ・インスタンスを設定することはできません。

## 高可用性ソリューションを構築するために必要な追加のコンポーネント

複数インスタンス・キュー・マネージャーは、高可用性ソリューションの一部です。有用な高可用性ソリューションを構築するには、いくつかの追加のコンポーネントが必要です。

- アクティブ・キュー・マネージャー・インスタンスの実行を引き継ぐコンピューターに IBM MQ 接続を転送するための、クライアントおよびチャネルの再接続。
- ロックを正しく管理し、メディアおよびファイル・サーバーを障害から保護するハイパフォーマンス共有ネットワーク・ファイル・システム (NFS)。

**重要:** NFS ドライブの保守を実行するには、その前に環境内で実行されているすべての複数インスタンス・キュー・マネージャー・インスタンスを停止する必要があります。NFS に障害が発生した場合に備えて、リカバリー用のキュー・マネージャー構成のバックアップを取っておいてください。

- 基本インフラストラクチャー内の Single Point of Failure を排除する、弾力性のあるネットワークおよび電源機構。
- フェイルオーバーを許容するアプリケーション。特に、トランザクション・アプリケーションの動作、および IBM MQ キューを参照するアプリケーションには十分注意する必要があります。
- アクティブ・インスタンスとスタンバイ・インスタンスの実行を確認し、障害が起きたアクティブ・インスタンスを再始動するための、それらのインスタンスのモニターおよび管理。複数インスタンス・キュー・マネージャーが自動的に再始動するとしても、スタンバイ・インスタンスが実行していて引き継ぎを行える状態であることと、障害が起きたインスタンスが新しいスタンバイ・インスタンスとしてオンラインに復帰できることを確認する必要があります。

IBM MQ MQI clients およびチャネルは、アクティブになったスタンバイ・キュー・マネージャーに自動的に再接続します。再接続および高可用性ソリューションにおけるその他のコンポーネントについての詳細は、関連トピックにあります。IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

## サポートされているオペレーティング・システム

IBM WebSphere MQ 7.0.1 以降でサポートされている任意の非 z/OS プラットフォームで、複数インスタンス・キュー・マネージャーを作成できます。

IBM WebSphere MQ 7.0.1 以降では、MQI クライアントの自動クライアント再接続がサポートされています。

## 複数インスタンス・キュー・マネージャーの作成

1つのサーバーでキュー・マネージャーを作成し、別のサーバーで IBM MQ を構成することで、複数インスタンス・キュー・マネージャーを作成します。複数インスタンス・キュー・マネージャーは、キュー・マネージャーのデータとログを共有します。

複数インスタンス・キュー・マネージャーを作成する作業の大半は、キュー・マネージャーの共有のデータ・ファイルとログ・ファイルをセットアップする作業です。ネットワーク・ストレージに共有ディレクトリーを作成し、ネットワーク共有の機能を使用してそれらのディレクトリーを他のサーバーからも利用できるようにする必要があります。これらの作業は、UNIX and Linux システムの *root* などの、管理権限を持つユーザーが実行する必要があります。手順は、次のとおりです。

1. データ・ファイルとログ・ファイルの共有フォルダーを作成します。
2. 1つのサーバーでキュー・マネージャーを作成します。
3. 最初のサーバーで `dspmqinf` コマンドを実行し、キュー・マネージャーの構成データを収集して、クリップボードにコピーします。

4. コピーしたデータを対象にして **addmqinf** コマンドを実行し、第 2 のサーバーでキュー・マネージャーの構成を作成します。

第 2 のサーバーで **crtmqm** を実行して、キュー・マネージャーを再び作成することはしません。

## ファイル・アクセス制御

他のすべてのサーバー上にあるユーザーおよびグループ **mqm** が、シェアへのアクセス許可を持っていることを確認する必要があります。

UNIX and Linux では、すべてのシステム上で **mqm** の **uid** と **gid** を同じものにする必要があります。このために、各システムの **/etc/passwd** を編集し、**mqm** に共通の **uid** および **gid** を設定して、ご使用のシステムをリブートすることが必要になる場合があります。

Microsoft Windows の場合、キュー・マネージャーのプロセスを実行するユーザー ID には、キュー・マネージャーのデータ・ファイルとログ・ファイルが含まれているディレクトリーに対するフル・コントロール権限が必要です。この権限を構成するには、2つの方法があります。

1. 代替セキュリティ・プリンシパルとしてグローバル・グループを使用してキュー・マネージャーを作成します。そのグローバル・グループに、キュー・マネージャーのデータ・ファイルとログ・ファイルが含まれているディレクトリーに対するフル・コントロール権限を与えます。[469 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 \(Windows\)』](#)を参照してください。キュー・マネージャーを実行するユーザー ID をそのグローバル・グループのメンバーにします。ローカル・ユーザーをグローバル・グループのメンバーにすることはできないので、キュー・マネージャーのプロセスは、ドメイン・ユーザー ID で実行しなければなりません。ドメイン・ユーザー ID は、ローカル・グループ **mqm** のメンバーでなければなりません。このような方法で保護したファイルを使用する複数インスタンス・キュー・マネージャーをセットアップする方法については、[443 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 \(Windows\)』](#)の作業を参照してください。
2. ドメイン・コントローラー上にキュー・マネージャーを作成して、ローカル **mqm** グループにドメイン・スコープ "ドメイン・ローカル" を設定します。ドメイン・ローカル **mqm** によってファイル共有を保護し、キュー・マネージャーのすべてのインスタンスでその同じドメイン・ローカル **mqm** グループを使用してキュー・マネージャーのプロセスを実行するようにします。このような方法で保護したファイルを使用する複数インスタンス・キュー・マネージャーをセットアップする方法については、[459 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』](#)の作業を参照してください。

## 構成情報

それぞれのキュー・マネージャー・インスタンスを実行する各サーバーに関する IBM MQ キュー・マネージャー構成情報を変更することによって、必要な数だけキュー・マネージャー・インスタンスを構成できます。各サーバーでは、同じバージョンの互換性のあるフィックス・レベルの IBM MQ をインストールしておく必要があります。追加のキュー・マネージャー・インスタンスを構成するときには、**dspmqinf** コマンドと **addmqinf** コマンドが役立ちます。あるいは、**mq5.ini** ファイルと **qm.ini** ファイルを直接編集することもできます。[483 ページの『Linux での複数インスタンス・キュー・マネージャーの作成』](#)、[443 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 \(Windows\)』](#)、および [459 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』](#) のトピックは、複数インスタンス・キュー・マネージャーの構成方法を示す例です。

Windows、UNIX and Linux システムでは、単一の **mq5.ini** ファイルをネットワーク・シェアに置き、**AMQ\_MQS\_INI\_LOCATION** 環境変数がそれを指すように設定することで、そのファイルを共有できます。

## 制限対象機能

1. 同じキュー・マネージャーの複数インスタンスは、同じオペレーティング・システム、アーキテクチャー、およびエンディアンを持つサーバー上でのみ構成してください。例えば、両方のマシンを 32 ビットか 64 ビットで統一する必要があります。

- すべての IBM MQ インストールは、リリース・レベル 7.0.1 以上でなければなりません。
- 一般に、アクティブ・インストールとスタンバイ・インストールは、同じ保守レベルで保守されます。すべてのインストールをまとめてアップグレードする必要があるかどうかを確認するために、それぞれのアップグレードの保守の手順を参照してください。

アクティブおよびパッシブのキュー・マネージャーの保守レベルは同一でなければならないことに注意してください。

- キュー・マネージャーのデータとログを共有するのは、同じ IBM MQ ユーザー、同じグループ、同じアクセス制御メカニズムで構成したキュー・マネージャー同士の間限定してください。 **IBM i** 例  
例えば、Linux サーバー上にセットアップされたネットワーク・シェアには、UNIX and Linux キュー・マネージャーのそれぞれのデータおよびログを入れることができますが、IBM i が使用するキュー・マネージャーのデータは入れることができません。

**IBM i** IBM i システム用と UNIX システム用の複数の共有場所を同じネットワーク・ストレージ上に作成できます。ただし、共有場所を別々にすることが必要です。異なる所有者にそれぞれ異なるシェアを付与することができます。この制限は、UNIX と IBM i の間で IBM MQ ユーザーおよびグループに異なる名前が使用された結果として生じます。そのユーザーとグループの uid と gid が同じ場合もあり得ますが、この制約事項に変わりはありません。

- UNIX and Linux システム上では、ネットワーク・ストレージ上のファイル共有システムを、ソフト・マウントではなく、割り込み可能なハード・マウントで構成します。割り込み可能なハード・マウントは、システム呼び出しにより中断されるまで、強制的にキュー・マネージャーを停止させます。ソフト・マウントでは、サーバーの障害後のデータ整合性は保証されません。
- この共有ログおよびデータ・ディレクトリーは、FAT または NFSv3 ファイル・システムには格納できません。Windows の複数インスタンスのキュー・マネージャーの場合、Windows ネットワークによって使用される Common Internet File System (CIFS) プロトコルでネットワーク・ストレージにアクセスする必要があります。
- z/OS** z/OS は、複数インスタンス・キュー・マネージャーをサポートしません。キュー共有グループを使用してください。

再接続可能クライアントは、z/OS キュー・マネージャーと連携可能です。

#### **Windows** Windows ドメインおよび複数インスタンス・キュー・マネージャー

Windows で複数インスタンス・キュー・マネージャーを実行する場合は、データとログを共有する必要があります。その共有フォルダーは、それぞれのサーバーやワークステーションで稼働するキュー・マネージャーのすべてのインスタンスからアクセスできる状態でなければなりません。キュー・マネージャーと共有フォルダーを 1 つの Windows ドメインの中で構成してください。キュー・マネージャーは、ドメイン・ワークステーションでも、ドメイン・サーバーでも、ドメイン・コントローラーでも実行できます。

複数インスタンス・キュー・マネージャーを構成する前に、473 ページの『[Windows でキュー・マネージャーのデータとログの非共有ディレクトリーおよびファイルを保護する操作](#)』と 469 ページの『[共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 \(Windows\)](#)』を読んで、キュー・マネージャーのデータ・ファイルとログ・ファイルへのアクセスを制御する方法を確認してください。これらのトピックは学習用です。Windows ドメインで複数インスタンス・キュー・マネージャーの共有ディレクトリーをセットアップする作業をすぐに始める場合は、443 ページの『[ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 \(Windows\)](#)』を参照してください。

### ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの実行

IBM WebSphere MQ 7.1 から、ドメインのメンバーであるワークステーションやサーバーで複数インスタンス・キュー・マネージャーを実行できるようになりました。IBM WebSphere MQ 7.1 より前のリリースでは、ドメイン・コントローラーでのみ複数インスタンス・キュー・マネージャーを実行できました。442 ページの『[ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの実行](#)』を参照してください。Windows で複数インスタンス・キュー・マネージャーを実行するには、ドメイン・コントロー

ラーとファイル・サーバーのほかに、同じドメインに接続する同じキュー・マネージャーを実行するワークステーションまたはサーバーが2つ必要です。

ドメインに含まれているサーバーまたはワークステーションで複数インスタンス・キュー・マネージャーを実行できるように設計を変更したのは、追加のセキュリティ・グループを使用してキュー・マネージャーを作成できるようにするためです。追加のセキュリティ・グループは、`crtmqm` コマンドの `-a` パラメーターで渡します。その場合は、キュー・マネージャーのデータとログが含まれているディレクトリーをそのグループで保護します。キュー・マネージャーのプロセスを実行するユーザー ID は、そのグループのメンバーでなければなりません。キュー・マネージャーがそれらのディレクトリーにアクセスすると、それらのディレクトリーにアクセスする権限がユーザー ID にあるかどうかを Windows が検査します。そのグループとユーザー ID の両方にドメイン・スコープを与えれば、キュー・マネージャーのプロセスを実行するユーザー ID は、そのグローバル・グループから資格情報を引き継ぐこととなります。キュー・マネージャーを別のサーバーで実行する場合でも、キュー・マネージャーのプロセスを実行するユーザー ID は、同じ資格情報を使用できます。ユーザー ID 自体は、同じでなくてもかまいません。いずれにしても、代替セキュリティ・グループとローカル mqm グループの両方のメンバーになっていることが必要です。

複数インスタンス・キュー・マネージャーを作成する作業は、IBM WebSphere MQ 7.0.1 の場合と基本的に同じですが、1つだけ変更があります。つまり、追加のセキュリティ・グループ名を `crtmqm` コマンドのパラメーターに追加しなければならない、ということです。この作業については、[443 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 \(Windows\)』](#) を参照してください。

ドメイン、ドメイン・サーバー、ドメイン・ワークステーションの構成には、複数の手順が必要になります。Windows でデータ・ディレクトリーとログ・ディレクトリーにアクセスする権限をキュー・マネージャーに与える方法も理解しておかなければなりません。ログ・ファイルとデータ・ファイルにアクセスする権限をキュー・マネージャーのプロセスに与える方法が分からない場合は、[473 ページの『Windows でキュー・マネージャーのデータとログの非共有ディレクトリーおよびファイルを保護する操作』](#) のトピックを参照してください。このトピックには、必要な手順を理解するために役立つ2つの作業が含まれています。その作業とは、[475 ページの『ローカル mqm グループによって許可されたデータ・ファイルとログ・ファイルの読み取りと書き込み』](#) と [479 ページの『代替ローカル・セキュリティ・グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りと書き込み』](#) です。さらに、[469 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 \(Windows\)』](#) のトピックでは、キュー・マネージャーのデータ・ファイルとログ・ファイルが含まれている共有ディレクトリーを代替セキュリティ・グループで保護する方法が取り上げられています。このトピックには、4つの作業が含まれています。つまり、Windows ドメインのセットアップ、ファイル共有の作成、IBM MQ for Windows のインストール、キュー・マネージャーでファイル共有を使用するための構成です。それらの作業を以下に示します。

1. [446 ページの『Active Directory と DNS のドメインの作成 \(Windows\)』](#) .
2. [450 ページの『Windows ドメイン内のサーバーまたはワークステーションに IBM MQ をインストールする操作』](#) .
3. [453 ページの『キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 \(Windows\)』](#) .
4. [456 ページの『代替グローバル・セキュリティ・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み』](#) .

次に、ドメインを使用して、[443 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 \(Windows\)』](#) の作業を実行します。実動ドメインで知識を活用する前に、これらの作業を実行して、複数インスタンス・キュー・マネージャーのセットアップについて検討してください。

## ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの実行

IBM WebSphere MQ 7.0.1 では、ドメイン・コントローラーでのみ複数インスタンス・キュー・マネージャーを実行できました。キュー・マネージャーのデータは、ドメイン mqm グループで保護できました。[469 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 \(Windows\)』](#) のトピックで説明されているように、ワークステーションやサーバーでは、ローカル mqm グループで保護されているディレクトリーを共有できません。一方、ドメイン・コントローラーでは、すべてのグループとプリンシパルのスコープがドメイン・スコープになります。ドメイン・コントローラーに

IBM MQ for Windows をインストールすると、キュー・マネージャーのデータ・ファイルとログ・ファイルはドメイン `mqm` グループで保護されます。このグループは共有できます。ドメイン・コントローラーで複数インスタンス・キュー・マネージャーを構成する場合は、[459 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』](#)の作業の手順を実行してください。

## 関連情報

### [Managing Authorization and Access Control](#)

### [Windows サーバー・クラスター・ノードをドメイン・コントローラーとして使用する方法](#)

**Windows** ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (*Windows*)

Windows ドメインの一部であるワークステーションまたはサーバーの Windows 上で複数インスタンス・キュー・マネージャーをセットアップする方法の例を示します。サーバーは、ドメイン・コントローラーである必要はありません。ここで取り上げるのは、実動スケールのセットアップではなく、関連する概念を説明する程度のセットアップです。この例は、Windows Server 2008 をベースとしています。それぞれのステップは、Windows Server の他のバージョンでは異なる場合があります。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

#### **sun**

Windows Server 2008 ドメイン・コントローラー。Sun、mars、および venus が含まれている `wmq.example.com` ドメインを所有します。例示目的で、これをファイル・サーバーとしても使用します。

#### **mars**

最初の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの QMGR という 1 つのインスタンスを組み込みます。

#### **venus**

2 番目の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの QMGR という 2 番目のインスタンスを組み込みます。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

## 始める前に

Windows では、キュー・マネージャー・データとログ・ファイルを保管するファイル・システムを検証する必要はありません。検査の手順、[共有ファイル・システムの動作の検証](#)は、UNIX and Linux に適用されます。Windows では検査は必ず成功します。

以下の作業手順を実行します。これらのタスクは、ドメイン・コントローラーおよびドメインを作成し、サーバーに IBM MQ for Windows をインストールし、データおよびログ・ファイル用のファイル共有を作成します。既存のドメイン・コントローラーを構成する場合でも、新しい Windows Server 2008 でこれらの手順を試してみる価値はあります。それぞれのドメインに合わせてこれらの手順を調整できます。

1. [446 ページの『Active Directory と DNS のドメインの作成 \(Windows\)』](#).
2. [450 ページの『Windows ドメイン内のサーバーまたはワークステーションに IBM MQ をインストールする操作』](#).
3. [453 ページの『キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 \(Windows\)』](#).
4. [456 ページの『代替グローバル・セキュリティ・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み』](#).

## このタスクについて

このタスクは、キュー・マネージャーのインスタンスを実行するように、ドメイン・コントローラーと 2 つのサーバーをドメイン内で構成するための、一連のタスクの 1 つです。このタスクでは、キュー・マネ

ージャークMGRのもう1つのインスタンスを実行するように、2番目のサーバー *venus* を構成します。キュー・マネージャークMGRの2番目のインスタンスを作成し、機能するかどうかをテストするには、このタスクの手順に従います。

このタスクは、前のセクションの4つのタスクとは別のものです。単一インスタンス・キュー・マネージャークを複数インスタンス・キュー・マネージャークに変換する手順が含まれています。他のすべての手順は、単一インスタンス・キュー・マネージャークでも複数インスタンス・キュー・マネージャークでも共通です。

## 手順

1. IBM MQ for Windows を実行するように2番目のサーバーを構成します。

- a) タスク 450 ページの『[Windows ドメイン内のサーバーまたはワークステーションに IBM MQ をインストールする操作](#)』の手順を実行して、2番目のドメイン・サーバーを作成します。この一連のタスクでは、2番目のサーバーを *venus* と呼びます。

**ヒント:** 2つのサーバーそれぞれにおいて、IBM MQ に同じインストール・デフォルトを使用して、2番目のインストール済み環境を作成します。デフォルトが異なる場合は、IBM MQ 構成ファイル `mqs.ini` に含まれている **QMGR QueueManager** スタンザ内の `Prefix` 変数と `InstallationName` 変数を調整しなければならないことがあります。変数は、インストール済み環境ごと、および各サーバー上のキュー・マネージャークごとに異なる可能性のあるパスを参照します。パスがすべてのサーバーで同一であると、より簡単に複数インスタンス・キュー・マネージャークを構成できます。

2. *venus* で QMGR の2番目のインスタンスを作成します。

- a) *mars* 上の QMGR が存在しない場合は、タスク 456 ページの『[代替グローバル・セキュリティ・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み](#)』を実行して作成します。
- b) `Prefix` パラメーターと `InstallationName` パラメーターの値が *venus* の正しい値になっていることを確認します。

*mars* で、**dspmqlnf** コマンドを実行します。

```
dspmqlnf QMGR
```

システム応答:

```
QueueManager:  
Name=QMGR  
Directory=QMGR  
Prefix=C: ¥ProgramData¥IBM¥MQ  
DataPath=\\sun\wmq\data\QMGR  
InstallationName=Installation1
```

- c) 機械可読形式の **QueueManager** スタンザをクリップボードにコピーします。

*mars* で、`-o command` パラメーターを指定して **dspmqlnf** コマンドを再度実行します。

```
dspmqlnf -o command QMGR
```

システム応答:

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C: ¥ProgramData¥IBM¥MQ"  
-v DataPath=\\sun\wmq\data\QMGR
```

- d) *venus* で **addmqinf** コマンドを実行し、クリップボードの情報を使用して、キュー・マネージャークのインスタンスを *venus* で作成します。

必要に応じて、Prefix パラメーターまたは InstallationName パラメーターの違いに合わせてコマンドを調整します。

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"
-v DataPath=\\sun\wmq\data\QMGR
```

IBM MQ configuration information added.

3. *venus* 上でスタンバイ・インスタンスを許可して、キュー・マネージャー *QMGR* を開始します。

a) *mars* 上の *QMGR* が停止していることを確認します。

*mars* で、**dspmq** コマンドを実行します。

```
dspmq -m QMGR
```

システム応答は、キュー・マネージャーが停止した方法に応じて異なります。以下に例を示します。

```
C:\Users\Administrator>dspmq -m QMGR
QMNAME(QMGR) STATUS(Ended immediately)
```

b) *venus* 上で、**strmqm** コマンドを実行し、スタンバイを許可して *QMGR* を開始します。

```
strmqm -x QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using 7.1.0.0.
```

## タスクの結果

複数インスタンス・キュー・マネージャーが切り替わるかどうかをテストするには、以下の手順に従います。

1. *mars* 上で、**strmqm** コマンドを実行し、スタンバイを許可して *QMGR* を開始します。

```
strmqm -x QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started.
The active instance is running elsewhere.
```

2. *venus* で、**endmqm** コマンドを実行します。

```
endmqm -r -s -i QMGR
```

*venus* 上のシステム応答は以下のとおりです。

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended, permitting switchover to  
a standby instance.
```

*mars* 上のシステム応答は以下のとおりです。

```
dspmq  
QMNAME(QMGR) STATUS(Running as standby)  
C:\Users\wmquser2>dspmq  
QMNAME(QMGR) STATUS(Running as standby)  
C:\Users\wmquser2>dspmq  
QMNAME(QMGR) STATUS(Running)
```

## 次のタスク

サンプル・プログラムを使用して複数インスタンス・キュー・マネージャーを検証するには、[467 ページ](#)の『Windows での複数インスタンス・キュー・マネージャーの検査』を参照してください。

### **Windows** Active Directory と DNS のドメインの作成 (Windows)

この作業では、*sun* という Windows 2008 のドメイン・コントローラーで *wmq.example.com* ドメインを作成します。さらに、そのドメインで Domain *mqm* グローバル・グループを構成し、適切な権限と 1 人のユーザーを設定します。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

#### **sun**

Windows Server 2008 ドメイン・コントローラー。Sun、*mars*、および *venus* が含まれている *wmq.example.com* ドメインを所有します。例示目的で、これをファイル・サーバーとしても使用します。

#### **mars**

最初の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの *QMGR* という 1 つのインスタンスを組み込みます。

#### **venus**

2 番目の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの *QMGR* という 2 番目のインスタンスを組み込みます。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

## 始める前に

1. この作業手順は、Windows Server 2008 がインストールされているものの、役割が何も設定されていない環境を対象にしています。既存のドメイン・コントローラーを構成する場合でも、新しい Windows Server 2008 でこれらの手順を試してみる価値はあります。それぞれのドメインに合わせてこれらの手順を調整できます。

## このタスクについて

この作業では、新しいドメイン・コントローラーで Active Directory と DNS のドメインを作成します。その後、そのドメインを構成し、そのドメインに加わる他のサーバーやワークステーションに IBM MQ をインストールする準備を整えます。Active Directory のインストールと構成を実行して Windows ドメインを作成する作業に慣れていない場合でも、以下の手順を実行してみてください。複数インスタンス・キュー・マネージャーの構成を作成するには、Windows ドメインを作成する必要があります。ここで取り上げる作業手順が Windows ドメインを構成するための最適な方法であるというわけではありません。実稼働環境で複数インスタンス・キュー・マネージャーをデプロイする場合は、Windows の資料を参照してください。

この作業では、以下の流れで手順を実行します。

1. Active Directory をインストールします。
2. ドメインを追加します。
3. ドメインを DNS に追加します。
4. グローバル・グループ Domain mqm を作成して、適切な権限を与えます。
5. 1 人のユーザーを追加して、そのユーザーをグローバル・グループ Domain mqm のメンバーにします。

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。さらに、441 ページの『[Windows ドメインおよび複数インスタンス・キュー・マネージャー](#)』の作業も必要です。

この作業では、ドメイン・コントローラーのホスト名が *sun*、2 つの IBM MQ サーバーの名前がそれぞれ *mars* と *venus* になります。ドメインの名前は、*wmq.example.com* です。この作業で斜体で表記している名前はすべて、それぞれの好みの名前に置き換えてもかまいません。

## 手順

1. ローカル管理者または Workgroup 管理者として、ドメイン・コントローラー *sun* にログオンします。  
サーバーが既にドメイン・コントローラーとして構成されている場合は、ドメイン管理者としてログオンする必要があります。
2. 「Active Directory ドメイン サービス」ウィザードを実行します。
  - a) クリックスタート > 実行... dcpromo と入力して、「OK」をクリックします。  
Active Directory のバイナリー・ファイルがインストールされていなければ、Windows によってこれらのファイルが自動的にインストールされます。
3. ウィザードの最初のウィンドウで、「詳細モード インストールを使用する」チェック・ボックスは、クリアされた状態のままにしておきます。「次へ」 > 「次へ」をクリックし、「新しいフォレストに新しいドメインを作成する」 > 「次へ」をクリックします。
4. 「フォレストルートドメインの FQDN」フィールドに *wmq.example.com* と入力します。「次へ」をクリックします。
5. 「フォレスト機能レベルの設定」ウィンドウで、「フォレスト機能レベル」 > 「次へ」のリストから「Windows Server 2003」以降を選択します。  
IBM MQ でサポートされている最も古いレベルの Windows サーバーは、Windows Server 2003 です。
6. オプション: 「ドメイン機能レベルの設定」ウィンドウで、「ドメイン機能レベル」 > 「次へ」のリストから「Windows Server 2003 以降」を選択します。  
このステップは、フォレストの機能レベルを「Windows Server 2003」に設定した場合にのみ必要です。
7. 「追加のドメイン コントローラ オプション」ウィンドウが開きます。追加のオプションとして「DNS サーバー」が選択された状態になります。「次へ」をクリックし、「はい」をクリックして警告ウィンドウを消去します。

**ヒント:** DNS サーバーが既にインストールされていると、このオプションは表示されません。この作業を厳密に実行したいと思う場合は、このドメイン・コントローラーからすべての役割を削除してから、再び作業を開始してください。

8. から移動 DatabaseLog Files と SYSVOL ディレクトリは変更せずに、次をクリックする。
9. 「ディレクトリー サービス復元モード Administrator パスワード」ウィンドウの「パスワード」フィールドと「パスワードの確認入力」フィールドにパスワードを入力します。「次へ」 > 「次へ」をクリックします。ウィザードの最後のウィンドウで「完了時に再起動する」を選択します。
10. ドメイン・コントローラーがリブートされたら、`wmq\Administrator` としてログオンします。

サーバー・マネージャーが自動的に始動します。

11. `wmq.example.com\Users` フォルダを開きます

a) [サーバーマネージャー] > [ロール] > **Active Directory Domain Services** > `wmq.example.com` > [ユーザー]を開きます。

12. 「ユーザー」 > 新規 > 「グループ」を右クリックします。

a) 「グループ名」フィールドにグループ名を入力します。

注: 優先的に使用するべきグループ名は、Domain mqm です。このとおりに入力してください。

- グループ Domain mqm を呼び出すと、ドメイン・ワークステーションまたはサーバー上の "IBM MQ" 準備ウィザードの動作が変更されます。これにより、"「IBM MQ」準備" ウィザードは、ドメイン内の IBM MQ の新規インストールのたびに、グループ Domain mqm をローカル mqm グループに自動的に追加します。

- Domain mqm というグローバル・グループがないドメインに、ワークステーションやサーバーをインストールすることもできます。これを行う場合は、Domain mqm グループと同じプロパティを持つグループを定義する必要があります。IBM MQ をドメインにインストールする場合は、必ずそのグループ、またはそのグループのメンバーであるユーザーをローカル mqm グループのメンバーにする必要があります。ドメイン・ユーザーを複数のグループに組み込むことも可能です。複数のドメイン・グループを作成し、別々に管理するインストール環境のセットごとに、それぞれのグループを対応させます。管理対象のインストール環境に基づいて、ドメイン・ユーザーをそれぞれのドメイン・グループに振り分けます。それぞれのドメイン・グループをそれぞれの対応する IBM MQ インストール環境のローカル mqm グループに追加します。特定のローカル mqm グループのメンバーになっているドメイン・グループのドメイン・ユーザーだけが、そのインストール環境のキュー・マネージャーを作成し、管理し、実行できます。

- ドメインに含まれているワークステーションまたはサーバーに IBM MQ をインストールするときに指名するドメイン・ユーザーは、Domain mqm グループのメンバーであるか、Domain mqm グループと同じプロパティで定義した代替グループのメンバーでなければなりません。

b) 「グループのスコープ」では、「グローバル」がクリックされたままの状態にしておくか、設定を「ユニバーサル」に変更します。「グループの種類」では、「セキュリティ」がクリックされたままの状態にしておきます。「OK」をクリックします。

13. 「グループのメンバーシップの読み取り」の「許可」と「groupMembershipSAM の読み取り」の「許可」の権限を Domain mqm グローバル・グループに追加します。

a) サーバー・マネージャーのアクション・バーで、「表示」 > 「高度な機能」をクリックします

b) サーバー・マネージャーのナビゲーション・ツリーで、「ユーザー」をクリックします。

c) 「Users (ユーザー)」ウィンドウで「Domain mqm (ドメイン mqm)」 > 「Properties (プロパティ)」を右クリックします。

d) 「セキュリティ」 > 「詳細設定」 > 「追加...」をクリックします。Domain mqm と入力し、「名前の確認」 > 「OK」をクリックします。

「名前」フィールドには、string Domain mqm (`domain name\Domain mqm`)が事前に入力されています。

e) 「プロパティ」をクリックします。「適用先」リストの「下位のユーザー オブジェクト」を選択します。

- f) 「アクセス許可」リストから、「グループのメンバーシップの読み取り」と「groupMembershipSAMの読み取り」の「許可」チェック・ボックスを選択し、「OK」>「適用」>「OK」>「OK」をクリックします。

#### 14. Domain mqm グローバル・グループに 2 人以上のユーザーを追加します。

1 人のユーザー (この例では *wmquuser1*) が IBM MQ サービスを実行し、もう 1 人のユーザー (*wmquuser2*) が対話式に使用されます。

ドメイン構成で代替セキュリティ・グループを使用するキュー・マネージャーを作成するには、ドメイン・ユーザーが必要です。管理者には **crtmqm** コマンドを実行する権限があるものの、管理者権限を持ったユーザー ID だけでは十分ではありません。このドメイン・ユーザーは、管理者である場合もありますが、いずれにしてもローカル mqm グループと代替セキュリティ・グループの両方のメンバーになっていなければなりません。

この例では、*wmquuser1* と *wmquuser2* を Domain mqm グローバル・グループのメンバーにします。「IBM MQ の準備」ウィザードは、ウィザードが実行されるローカル mqm グループのメンバーとして Domain mqm を自動的に構成します。

1 台のコンピューターに存在する IBM MQ のインストール環境ごとに、IBM MQ サービスを実行するユーザーを別々に設定する必要があります。別々のコンピューターで同じユーザーを再利用することは可能です。

- サーバー・マネージャーのナビゲーション・ツリーで、「ユーザー」>「新規作成」>「ユーザー」をクリックします。
- 「新規オブジェクト-ユーザー」ウィンドウで、「ユーザー・ログオン名」フィールドに *wmquuser1* と入力します。「名」フィールドに *WebSphere* と入力し、「姓」フィールドに *MQ1* と入力します。「次へ」をクリックします。
- 「パスワード」フィールドと「パスワードの確認入力」フィールドにパスワードを入力し、「ユーザーは次回ログオン時にパスワードの変更が必要」チェック・ボックスをクリアします。「次へ」>「完了」をクリックします。
- 「ユーザー」ウィンドウで、**WebSphere MQ** > 「グループに追加 ...」を右クリックします。Domain mqm と入力し、「名前の確認」>「OK」>「OK」をクリックします。
- ステップ a から d を繰り返して、*WebSphere MQ2* を *wmquuser2* として追加します。

#### 15. サービスとして IBM MQ を実行します。

IBM MQ をサービスとして実行し、サービスとして実行する権限をドメイン・ユーザー (ドメイン管理者から取得した) に付与する必要がある場合、以下の手順を実行します。

- 「スタート」>「実行...」をクリックします。  
コマンド *secpol.msc* を入力して「OK」をクリックします。
- 「セキュリティの設定」>「ローカル ポリシー」>「ユーザー権利の割り当て」を開きます。  
ポリシーのリストで、「サービスとしてログオン」>「プロパティ」を右クリックします。
- 「ユーザーまたはグループの追加...」をクリックします。  
ドメイン管理者から取得したユーザーの名前を入力し、「名前の確認」をクリックします。
- 「Windows セキュリティ」ウィンドウが表示されたら、十分な権限を持つアカウント・ユーザーまたは管理者のユーザー名とパスワードを入力し、「OK」>「適用」>「OK」をクリックします。  
「ローカルセキュリティ ポリシー」ウィンドウを閉じます。

注: Windows Server Server 2008 および Windows Server 2012 では、ユーザー・アカウント制御 (UAC) 機能はデフォルトで有効です。

UAC 機能は、ユーザーが Administrators グループのメンバーである場合でも、特定のオペレーティング・システム機能に対してユーザーが実行できる操作を制限します。この制限に対応するため、適切な手段を講じる必要があります。

## 次のタスク

次に、450 ページの『Windows ドメイン内のサーバーまたはワークステーションに IBM MQ をインストールする操作』の作業に進みます。

## 関連タスク

**Windows** [Windows ドメイン内のサーバーまたはワークステーションに IBM MQ をインストールする操作](#)

**Windows** [キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 \(Windows\)](#)

**Windows** [代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み](#)

**Windows** [Windows ドメイン内のサーバーまたはワークステーションに IBM MQ をインストールする操作](#)

このタスクでは、*wmq.example.com* Windows ドメイン内のサーバーまたはワークステーションにインストールし、IBM MQ 設定します。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

### sun

Windows Server 2008 ドメイン・コントローラー。Sun、mars、および venus が含まれている *wmq.example.com* ドメインを所有します。例示目的で、これをファイル・サーバーとしても使用します。

### mars

最初の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの QMGR という 1 つのインスタンスを組み込みます。

### venus

2 番目の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの QMGR という 2 番目のインスタンスを組み込みます。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

## 始める前に

1. [446 ページの『Active Directory と DNS のドメインの作成 \(Windows\)』](#)の手順を実行して、ドメイン *wmq.example.com* のドメイン・コントローラー sun を作成します。斜体の名前は、それぞれの構成に合わせて変更してください。
2. IBM MQ を実行できるその他の Windows バージョンについては、[Windows システムでのハードウェア要件およびソフトウェア要件](#)を参照してください。

## このタスクについて

この作業では、mars という名前の Windows Server 2008 を *wmq.example.com* ドメインのメンバーとして構成します。IBM MQ をインストールし、そのインストール環境を *wmq.example.com* ドメインのメンバーとして実行できるように構成します。

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。さらに、[441 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』](#)の作業も必要です。

この作業では、ドメイン・コントローラーのホスト名が sun、2 つの IBM MQ サーバーの名前がそれぞれ mars と venus になります。ドメインの名前は、*wmq.example.com* です。この作業で斜体で表記している名前はすべて、それぞれの好みの名前に置き換えてもかまいません。

## 手順

- ドメイン・コントローラー *sun.wmq.example.com* を DNS サーバーとして *mars* に追加します。
  - mars* で、*mars\Administrator* としてログオンし、**スタート**をクリックします。
  - 「ネットワーク」>「プロパティ」>「ネットワーク接続の管理」を右クリックします。
  - ネットワーク・アダプターを右クリックして、「プロパティ」をクリックします。

システムの応答として、「ローカルエリア接続のプロパティ」ウィンドウが開き、接続で使用する項目のリストが表示されます。
  - 「ローカル・エリア接続のプロパティ」ウィンドウの項目リストから、**Internet Protocol 「バージョン 4」** または **Internet Protocol 「バージョン 6」** を選択します。「プロパティ」>「拡張...」をクリックします。「DNS」タブをクリックします。
  - DNS サーバー・アドレスで、「追加...」をクリックします。
  - DNS サーバーでもあるドメイン・コントローラーの IP アドレスを入力して、「追加」をクリックします。
  - 「以下の DNS サフィックスを順に追加する」>「追加...」をクリックします。
  - wmq.example.com* と入力して、「追加」をクリックします。
  - 「この接続の DNS サフィックス」フィールドに *wmq.example.com* と入力します。
  - 「この接続のアドレスを DNS に登録する」と「この接続の DNS サフィックスを DNS 登録に使う」を選択します。「OK」>「OK」>「閉じる」をクリックします。
  - コマンド・ウィンドウで、**ipconfig /all** というコマンドを入力して、TCP/IP 設定を確認します。
- mars* で、このコンピューターを *wmq.example.com* ドメインに追加します。
  - 「スタート」をクリックします。
  - コンピュータ>プロパティを右クリックします。「コンピュータ名、ドメインおよびワークグループの設定」のセクションで、「設定の変更」をクリックします。
  - システム・プロパティ・ウィンドウで、「変更...」をクリックします。
  - 「ドメイン」をクリックし、*wmq.example.com* と入力して、「OK」をクリックします。
  - ドメイン・コントローラーの管理者(このコンピューターをドメインに加わせる権限を持った管理者)の**ユーザー名とパスワード**を入力して、「OK」をクリックします。
  - "*wmq.example.com* ドメインへようこそ"メッセージに応答して、**OK > OK > クローズ > 即時再始動**をクリックします。
- このコンピューターが *wmq.example.com* ドメインのメンバーになっていることを確認します。
  - sun* で、ドメイン・コントローラーに *wmq\Administrator* としてログオンします。
  - サーバー・マネージャー > **Active Directory** ドメイン・サービス > *wmq.example.com* > **コンピューターズ**を開き、ウィンドウに *mars* が正しくリストされていることを確認します。
- IBM MQ for Windows を *mars* にインストールします。

IBM MQ for Windows インストール・ウィザードの実行について、詳しくは、[Windows への IBM MQ サーバーのインストール](#)を参照してください。

  - mars* に、ローカル管理者 *mars\Administrator* としてログオンします。
  - Setup** コマンドを IBM MQ for Windows インストール・メディアで実行します。

IBM MQ のランチパッド・アプリケーションが始動します。
  - 「ソフトウェア要件」をクリックして、前提ソフトウェアがインストールされていることを確認します。
  - 「ネットワーク構成」>「はい」をクリックして、ドメイン・ユーザー ID を構成します。

[446 ページの『Active Directory と DNS のドメインの作成 \(Windows\)』](#)の作業によって、この一連の作業のドメイン・ユーザー ID を構成します。

- e) 「**IBM MQ インストール**」をクリックし、インストール言語を選択し、「IBM MQ インストーラーの起動」をクリックします。
- f) 使用条件を確認し、「次へ」 > 「次へ」 > 「インストール」をクリックして、デフォルト構成を受け入れます。インストールが完了するのを待って、「完了」をクリックします。

インストール環境の名前を変更したり、別のコンポーネントをインストールしたり、キュー・マネージャーのデータとログのために別のディレクトリーを構成したり、別のディレクトリーにインストールしたりする場合があります。その場合は、「標準」ではなく「カスタム」をクリックします。

IBM MQ がインストールされ、インストーラーによって "IBM MQ" 準備ウィザードが開始されます。

**重要:** ウィザードはまだ実行しないでください。

5. IBM MQ サービスを実行するユーザーを構成して、「サービスとして実行」の権限を与えます。

その権限で IBM MQ サービスを実行するローカル mqm グループを構成するのか、Domain mqm グループを構成するのか、それともユーザーを構成するのかを選択します。この例では、ユーザーにその権限を与えます。

- a) 「スタート」 > 「ファイル名を指定して実行 ...」をクリックします。コマンド **secpol.msc** を入力し、「OK」をクリックします。
- b) 「セキュリティの設定」 > 「ローカル ポリシー」 > 「ユーザー権利の割り当て」を開きます。ポリシーのリストで、「サービスとしてログオン」 > 「プロパティ」を右クリックします。
- c) 「ユーザーまたはグループの追加...」をクリックします。 *wmuser1* と入力し、「名前の確認」をクリックします。
- d) ドメイン管理者 *wmq\Administrator* のユーザー名とパスワードを入力し、「OK」 > 「適用」 > 「OK」をクリックします。「ローカルセキュリティ ポリシー」ウィンドウを閉じます。

6. "IBM MQ" 準備ウィザードを実行します。

"IBM MQ" 準備ウィザードの実行について詳しくは、[IBM MQ 準備ウィザードを使用した IBM MQ の構成](#)を参照してください。

- a) IBM MQ インストーラーは、"Prepare IBM MQ" を自動的に実行します。

ウィザードを手動で開始するには、「スタート」 > 「すべてのプログラム」 > **IBM MQ** フォルダーで "準備 IBM MQ" へのショートカットを見つけます。複数インストール構成の場合は、IBM MQ のインストールに対応するショートカットを選択します。

- b) 「次」をクリックし、"ネットワーク内に Windows 2000 またはそれ以降のドメイン・コントローラーがあるかどうかを判別する"ために、応答として、**Yes** をクリックします。
- c) 最初の「IBM MQ for Windows for Windows ドメイン・ユーザーの構成」ウィンドウで、「はい」 > 「次へ」をクリックします。
- d) 2 番目の「IBM MQ for Windows for Windows ドメイン・ユーザーの構成」ウィンドウで、「ドメイン」フィールドに *wmq* と入力します。「ユーザー名」フィールドに *wmuser1* と入力し、「パスワード」フィールドにパスワード (設定した場合) を入力します。「次へ」をクリックします。

ウィザードによって、IBM MQ が構成され、*wmuser1* で開始されます。

- e) ウィザードの最終ページで、必要に応じて各種のチェック・ボックスを選択したりクリアしたりして、「完了」をクリックします。

## 次のタスク

1. 475 ページの『ローカル mqm グループによって許可されたデータ・ファイルとログ・ファイルの読み取りと書き込み』の作業を実行して、インストールと構成が正しく機能していることを検証します。
2. 453 ページの『キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)』の作業を実行して、複数インスタンス・キュー・マネージャーのデータ・ファイルとログ・ファイルのファイル共有を構成します。

## 関連タスク

**Windows** Active Directory と DNS のドメインの作成 (Windows)

**Windows** キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)

**Windows** 代替グローバル・セキュリティ・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み

## 関連情報

[IBM MQ Windows サービスに必要なユーザー権限](#)

**Windows** キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 (Windows)

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

### **sun**

Windows Server 2008 ドメイン・コントローラー。Sun、mars、および venus が含まれている *wmq.example.com* ドメインを所有します。例示目的で、これをファイル・サーバーとしても使用します。

### **mars**

最初の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの QMGR という 1 つのインスタンスを組み込みます。

### **venus**

2 番目の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの QMGR という 2 番目のインスタンスを組み込みます。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

## 始める前に

1. ここで取り上げる手順のとおりこの作業を実行するには、[446 ページの『Active Directory と DNS のドメインの作成 \(Windows\)』](#)の作業手順を実行し、ドメイン・コントローラー *sun* でドメイン *sun.wmq.example.com* を作成する必要があります。斜体の名前は、それぞれの構成に合わせて変更してください。

## このタスクについて

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。さらに、[441 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』](#)の作業も必要です。

この作業では、データとログのディレクトリーが含まれている共有フォルダーと、グローバル・グループを作成し、その共有フォルダーに対するアクセス権限をそのグローバル・グループに与えます。crtmqm コマンドでは、その共有フォルダーに対するアクセス権限を与えるグローバル・グループの名前を -a パラメーターで渡します。このグローバル・グループを使用すれば、この共有フォルダーのユーザーと他の共有フォルダーのユーザーを柔軟に分離できるようになります。そのような柔軟性が不要な場合は、新しいグローバル・グループを作成する代わりに、その共有フォルダーに対する権限を Domain mqm グループに与えます。

この作業で共有のために使用するグローバル・グループの名前は *wmqha*、共有フォルダーの名前は *wmq* です。それらの定義先は、Windows ドメイン *wmq.example.com* のドメイン・コントローラー *sun* です。

グローバル・グループ *wmqha* には、この共有フォルダーに関するフル・コントロール権限を与えます。この作業で斜体で表記している名前をそれぞれの好みの名前に置き換えてください。

この作業では、ドメイン・コントローラーとして、ファイル・サーバーと同じサーバーを使用します。実際のアプリケーション環境では、パフォーマンスと可用性の観点から、ディレクトリー・サービスとファイル・サービスを別々のサーバーに分離するようにしてください。

キュー・マネージャーを実行するユーザー ID は、2つのグループのメンバーとして構成する必要があります。つまり、IBM MQ サーバーのローカル *mqm* グループと、*wmqha* グローバル・グループの両方のメンバーにしなければなりません。

この一連の作業では、キュー・マネージャーをサービスとして実行するときにユーザー ID *wmquser1* を使用するので、*wmquser1* は *wmqha* のメンバーでなければなりません。キュー・マネージャーを対話式に実行するときにはユーザー ID *wmquser2* を使用するので、*wmquser2* も *wmqha* のメンバーでなければなりません。*wmquser1* も *wmquser2* も、グローバル・グループ Domain *mqm* のメンバーです。Domain *mqm* は、*mars* サーバーおよび *venus* IBM MQ サーバー上のローカル *mqm* グループのメンバーです。したがって、*wmquser1* と *wmquser2* は、両方の IBM MQ サーバーのローカル *mqm* グループのメンバーになります。

## 手順

- ドメイン管理者としてドメイン・コントローラー *sun.wmq.example.com* にログオンします。
- グローバル・グループ *wmqha* を作成します。
  - [サーバーマネージャー] > [ロール] > **Active Directory Domain Services** > *wmq.example.com* > [ユーザー]を開きます。
  - wmq.example.com\Users* フォルダーを開きます
  - 「ユーザー」 > 新規 > 「グループ」を右クリックします。
  - 「グループ名」 フィールドに *wmqha* と入力します。
  - 「グループの範囲」では、「グローバル」がクリックされたままの状態にしておき、「グループの種類」では、「セキュリティ」がクリックされたままの状態にしておきます。「OK」をクリックします。
- ドメイン・ユーザー *wmquser1* と *wmquser2* をグローバル・グループ *wmqha* に追加します。
  - サーバー・マネージャーのナビゲーション・ツリーで、**利用者**をクリックし、ユーザーのリストで ***wmqha* > プロパティ**を右クリックします。
  - 「*wmqha* のプロパティ」ウィンドウの「メンバー」タブをクリックします。
  - 「追加 ...」をクリックします。*wmquser1 ; wmquser2* と入力し、「**名前の確認**」 > 「OK」 > 「適用」 > 「OK」をクリックします。
- キュー・マネージャーのデータ・ファイルとログ・ファイルを格納するディレクトリー・ツリーを作成します。
  - コマンド・プロンプトを開きます。
  - 次のコマンドを入力します。

```
md c:\wmq\data, c:\wmq\logs
```
- グローバル・グループ *wmqha* に対して、*c:\wmq* ディレクトリーおよび共有に対する全制御権限を付与する権限を付与します。
  - Windows エクスプローラーで、**c: ¥ *wmq*** > 「プロパティ」を右クリックします。
  - 「セキュリティ」タブをクリックし、「詳細設定」 > 「編集...」をクリックします。
  - 「このオブジェクトの親からの継承可能なアクセス許可を含める」チェック・ボックスをクリアします。「Windows セキュリティ」ウィンドウで「コピー」をクリックします。

- d) 「アクセス許可エントリ」のリストで「Users」の各行を選択し、「削除」をクリックします。「アクセス許可エントリ」のリストにある「SYSTEM」、「Administrators」、「CREATOR OWNER」の各行はそのままにしておきます。
- e) 「追加 ...」をクリックします。グローバル・グループ *wmqha* の名前を入力します。「名前の確認」> 「OK」をクリックします。
- f) 「wmq のアクセス許可」ウィンドウの「アクセス許可」リストで「フルコントロール」を選択します。
- g) 「OK」> 「適用」> 「OK」> 「OK」> 「OK」をクリックします。
- h) Windows エクスプローラーで、**c: ¥ wmq** > 「共有 ...」を右クリックします。
- i) 「拡張共有 ...」をクリックします。次に、「このフォルダーを共有する」チェック・ボックスを選択します。共有名は、*wmq* のままにしておきます。
- j) 許可 > 「追加 ...」をクリックします。グローバル・グループ *wmqha* の名前を入力します。「名前の確認」> 「OK」をクリックします。
- k) 「グループ名またはユーザー名」リストで *wmqha* を選択します。**wmqha** の許可のリストにある「フルコントロール」チェック・ボックスを選択します。「適用」をクリックします
- l) 「グループ名またはユーザー名」リストで *Administrators* を選択します。「管理者の権限」のリストで「フルコントロール」チェック・ボックスを選択します。「適用」> 「OK」> 「OK」> 「閉じる」をクリックします。

## 次のタスク

それぞれの IBM MQ サーバーから、共有ディレクトリーのファイルを読み取ったり、共有ディレクトリーにファイルを書き込んだりする操作を実行できることを確認します。IBM MQ サービスのユーザー ID *wmquser1* と対話式ユーザー ID *wmquser2* を確認します。

1. リモート・デスクトップを使用している場合は、*mars* 上のローカル・グループ Remote Desktop Users に *wmq\wmquser1* および *wmquser2* を追加する必要があります。
  - a. *wmq\Administrator* として *mars* にログオンします
  - b. **lusrmgr.msc** コマンドを実行して、「ローカルユーザーとグループ」ウィンドウを開きます。
  - c. 「グループ」をクリックします。「リモート・デスクトップ・ユーザー」> 「プロパティ」> 「追加 ...」を右クリックします。*wmquser1 ; wmquser2* と入力し、「名前の確認」をクリックします。
  - d. ドメイン管理者 *wmq\Administrator* のユーザー名とパスワードを入力し、「OK」> 「適用」> 「OK」をクリックします。
  - e. 「ローカルユーザーとグループ」ウィンドウを閉じます。
2. *wmq\wmquser1* として *mars* にログオンします。
  - a. Windows エクスプローラーのウィンドウを開いて、**\\sun\wmq** と入力します。  
システムの応答として、*sun.wmq.example.com* の共有フォルダー *wmq* が開き、データとログのディレクトリーが表示されます。
  - b. データ・サブディレクトリーでファイルを作成したり、何かの内容を追加したり、その内容を読み取ったり、削除したりして、*wmquser1* に対するアクセス許可を確認します。
3. *wmq\wmquser2* として *mars* にログオンし、検査を繰り返します。
4. 次に、データとログの共有ディレクトリーを使用するキュー・マネージャーを作成する作業を実行します。456 ページの『[代替グローバル・セキュリティー・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み](#)』を参照してください。

## 関連タスク

**Windows** Active Directory と DNS のドメインの作成 (Windows)

**Windows** Windows ドメイン内のサーバーまたはワークステーションに IBM MQ をインストールする操作

**Windows** 代替グローバル・セキュリティ・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み

**Windows** 代替グローバル・セキュリティ・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み

この作業では、**crtmqm** コマンドで **-a** フラグを使用する方法を確認します。**-a** フラグを使用すると、キュー・マネージャーは、代替セキュリティ・グループを使用して、リモート・ファイル共有フォルダーにあるログ・ファイルとデータ・ファイルにアクセスできるようになります。

実動スケール構成では、構成を既存のドメインに合わせて調整しなければならない可能性があります。例えば、さまざまなドメイン・グループを定義して、それぞれに異なる共有を許可したり、キュー・マネージャーを実行するユーザー ID をグループ化したりします。

この構成例では、3 つのサーバーを使用します。

#### **sun**

Windows Server 2008 ドメイン・コントローラー。Sun、mars、および venus が含まれている *wmq.example.com* ドメインを所有します。例示目的で、これをファイル・サーバーとしても使用します。

#### **mars**

最初の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの QMGR という 1 つのインスタンスを組み込みます。

#### **venus**

2 番目の IBM MQ サーバーとして使用する Windows Server 2008。複数インスタンス・キュー・マネージャーの QMGR という 2 番目のインスタンスを組み込みます。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

## 始める前に

以下の作業手順を実行します。これらのタスクは、ドメイン・コントローラーおよびドメインを作成し、サーバーに IBM MQ for Windows をインストールし、データおよびログ・ファイル用のファイル共有を作成します。既存のドメイン・コントローラーを構成する場合でも、新しい Windows Server 2008 でこれらの手順を試してみる価値はあります。それぞれのドメインに合わせてこれらの手順を調整できます。

1. [446 ページの『Active Directory と DNS のドメインの作成 \(Windows\)』](#).
2. [450 ページの『Windows ドメイン内のサーバーまたはワークステーションに IBM MQ をインストールする操作』](#).
3. [453 ページの『キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 \(Windows\)』](#).

## このタスクについて

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。さらに、[441 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』](#)の作業も必要です。

この作業では、ファイル・サーバーのリモート・ディレクトリーにデータとログを格納するキュー・マネージャーを作成します。この例では、ファイル・サーバーとして、ドメイン・コントローラーと同じサーバーを使用します。データ・フォルダーとログ・フォルダーを含むディレクトリーは、グローバル・グループ *wmqha* に付与された完全な制御権限で共有されます。

## 手順

1. ローカル管理者 `mars\Administrator` として、ドメイン・サーバー `mars` にログオンします。
2. コマンド・ウィンドウを開きます。
3. IBM MQ サービスを再開します。

サービスを実行するユーザー ID で、ここで構成した追加のセキュリティー資格情報を有効にするには、サービスを再始動する必要があります。

次のコマンドを入力します。

```
endmqsvc  
strmqsvc
```

システム応答:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

および:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' started successfully.
```

4. キュー・マネージャーを作成します。

```
crtmqm -a wmq\wmqha -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\sun\wmq\data -ld \\sun\wmq\logs  
QMGR
```

グローバル・グループ `wmq\wmqha` の完全なドメイン名を指定することにより、代替セキュリティー・グループ `wmqha` のドメイン `wmq` を指定する必要があります。

共有 `\\sun\wmq` の汎用命名規則(UNC)を入力し、マップされたドライブ参照を使用しないようにする必要があります。

システム応答:

```
IBM MQ キュー・マネージャーが作成されました。  
Directory '&#xa5;&#xa5;sun&#xa5;wmq&#xa5;data&#xa5;QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
デフォルトのオブジェクト統計 : 74 が作成された 0 が置き換えられた  
Completing setup.  
Setup completed.
```

## 次のタスク

キューにメッセージを書き込み、キューからメッセージを取り出して、キュー・マネージャーをテストします。

1. キュー・マネージャーを始動します。

```
strmqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.
```

```
The queue manager is associated with installation '1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
IBM MQ queue manager 'QMGR' started using 7.1.0.0.
```

2. テスト・キューを作成します。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

システム応答:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)
AMQ8006: IBM MQ queue created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. サンプル・プログラム **amqsput** を使用してテスト・メッセージを書き込みます。

```
echo 'A test message' | amqsput QTEST QMGR
```

システム応答:

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. サンプル・プログラム **amqsget** を使用してテスト・メッセージを取り出します。

```
amqsget QTEST QMGR
```

システム応答:

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. キュー・マネージャーを停止させます。

```
endmqm -i QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. キュー・マネージャーを削除してください。

```
dltmqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' deleted.
```

- 作成したディレクトリーを削除します。

**ヒント:**それぞれのファイルやディレクトリーの削除に関する確認プロンプトが表示されないようにするには、コマンドに /Q オプションを追加します。

```
del /F /S C:\wmq\*. *
rmdir /S C:\wmq
```

## 関連タスク

**Windows** [Active Directory と DNS のドメインの作成 \(Windows\)](#)

**Windows** [Windows ドメイン内のサーバーまたはワークステーションに IBM MQ をインストールする操作](#)

**Windows** [キュー・マネージャーのデータ・ファイルとログ・ファイルのための共有ディレクトリーの作成 \(Windows\)](#)

**Windows** [Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成](#)  
Windows のドメイン・コントローラーで複数インスタンス・キュー・マネージャーをセットアップする例を取り上げます。ここで取り上げるのは、実動スケールのセットアップではなく、関連する概念を説明する程度のセットアップです。この例は、Windows Server 2008 をベースとしています。それぞれのステップは、Windows Server の他のバージョンでは異なる場合があります。

構成では、ミニ・ドメインの概念、または"ドメインレット"の概念を使用します。[Windows 2000、WindowsServer 2003、および WindowsServer 2008 クラスタ・ノードをドメイン・コントローラーとして、を参照してください。](#) [複数インスタンス・キュー・マネージャーを既存のドメインに追加する場合は、443 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 \(Windows\)』を参照してください。](#)

この構成例では、3つのサーバーを使用します。

### sun

第1のドメイン・コントローラーとして使用する Windows Server 2008 サーバーです。 *sun*、*earth*、*mars* が含まれている *wmq.example.com* ドメインを定義します。複数インスタンス・キュー・マネージャーの *QMGR* という1つのインスタンスを組み込みます。

### earth

第2のドメイン・コントローラー IBM MQ サーバーとして使用する Windows Server 2008 です。複数インスタンス・キュー・マネージャーの *QMGR* という2番目のインスタンスを組み込みます。

### mars

ファイル・サーバーとして使用する Windows Server 2008 です。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

## 始める前に

- Windows では、キュー・マネージャー・データとログ・ファイルを保管するファイル・システムを検証する必要はありません。検査の手順、共有ファイル・システムの動作の検証は、UNIX and Linux に適用されます。Windows では検査は必ず成功します。
- [446 ページの『Active Directory と DNS のドメインの作成 \(Windows\)』](#)の手順を実行して、第1のドメイン・コントローラーを作成します。

3. 463 ページの『[サンプル・ドメインへの 2 番目の Windows ドメイン・コントローラーの追加](#)』の手順を実行して、第 2 のドメイン・コントローラーを追加し、両方のドメイン・コントローラーに IBM MQ for Windows をインストールし、インストール環境を検証します。
4. 465 ページの『[サンプル・ドメインでの Windows ドメイン・コントローラーへの IBM MQ のインストール](#)』の手順を実行して、2 つのドメイン・コントローラーに IBM MQ をインストールします。

## このタスクについて

同じドメインにあるファイル・サーバーで、キュー・マネージャーのログ・ディレクトリーとデータ・ディレクトリーのための共有フォルダーを作成します。次に、そのファイル共有フォルダーを使用する複数インスタンス・キュー・マネージャーの最初のインスタンスをどちらかのドメイン・コントローラーで作成します。もう 1 つのインスタンスをもう 1 つのドメイン・コントローラーで作成し、最後に構成を検証します。ドメイン・コントローラーでファイル共有フォルダーを作成することも可能です。

この例では、*sun* が第 1 のドメイン・コントローラー、*earth* が第 2 のドメイン・コントローラー、*mars* がファイル・サーバーです。

## 手順

1. キュー・マネージャーのデータ・ファイルとログ・ファイルを格納するディレクトリーを作成します。
  - a) *mars* で、次のコマンドを入力します。

```
md c:\wmq\data , c:\wmq\logs
```

2. キュー・マネージャーのデータ・ファイルとログ・ファイルを格納するディレクトリーの共有を設定します。

ドメイン・ローカル・グループ *mqm* と、キュー・マネージャーを作成するために使用するユーザー ID にフル・コントロール権限を与える必要があります。この例の場合、Domain Administrators のメンバーになっているユーザー ID には、キュー・マネージャーを作成する権限があります。

ファイル共有は、ドメイン・コントローラーと同じドメインにあるサーバーで設定する必要があります。この例の場合、サーバー *mars* は、ドメイン・コントローラーと同じドメインにあります。

- a) Windows エクスプローラーで、**c: ¥ *wmq*** > 「プロパティ」を右クリックします。
- b) 「セキュリティ」タブをクリックし、「詳細設定」 > 「編集...」をクリックします。
- c) 「このオブジェクトの親からの継承可能なアクセス許可を含める」チェック・ボックスをクリアします。「Windows セキュリティ」ウィンドウで「コピー」をクリックします。
- d) 「アクセス許可エントリ」のリストで「Users」の各行を選択し、「削除」をクリックします。「アクセス許可エントリ」のリストにある「SYSTEM」、「Administrators」、「CREATOR OWNER」の各行はそのままにしておきます。
- e) 「追加 ...」をクリックします。ドメイン・ローカル・グループ *mqm* の名前を入力します。「名前の確認」をクリックします。
- f) 「Windows セキュリティー」ウィンドウに対する応答として、「Domain Administrator」の名前とパスワードを入力して、「OK > OK」をクリックします。
- g) 「wmq のアクセス許可」ウィンドウの「アクセス許可」リストで「フルコントロール」を選択します。
- h) 「OK」 > 「適用」 > 「OK」 > 「OK」 > 「OK」をクリックします。
- i) e から h までの手順を繰り返して、Domain Administrators のメンバーを追加します。
- j) Windows エクスプローラーで、**c: ¥ *wmq*** > 「共有 ...」を右クリックします。
- k) 「拡張共有 ...」をクリックします。次に、「このフォルダーを共有する」チェック・ボックスを選択します。共有名は、*wmq* のままにしておきます。
- l) 許可 > 「追加 ...」をクリックします。ドメイン・ローカル・グループ *mqm* ; Domain Administrators の名前を入力します。「名前の確認」をクリックします。

m) 「Windows セキュリティー」 ウィンドウに対する応答として、「Domain Administrator」の名前とパスワードを入力して、「OK > OK」をクリックします。

3. 第1のドメイン・コントローラー *sun* でキュー・マネージャー *QMGR* を作成します。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\mars\wmq\data -ld \\mars\wmq\logs QMGR
```

システム応答:

```
IBM MQ queue manager created.  
Directory '\\mars\wmq\data\QMGR' created.  
The queue manager is associated with installation 'Installation1'.  
Creating or replacing default objects for queue manager 'QMGR'.  
Default objects statistics : 74 created. 0 replaced. 0 failed.  
Completing setup.  
Setup completed.
```

4. *sun* でキュー・マネージャーを開始して、スタンバイ・インスタンスにします。

```
strmqm -x QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation 'Installation1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using 7.1.0.0.
```

5. *earth* で *QMGR* の2番目のインスタンスを作成します。

a) Prefix パラメーターと InstallationName パラメーターの値が *earth* の正しい値になっていることを確認します。

*sun* で **dspmqlinf** コマンドを実行します。

```
dspmqlinf QMGR
```

システム応答:

```
QueueManager:  
Name=QMGR  
Directory=QMGR  
Prefix=C:\ProgramData\IBM\MQ  
DataPath=\\mars\wmq\data\QMGR  
InstallationName=Installation1
```

b) 機械可読形式の **QueueManager** スタンザをクリップボードにコピーします。

*sun* で **dspmqlinf** コマンドを再び実行します。今回は、**-o command** パラメーターを付けます。

```
dspmqlinf -o command QMGR
```

システム応答:

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

- c) *earth* で **addmqinf** コマンドを実行し、クリップボードの情報を使用して、キュー・マネージャーのインスタンスを *earth* で作成します。

必要に応じて、Prefix パラメーターまたは InstallationName パラメーターの違いに合わせてコマンドを調整します。

```
addmqinf -s QueueManager -v Name= QMGR
-v Directory= QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

IBM MQ configuration information added.

6. *earth* で、キュー・マネージャーのスタンバイ・インスタンスを開始します。

```
strmqm -x QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started. The active
instance is running elsewhere.
```

## タスクの結果

キュー・マネージャーが *sun* から *earth* に切り替わることを確認します。

1. *sun* で、次のコマンドを実行します。

```
endmqm -i -r -s QMGR
```

*sun* のシステム応答:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended, permitting switchover to
a standby instance.
```

2. *earth* で、以下のコマンドを繰り返し入力します。

```
dspmq
```

システム応答:

```
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running)
```

## 次のタスク

サンプル・プログラムを使用して複数インスタンス・キュー・マネージャーを検証するには、[467 ページの『Windows での複数インスタンス・キュー・マネージャーの検査』](#)を参照してください。

### 関連タスク

[463 ページの『サンプル・ドメインへの 2 番目の Windows ドメイン・コントローラーの追加』](#)

[465 ページの『サンプル・ドメインでの Windows ドメイン・コントローラーへの IBM MQ のインストール』](#)

### 関連情報

[Windows 2000、Windows Server 2003、および Windows Server 2008 クラスター・ノードをドメイン・コントローラーとして構成する](#)

**Windows** サンプル・ドメインへの 2 番目の *Windows* ドメイン・コントローラーの追加  
*wmq.example.com* ドメインに第 2 のドメイン・コントローラーを追加して、それらのドメイン・コントローラーとファイル・サーバーで複数インスタンス・キュー・マネージャーを実行するための *Windows* ドメインを構成します。

この構成例では、3 つのサーバーを使用します。

#### **sun**

第 1 のドメイン・コントローラーとして使用する *Windows Server 2008* サーバーです。 *sun*、*earth*、*mars* が含まれている *wmq.example.com* ドメインを定義します。複数インスタンス・キュー・マネージャーの *QMGR* という 1 つのインスタンスを組み込みます。

#### **earth**

第 2 のドメイン・コントローラー *IBM MQ* サーバーとして使用する *Windows Server 2008* です。複数インスタンス・キュー・マネージャーの *QMGR* という 2 番目のインスタンスを組み込みます。

#### **mars**

ファイル・サーバーとして使用する *Windows Server 2008* です。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

## 始める前に

1. [446 ページの『Active Directory と DNS のドメインの作成 \(Windows\)』](#)の手順を実行して、ドメイン *wmq.example.com* のドメイン・コントローラー *sun* を作成します。斜体の名前は、それぞれの構成に合わせて変更してください。
2. デフォルトのワークグループ *WORKGROUP* に含まれているサーバーに *Windows Server 2008* をインストールします。この例では、そのサーバーの名前は *earth* です。

## このタスクについて

この例では、*earth* という *Windows Server 2008* を *wmq.example.com* ドメインの第 2 のドメイン・コントローラーとして構成します。

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。さらに、[441 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』](#)の作業も必要です。

## 手順

1. ドメイン・コントローラー *sun.wmq.example.com* を DNS サーバーとして *earth* に追加します。
  - a) *earth* で、*earth\Administrator* としてログオンし、**スタート**をクリックします。
  - b) 「ネットワーク」 > 「プロパティ」 > 「ネットワーク接続の管理」を右クリックします。
  - c) ネットワーク・アダプターを右クリックして、「プロパティ」をクリックします。

システムの応答として、「ローカルエリア接続のプロパティ」ウィンドウが開き、接続で使用する項目のリストが表示されます。

- d) 「ローカル・エリア接続のプロパティ」ウィンドウの項目リストから、**Internet Protocol** 「バージョン 4」または **Internet Protocol** 「バージョン 6」を選択します。「プロパティ」 > 「拡張...」をクリックします。「DNS」タブをクリックします。
  - e) DNS サーバー・アドレスで、「追加...」をクリックします。
  - f) DNS サーバーでもあるドメイン・コントローラーの IP アドレスを入力して、「追加」をクリックします。
  - g) 「以下の DNS サフィックスを順に追加する」 > 「追加...」をクリックします。
  - h) *wmq.example.com* と入力して、「追加」をクリックします。
  - i) 「この接続の DNS サフィックス」フィールドに *wmq.example.com* と入力します。
  - j) 「この接続のアドレスを DNS に登録する」と「この接続の DNS サフィックスを DNS 登録に使う」を選択します。「OK」 > 「OK」 > 「閉じる」をクリックします。
  - k) コマンド・ウィンドウで、**ipconfig /all** というコマンドを入力して、TCP/IP 設定を確認します。
2. ローカル管理者または Workgroup 管理者として、ドメイン・コントローラー *sun* にログオンします。  
サーバーが既にドメイン・コントローラーとして構成されている場合は、ドメイン管理者としてログオンする必要があります。
  3. 「Active Directory ドメイン サービス」ウィザードを実行します。
    - a) クリックスタート > 実行... **dcpromo** と入力して、「OK」をクリックします。  
Active Directory のバイナリー・ファイルがインストールされていない場合は、Windows によってそれらのファイルが自動的にインストールされます。
  4. *earth* を *wmq.example.com* ドメインの第 2 のドメイン・コントローラーとして構成します。
    - a) ウィザードの最初のウィンドウで、「詳細モード インストールを使用する」チェック・ボックスは、クリアされた状態のままにしておきます。「次へ」 > 「次へ」をクリックし、「既存のドメインにドメインコントローラを追加する」 > 「次へ」をクリックします。
    - b) 「このフォレスト内の任意のドメインの名前を入力してください...」に *wmq* と入力します。フィールド。「代替の資格情報」ラジオ・ボタンがクリックされた状態になっていることを確認し、「設定...」をクリックします。ドメイン管理者の名前とパスワードを入力し、「OK」 > 「次へ」 > 「次へ」 > 「次へ」をクリックします。
    - c) 「追加のドメインコントローラ オプション」ウィンドウで、「DNS サーバー」オプションと「グローバルカタログ」オプションが選択された状態になります。それらのオプションをそのまま受け入れ、「次へ」 > 「次へ」をクリックします。
    - d) 「ディレクトリー サービス復元モード Administrator パスワード」ウィンドウの「パスワード」と「パスワードの確認入力」にパスワードを入力します。「次へ」 > 「次へ」をクリックします。
    - e) 「ネットワーク資格情報」のプロンプト画面が表示されたら、ドメイン管理者のパスワードを入力します。ウィザードの最後のウィンドウで「完了時に再起動する」を選択します。
    - f) しばらくすると、DNS の委任に関する **DCPromo** エラーのウィンドウが表示されることがあります。「OK」をクリックします。サーバーがリブートします。

## タスクの結果

*earth* がリブートしたら、ドメイン管理者としてログオンします。*wmq.example.com* ドメインが *earth* に複製されていることを確認します。

## 次のタスク

IBM MQ のインストールに進みます。465 ページの『[サンプル・ドメインでの Windows ドメイン・コントローラーへの IBM MQ のインストール](#)』を参照してください。

## 関連タスク

**Windows** サンプル・ドメインでの *Windows* ドメイン・コントローラーへの *IBM MQ* のインストール  
446 ページの『[Active Directory と DNS のドメインの作成 \(Windows\)](#)』

**Windows** サンプル・ドメインでの *Windows* ドメイン・コントローラーへの *IBM MQ* のインストール  
*wmq.example.com* ドメイン内の両方のドメイン・コントローラーに *IBM MQ* のインストールをインストールおよび構成します。  
ここに簡略説明を入力してください。先頭パラグラフと要約に使用されます。

この構成例では、3つのサーバーを使用します。

### **sun**

第1のドメイン・コントローラーとして使用する *Windows Server 2008* サーバーです。 *sun*、*earth*、*mars* が含まれている *wmq.example.com* ドメインを定義します。複数インスタンス・キュー・マネージャーの *QMGR* という1つのインスタンスを組み込みます。

### **earth**

第2のドメイン・コントローラー *IBM MQ* サーバーとして使用する *Windows Server 2008* です。複数インスタンス・キュー・マネージャーの *QMGR* という2番目のインスタンスを組み込みます。

### **mars**

ファイル・サーバーとして使用する *Windows Server 2008* です。

この例で斜体で表記している名前をそれぞれ任意の名前に置き換えてください。

## 始める前に

- 446 ページの『[Active Directory と DNS のドメインの作成 \(Windows\)](#)』の手順を実行して、ドメイン *wmq.example.com* のドメイン・コントローラー *sun* を作成します。斜体の名前は、それぞれの構成に合わせて変更してください。
- 463 ページの『[サンプル・ドメインへの2番目の Windows ドメイン・コントローラーの追加](#)』の手順を実行して、ドメイン *wmq.example.com* の第2のドメイン・コントローラー *earth* を作成します。斜体の名前は、それぞれの構成に合わせて変更してください。
- IBM MQ* を実行できるその他の *Windows* バージョンについては、[Windows システムでのハードウェア要件およびソフトウェア要件](#) を参照してください。

## このタスクについて

*wmq.example.com* ドメイン内の両方のドメイン・コントローラーに *IBM MQ* のインストールをインストールおよび構成します。

## 手順

- sun* と *earth* で *IBM MQ* をインストールします。

*IBM MQ for Windows* インストール・ウィザードの実行について、詳しくは、[Windows への \*IBM MQ\* サーバーのインストール](#) を参照してください。

- sun* と *earth* の両方で、ドメイン管理者 *wmq\Administrator* としてログオンします。
- Setup** コマンドを *IBM MQ for Windows* インストール・メディアで実行します。

*IBM MQ* のランチパッド・アプリケーションが始動します。

- 「ソフトウェア要件」をクリックして、前提ソフトウェアがインストールされていることを確認します。
- 「ネットワーク構成」 > 「いいえ」をクリックします。

このインストール環境では、ドメイン・ユーザー ID を構成してもしなくてもかまいません。作成されるユーザー ID は、ドメイン・ローカル・ユーザー ID になります。

- e) 「**IBM MQ インストール**」をクリックし、インストール言語を選択し、「IBM MQ インストーラーの起動」をクリックします。
- f) 使用条件を確認し、「次へ」 > 「次へ」 > 「インストール」をクリックして、デフォルト構成を受け入れます。インストールが完了するのを待って、「完了」をクリックします。

インストール環境の名前を変更したり、別のコンポーネントをインストールしたり、キュー・マネージャーのデータとログのために別のディレクトリーを構成したり、別のディレクトリーにインストールしたりする場合は、「標準」ではなく「**カスタム**」をクリックします。

IBM MQ がインストールされ、インストーラーによって "IBM MQ" 準備ウィザードが開始されます。

IBM MQ for Windows のインストールで、ドメイン・ローカル・グループ `mqm` とドメイン・グループ `Domain mqm` が構成されます。Domain `mqm` は、`mqm` のメンバーになります。その同じドメインにドメイン・コントローラーを追加すると、そのドメイン・コントローラーも、`mqm` グループと `Domain mqm` グループを共有します。

2. `earth` と `sun` の両方で、"IBM MQ" 準備ウィザードを実行します。

"IBM MQ" 準備ウィザードの実行について詳しくは、[IBM MQ 準備ウィザードを使用した IBM MQ の構成](#)を参照してください。

- a) IBM MQ インストーラーは、"準備 IBM MQ" を自動的に実行します。

ウィザードを手動で開始するには、「スタート」 > 「すべてのプログラム」 > **IBM MQ** フォルダーで "準備 IBM MQ" へのショートカットを見つけます。複数インストール構成の場合は、IBM MQ のインストールに対応するショートカットを選択します。

- b) 「次」をクリックして、質問に答えていいえを「クリック」してください。"ネットワーク内に Windows 2000 またはそれ以降のドメイン・コントローラーがあるかどうかを確認する"<sup>1</sup>。
- c) ウィザードの最終ページで、必要に応じて各種のチェック・ボックスを選択したりクリアしたりして、「完了」をクリックします。

"「IBM MQ」準備" ウィザードは、最初のドメイン・コントローラー上にドメイン・ローカル・ユーザー `MUSR_MQADMIN` を作成し、2 番目のドメイン・コントローラー上に別のドメイン・ローカル・ユーザー `MUSR_MQADMIN1` を作成します。さらに、このウィザードによって、各コントローラーに IBM MQ サービスが作成され、このサービスにログオンするユーザーが `MUSR_MQADMIN` または `MUSR_MQADMIN1` になります。

3. キュー・マネージャーを作成する権限を持ったユーザーを定義します。

このユーザーは、ローカル・ログオンの権限を持っていなければならない、ドメイン・ローカル・グループ `mqm` のメンバーでなければなりません。ドメイン・コントローラーの場合、ローカル・ログオンの権限は、ドメイン・ユーザーではなく管理者にあります。デフォルトでその両方の属性を兼ね備えているユーザーはいません。この作業では、ドメイン・ローカル・グループ `mqm` にドメイン管理者を追加します。

- a) [サーバーマネージャー] > [ロール] > **Active Directory Domain Services** > `wmq.example.com` > [ユーザー]を開きます。
- b) ドメイン管理者 > 「グループに追加 ...」を右クリックします。次に `mqm` と入力します。「名前の確認」 > **OK** > 「OK」をクリックします。

## タスクの結果

1. "準備 IBM MQ" によってドメイン・ユーザー `MUSR_MQADMIN` が作成されたことを確認します。

- a. [サーバーマネージャー] > [ロール] > **Active Directory Domain Services** > `wmq.example.com` > [ユーザー]を開きます。

---

<sup>1</sup> ドメインのインストールを構成することができます。ドメイン・コントローラーに存在するすべてのユーザーとグループのスコープはドメイン・スコープなので、どちらにしても違いはありません。IBM MQ をドメイン内にはないものとして、インストールする方が簡単です。

- b. 右クリック **MUSR\_MQADMIN** > **プロパティ...** > 次のメンバー。これが Domain users および mqm のメンバーであることを確認します。
2. MUSR\_MQADMIN に、サービスとして実行する権限があることを確認します。
  - a. 「スタート」 > 「ファイル名を指定して実行...」をクリックします。コマンド **secpol.msc** を入力し、「OK」をクリックします。
  - b. 「セキュリティの設定」 > 「ローカル ポリシー」 > 「ユーザー権利の割り当て」を開きます。ポリシーのリストで、「サービスとしてログオン」 > 「プロパティ」 サービスとしてログオンする権限があるユーザーとして MUSR\_MQADMIN が表示されることを確認します。「OK」をクリックします。

## 次のタスク

1. 475 ページの『ローカル mqm グループによって許可されたデータ・ファイルとログ・ファイルの読み取りと書き込み』の作業を実行して、インストールと構成が正しく機能していることを検証します。
2. 459 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』の作業に戻り、ドメイン・コントローラーで複数インスタンス・キュー・マネージャーを構成する作業を完了します。

## 関連タスク

**Windows** サンプル・ドメインへの 2 番目の Windows ドメイン・コントローラーの追加

## 関連情報

[IBM MQ Windows サービスに必要なユーザー権限](#)

**Windows** Windows での複数インスタンス・キュー・マネージャーの検査

複数インスタンス・キュー・マネージャー構成を検査するには、サンプル・プログラム **amqsgbac**、**amqspbac**、および **amqsmbac** を使用します。このトピックでは、Windows Server 2003 上で複数インスタンス・キュー・マネージャーの構成を検査するためのサンプル構成を記載しています。

高可用性サンプル・プログラムは、クライアントの自動再接続を使用します。接続されたキュー・マネージャーに障害が起きた場合、クライアントは同じキュー・マネージャー・グループ内のキュー・マネージャーへの再接続を試行します。サンプルの説明、高可用性のサンプル・プログラムでは、単純にするために単一インスタンス・キュー・マネージャーを使用してクライアントの再接続を説明しています。複数インスタンス・キュー・マネージャーがある同じサンプルを使用して、複数インスタンス・キュー・マネージャー構成を検査することができます。

このサンプルでは、459 ページの『Windows ドメイン・コントローラーでの複数インスタンス・キュー・マネージャーの作成』で説明されている複数インスタンス構成を使用しています。この構成を使用して、複数インスタンス・キュー・マネージャーがスタンバイ・インスタンスに切り替わることを検査します。**endmqm** コマンドを使用してキュー・マネージャーを停止し、**-s** 切り替えオプションを使用します。クライアント・プログラムは、新規キュー・マネージャー・インスタンスに再接続して、少しの遅延の後に新規インスタンスの処理を続行します。

クライアントは、Windows7 Service Pack 1 を実行している 400MB の VMware イメージにインストールされています。セキュリティ上の理由から、マルチインスタンス・キュー・マネージャーを実行しているドメイン・サーバーと同じ VMware ホスト専用ネットワーク上で接続されています。構成を単純化するために、クライアント接続テーブルが含まれている /MQHA フォルダを共有します。

## IBM MQ Explorer を使用したフェイルオーバーの検査

フェイルオーバーを検査するためにサンプル・アプリケーションを使用する前に、IBM MQ Explorer を各サーバー上で実行します。「リモート・キュー・マネージャーの追加」 > 「複数インスタンス・キュー・マネージャーへの直接接続」ウィザードを使用して、両方のキュー・マネージャー・インスタンスを各エクスプローラーに追加します。両方のインスタンスをスタンバイを許可して実行しておきます。アクティブ・インスタンスがある VMware イメージを実行しているウィンドウを閉じて仮想的にサーバーの電源をオフにするか、またはアクティブ・インスタンスを停止して、スタンバイ・インスタンスおよび再接続する再接続可能クライアントへの切り替えを許可します。



**重要:** サーバーの電源をオフにする場合は、これが MQHA フォルダをホスティングしているものではないことを確認してください。

注: 「キュー・マネージャーの停止」ダイアログで「スタンバイ・インスタンスへの切り替えを許可」オプションが表示されない場合があります。このオプションがないのは、キュー・マネージャーが単一インスタンス・キュー・マネージャーとして実行されているためです。それを開始したときに、「スタンバイ・インスタンスの許可」オプションを指定しませんでした。キュー・マネージャーの停止要求が拒否された場合は、「詳細」ウィンドウを参照してください。実行中のスタンバイ・インスタンスがない可能性があります。

## サンプル・プログラムを使用したフェイルオーバーの検査

### アクティブ・インスタンスを実行するサーバーの選択

MQHA ディレクトリーまたはファイル・システムのホストになるサーバーが1つ選択されている場合があります。アクティブ・サーバーを実行中の VMware ウィンドウを閉じることによってフェイルオーバーをテストする予定であれば、それが MQHA のホストになるサーバーではないことを確認してください。

### アクティブ・キュー・マネージャー・インスタンスを実行するサーバー上

1. `ipaddr1` および `ipaddr2` を変更し、以下のコマンドを `N:\hasample.tst.` に保管する

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME(' ipaddr1 (1414), ipaddr2 (1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

注: **MCAUSER** パラメーターをブランクのままにすると、クライアント・ユーザー ID がサーバーに送信されます。クライアント・ユーザー ID には、サーバーに対する正しいアクセス権が必要です。別の方法として、SVRCONN チャンネル内の **MCAUSER** パラメーターを、サーバー上で構成済みのユーザー ID に設定します。

2. パス `N:\` を指定して、コマンド・プロンプトを開き、以下のコマンドを実行します:

```
runmqsc -m QM1 < hasample.tst
```

3. **runmqsc** コマンドの出力を検査することで、リスナーが実行中であり、キュー・マネージャー制御があることを確認します。

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

または、IBM MQ Explorer を使用して、TCP/IP リスナーが実行中で、Control = Queue Manager があることを確認します。

### クライアント上

1. サーバー上の共有ディレクトリー `C:\MQHA`、クライアント上の `N:\` にマップします。
2. パス `N:\` を指定してコマンド・プロンプトを開きます。次のようにして、環境変数 `MQCHLLIB` を、サーバー上のクライアント・チャンネル定義テーブル (CCDT) を指すように設定します。

```
SET MQCHLLIB=N:\data\QM1\@ipcc
```

3. コマンド・プロンプトで、以下のコマンドを入力します。

```
start amqsgnac TARGET QM1
```

```
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

**注:**問題がある場合は、コマンド・プロンプトでアプリケーションを開始し、理由コードがコンソールに出力されるようにするか、N:\data\QM1\errors フォルダー内の AMQERR01.LOG ファイルを調べてください。

## アクティブ・キュー・マネージャー・インスタンスを実行するサーバー上

### 1. 次のいずれかの場合:

- アクティブ・サーバー・インスタンスがある VMware イメージを実行するウィンドウを閉じます。
- IBM MQ Explorer を使用して、アクティブ・キュー・マネージャー・インスタンスを停止して、スタンバイ・インスタンスへの切り替えを許可し、再接続可能クライアントに再接続を指示します。

### 2. 最終的には 3 つのクライアントで接続が切断していることが検出され、次いで再接続されます。この構成では、サーバー・ウィンドウを閉じた場合、3 つの接続がすべて再確立されるには約 7 分かかります。他の接続よりも速く再確立される接続もあります。

## 結果

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsgnac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

## Windows 共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護

### (Windows)

このトピックでは、グローバル代替セキュリティ・グループを使用して、キュー・マネージャーのデータ・ファイルとログ・ファイルの共有場所を保護する方法を取り上げます。別々のサーバーで実行するキュー・マネージャーの複数インスタンスで、その場所を共有できます。

通常、キュー・マネージャーのデータ・ファイルとログ・ファイルの共有格納場所をセットアップすることはありません。IBM MQ for Windows をインストールするときに、そのサーバーで作成するキュー・マネージャーのために指定したホーム・ディレクトリーがインストール・プログラムによって作成されます。キュー・マネージャーのディレクトリーは、ローカル mqm グループによって保護され、それらのディレクトリーにアクセスする IBM MQ サービスのユーザー ID が構成されます。

セキュリティ・グループを使用して共有フォルダーを保護する場合、そのフォルダーにアクセスする権限を持ったユーザーには、そのグループの資格情報が必要になります。例えば、mars というサーバーのローカル mqm グループによって、リモート・ファイル・サーバーにあるフォルダーを保護するとしましょう。その場合は、キュー・マネージャーのプロセスを実行するユーザーを mars のローカル mqm グループのメンバーにします。そのユーザーには、リモート・ファイル・サーバーにあるフォルダーの資格情報に合致する資格情報を設定します。その資格情報を使用して、キュー・マネージャーは、そのフォルダーにあるデータ・ファイルとログ・ファイルにアクセスできます。別のサーバーでキュー・マネージャーのプロセスを実行するユーザーは、別のローカル mqm グループのメンバーであり、資格情報が合致しません。mars とは異なるサーバーでキュー・マネージャーを実行すると、そのキュー・マネージャーは、mars での稼働時に作成したデータ・ファイルとログ・ファイルにアクセスできません。そのユーザーをドメイン・ユーザーにしたとしても、そのユーザーの資格情報は異なります。そのユーザーは、mars のローカル mqm グループから資格情報を取得する必要があり、別のサーバーからその資格情報を取得することはできないからです。

この問題は、キュー・マネージャーのためにグローバル代替セキュリティ・グループを設定することによって解決できます。471 ページの図 75 を参照してください。つまり、グローバル・グループによってリモート・フォルダーを保護するということです。mars でキュー・マネージャーを作成するときに、キュー・マネージャーにそのグローバル・グループの名前を渡します。そのグローバル・グループ名を代替セキュリティ・グループとして渡すときには、**crtmqm** コマンドの **-a[r]** パラメーターを使用します。キュー・マネージャーを別のサーバーに移動する場合は、そのセキュリティ・グループの名前も一緒に移動します。具体的には、qm.ini ファイルの **AccessMode** スタンザにある **SecurityGroup** でその名前を移動することになります。例えば、以下のとおりです。

```
AccessMode:  
SecurityGroup=wmq\wmq
```

qm.ini の **AccessMode** スタンザには、**RemoveMQMAccess** も含まれます。例えば、以下のとおりです。

```
AccessMode:  
RemoveMQMAccess=true/false
```

値 **true** を使用してこの属性を指定し、アクセス・グループも指定していると、ローカル mqm グループはキュー・マネージャー・データ・ファイルへのアクセス権を付与されません。

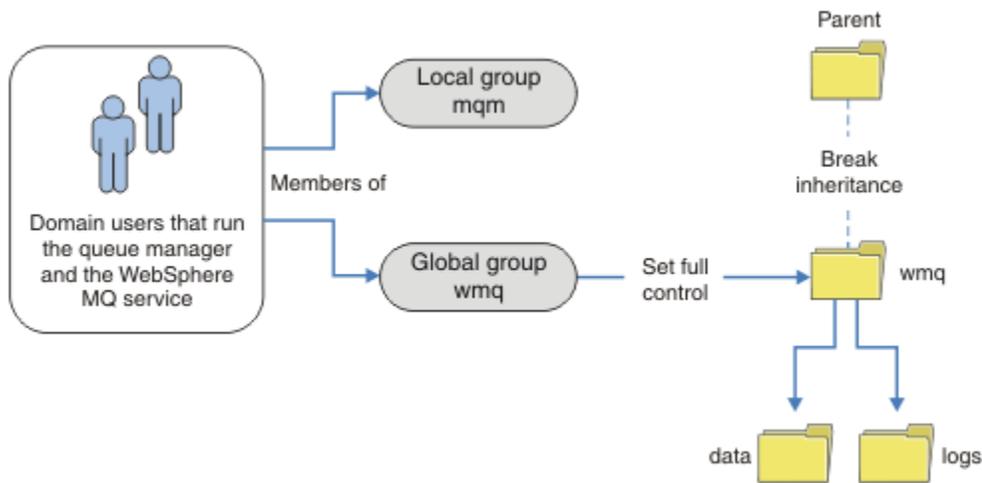


図 75. 代替グローバル・セキュリティ・グループを使用してキュー・マネージャーのデータとログを保護する操作 (1)

キュー・マネージャーのプロセスを実行するユーザー ID に、グローバル・セキュリティ・グループと合致する資格情報を設定するには、そのユーザー ID のスコープがグローバル・スコープになっていることも必要です。ローカル・グループまたはローカル・プリンシパルをグローバル・グループのメンバーにすることはできません。471 ページの図 75 では、キュー・マネージャーのプロセスを実行するユーザーをドメイン・ユーザーとしています。

多数の IBM MQ サーバーをデプロイする場合は、471 ページの図 75 にあるようにユーザーをグループ化する作業が面倒になります。ローカル・グループにユーザーを追加するプロセスを IBM MQ サーバーごとに繰り返さなければならないからです。代わりに、ドメイン・コントローラー上に Domain mqm グローバル・グループを作成し、Domain mqm グループの IBM MQ メンバーを実行するユーザーを作成します。472 ページの図 76 を参照してください。IBM MQ をドメイン・インストールとしてインストールすると、「IBM MQ の準備」ウィザードにより、Domain mqm グループが自動的にローカル mqm グループのメンバーになります。同じユーザーが、グローバル・グループ Domain mqm と wmq の両方に存在します。

**ヒント:** 同じユーザーが別々のサーバーで IBM MQ を実行することは可能ですが、1 つのサーバーで IBM MQ をサービスとして実行するユーザーと、対話式に実行するユーザーは、別々にしなければなりません。1 つのサーバーにあるインストール環境ごとに別々のユーザーを使用することも必要です。したがって、Domain mqm には通常、多数のユーザーが含まれます。

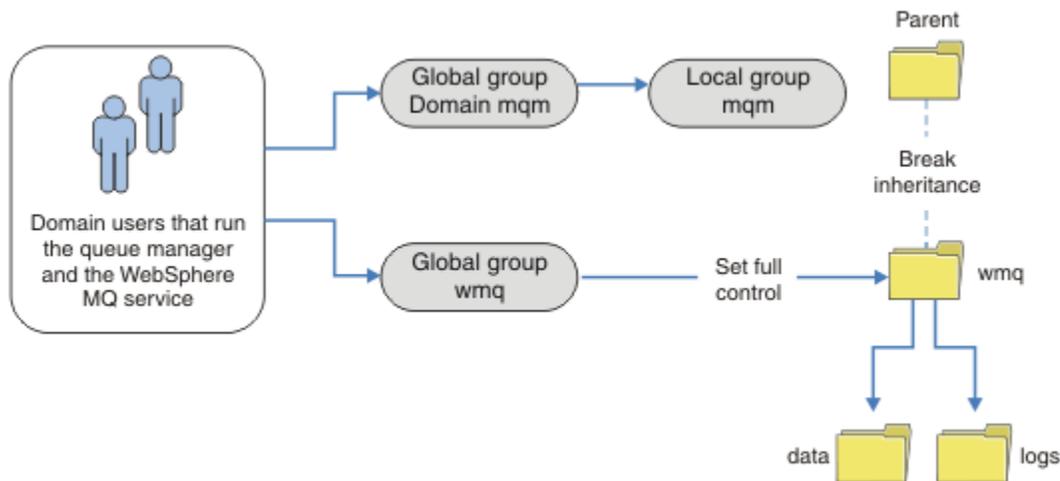


図 76. 代替グローバル・セキュリティ・グループを使用してキュー・マネージャーのデータとログを保護する操作 (2)

472 ページの図 76 の編成は、必要以上に複雑になっています。この配置では、2つのグローバル・グループに同じメンバーが入っています。この編成を簡略化して、グローバル・グループを1つだけ定義した環境が 472 ページの図 77 です。

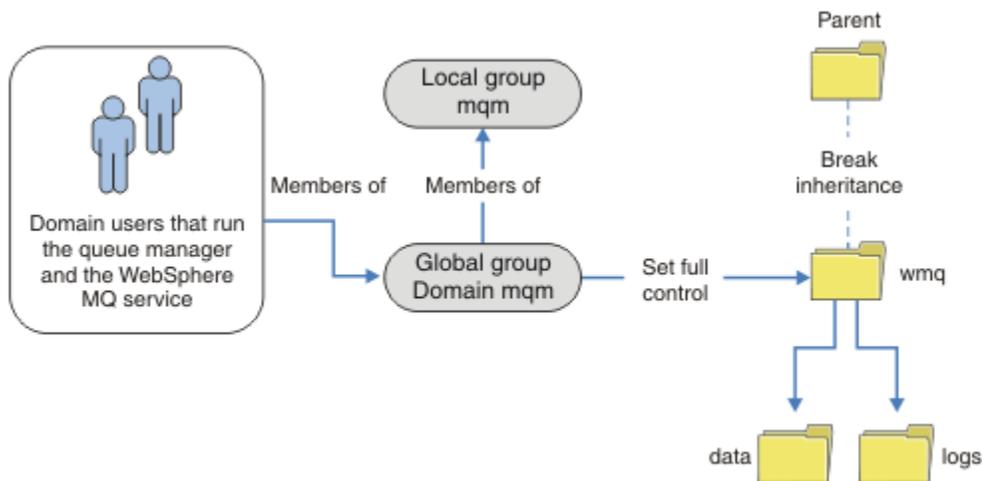


図 77. 代替グローバル・セキュリティ・グループを使用してキュー・マネージャーのデータとログを保護する操作 (3)

場合によっては、それぞれのキュー・マネージャーからアクセスできるフォルダーを制限することによって、アクセス制御をきめ細かく設定しなければならない環境もあります。473 ページの図 78 を参照してください。473 ページの図 78 では、ドメイン・ユーザーの2つのグループを定義し、別々のグローバル・グループで、別々のキュー・マネージャーのログ・ファイルとデータ・ファイルを保護しています。2つのローカル mqm グループがあり、それぞれを別々の IBM MQ サーバーに配置しなければなりません。この例では、各キュー・マネージャーを2つのセットに区別化し、その2つのセットに別々のユーザーを割り振ります。例えば、テスト用のキュー・マネージャーと実動用のキュー・マネージャーという2つのセットを設定する環境などが考えられます。それぞれの代替セキュリティ・グループの名前は、wmq1 と wmq2 です。テストの区分に所属するのか、実動の区分に所属するのかに応じて、グローバル・グループ wmq1 と wmq2 を正しいキュー・マネージャーに手動で追加しなければなりません。2つのユーザー・グル

ープがあるため、472 ページの図 77 の場合と同様に、IBM MQ のインストールによって Domain mqm がローカル mqm グループに伝搬されるということを構成で利用することはできません。

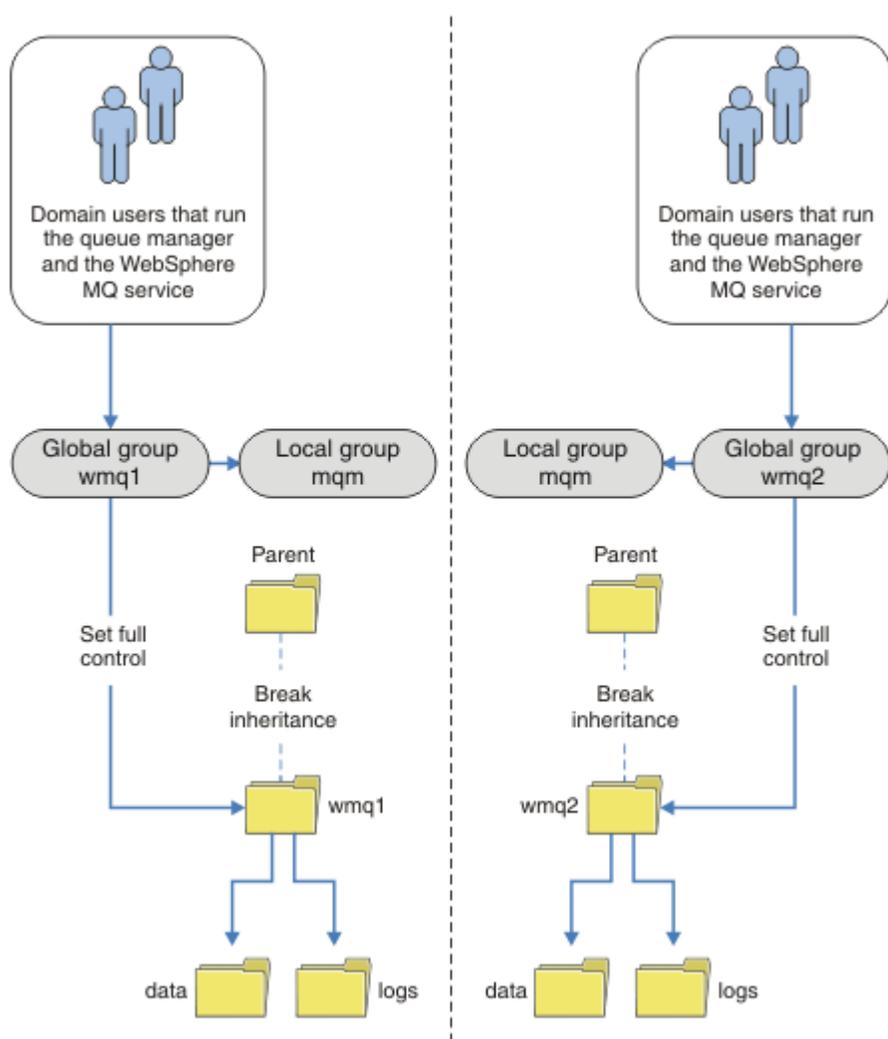


図 78. 代替グローバル・セキュリティ・プリンシパルを使用してキュー・マネージャーのデータとログを保護する操作 (4)

2 つに区分するためのもう 1 つの方法は、それぞれの区分を 2 つの Windows ドメインに配置することです。その場合は、472 ページの図 77 にあるようなシンプルモデルに戻ることができます。

#### **Windows** Windows でキュー・マネージャーのデータとログの非共有ディレクトリーおよびファイルを保護する操作

このトピックでは、ローカル mqm グループと代替セキュリティ・グループの両方を使用して、キュー・マネージャーのデータ・ファイルとログ・ファイルの代替格納場所を保護する方法を取り上げます。

通常、キュー・マネージャーのデータ・ファイルとログ・ファイルの代替格納場所をセットアップすることはありません。IBM MQ for Windows をインストールするときに、作成するキュー・マネージャーのために指定したホーム・ディレクトリーがインストール・プログラムによって作成されます。キュー・マネージャーのディレクトリーは、ローカル mqm グループによって保護され、それらのディレクトリーにアクセスする IBM MQ サービスのユーザー ID が構成されます。

IBM MQ のアクセス制御を構成する方法を 2 つの例で確認します。これらの例では、インストール時に作成されるデータとログのパスに存在しないディレクトリーにデータとログを格納するキュー・マネージャーを作成する方法を見ていきます。最初の例 (475 ページの『ローカル mqm グループによって許可されたデータ・ファイルとログ・ファイルの読み取りと書き込み』) では、ローカル mqm グループによって、キュー

ーとログのディレクトリーに対するアクセスを許可します。2番目の例(479ページの『代替ローカル・セキュリティ・グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りと書き込み』)では、代替セキュリティ・グループによって、それらのディレクトリーに対するアクセスを許可します。1つのサーバーだけで実行するキュー・マネージャーがそれらのディレクトリーにアクセスする場合は、代替セキュリティ・グループでデータ・ファイルとログ・ファイルを保護することによって、他のローカル・グループやローカル・プリンシパルで他のキュー・マネージャーを保護するという選択肢が可能になります。一方、複数のサーバーで実行するキュー・マネージャー(複数インスタンス・キュー・マネージャーなど)がそれらのディレクトリーにアクセスする場合は、代替セキュリティ・グループによってデータ・ファイルとログ・ファイルを保護することが唯一の選択肢になります。469ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護(Windows)』を参照してください。

Windowsの場合、キュー・マネージャーのデータ・ファイルとログ・ファイルに関するセキュリティ権限を構成する作業は、一般的な作業ではありません。IBM MQ for Windowsのインストール時に、キュー・マネージャーのデータとログのディレクトリーを指定するか、デフォルトのディレクトリーをそのまま受け入れます。インストール・プログラムは、これらのディレクトリーをローカルmqmグループで自動的に保護し、完全な制御権限を付与します。インストール・プロセスによって、キュー・マネージャーを実行するユーザーIDは、そのローカルmqmグループのメンバーになります。それぞれのアクセス要件に合わせて、それらのディレクトリーに対する他のアクセス権限を変更することもできます。

データ・ファイルとログ・ファイルのディレクトリーを新しい場所に移動する場合は、その新しい場所のセキュリティを構成する必要があります。それらのディレクトリーの場所を変更する可能性があるのは、キュー・マネージャーをバックアップして別のコンピューターにリストアする場合や、キュー・マネージャーを複数インスタンス・キュー・マネージャーに変更する場合などです。新しい場所に移したキュー・マネージャーのデータとログのディレクトリーを保護する方法には、2つの選択肢があります。つまり、アクセス権をローカルmqmグループに限定する方法と、任意のセキュリティ・グループに限定する方法のいずれかで、それらのディレクトリーを保護できます。

それらのディレクトリーを保護するための手順の数が少なくすむのは、ローカルmqmグループを使用する場合です。データとログのディレクトリーに対するアクセス許可を設定し、ローカルmqmグループにフル・コントロール権限を与えます。そのための標準的な方法は、既存のアクセス許可をコピーして、親フォルダーからの継承を解除することです。その後、他のプリンシパルのアクセス許可を削除するか、制限できます。

「IBM MQの準備」ウィザードによってセットアップされるサービスのユーザーIDとは異なるユーザーIDでキュー・マネージャーを実行する場合は、そのユーザーIDをローカルmqmグループのメンバーにしなければなりません。そのための手順については、475ページの『ローカルmqmグループによって許可されたデータ・ファイルとログ・ファイルの読み取りと書き込み』の作業を参照してください。

代替セキュリティ・グループを使用して、キュー・マネージャーのデータ・ファイルとログ・ファイルを保護することもできます。代替セキュリティ・グループを使用して、キュー・マネージャーのデータ・ファイルとログ・ファイルを保護するプロセスでは、いくつかの手順を実行しなければなりません。475ページの図79を参照してください。ローカル・グループwmqは、代替セキュリティ・グループの一例です。

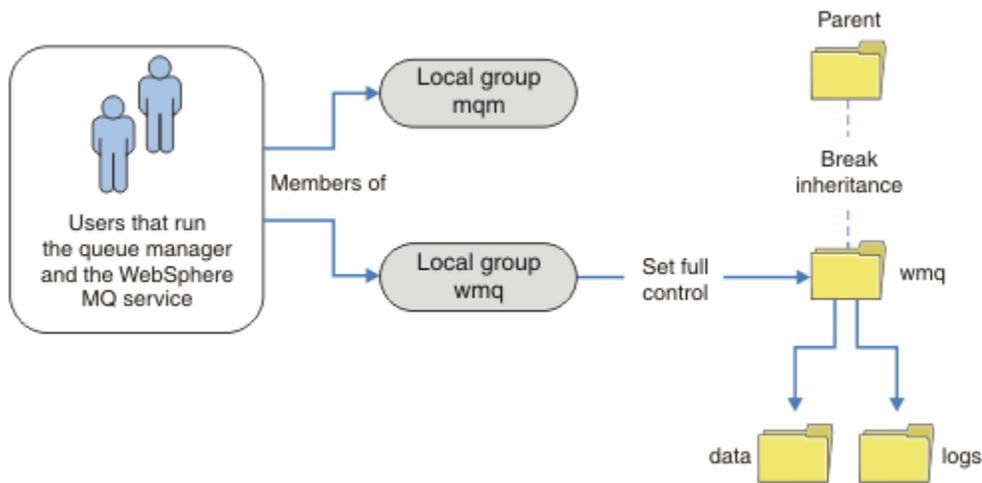


図 79. 代替ローカル・セキュリティ・グループ *wmq* を使用してキュー・マネージャーのデータとログを保護する操作

1. キュー・マネージャーのデータとログのために別々のディレクトリーを作成するか、共通のディレクトリーを作成するか、共通の親ディレクトリーを作成します。
2. 既存の継承アクセス許可をそれらのディレクトリーまたは親ディレクトリーにコピーして、それぞれの要件に合わせて変更します。
3. キュー・マネージャーのデータとログを格納するディレクトリーを保護するために、それらのディレクトリーに対するフル・コントロール権限を代替グループ *wmq* に与えます。
4. キュー・マネージャーのプロセスを実行するすべてのユーザー ID に、代替セキュリティ・グループまたは代替セキュリティ・プリンシパルの資格情報を与えます。
  - a. ユーザーを代替セキュリティ・プリンシパルとして定義する場合、そのユーザーは、キュー・マネージャーを実行するユーザーと同じユーザーでなければなりません。そのユーザーは、ローカル *mqm* グループのメンバーでなければなりません。
  - b. ローカル・グループを代替セキュリティ・グループとして定義する場合は、その代替グループに、キュー・マネージャーを実行するユーザーを追加しなければなりません。そのユーザーは、ローカル *mqm* グループのメンバーでもなければなりません。
  - c. グローバル・グループを代替セキュリティ・グループとして定義する場合は、[469 ページの『共有されるキュー・マネージャーのデータおよびログのディレクトリーとファイルの保護 \(Windows\)』](#)を参照してください。
5. `crtmqm` コマンドの `-a` パラメーターで代替セキュリティ・グループまたは代替セキュリティ・プリンシパルを指定して、キュー・マネージャーを作成します。

**Windows** ローカル *mqm* グループによって許可されたデータ・ファイルとログ・ファイルの読み取りと書き込み

この作業では、キュー・マネージャーを作成し、そのデータ・ファイルとログ・ファイルを任意のディレクトリーに格納する方法を確認します。それらのファイルに対するアクセスを保護するために、ローカル *mqm* グループを使用します。ディレクトリーは共有しません。

## 始める前に

1. IBM MQ for Windows をプライマリー・インストールとしてインストールします。
2. "IBM MQ" 準備ウィザードを実行します。この作業では、ローカル・ユーザー ID またはドメイン・ユーザー ID のどちらかでインストール環境を実行するように構成します。最終的に、[441 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』](#)のすべての作業を完了するには、ドメインの一部としてインストール環境を構成する必要があります。

3. この作業の最初の部分を実行するために、管理者権限でログオンします。

## このタスクについて

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの1つです。これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。さらに、[441 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』](#)の作業も必要です。

Windows では、IBM MQ for Windows のデフォルトのデータ・パスとログ・パスを任意のディレクトリーに作成できます。インストール構成ウィザードでは、ローカル mqm グループと、キュー・マネージャーのプロセスを実行するユーザー ID に、それらのディレクトリーに対するアクセス権が自動的に与えられます。キュー・マネージャーのデータ・ファイルとログ・ファイルのために別のディレクトリーを指定してキュー・マネージャーを作成する場合は、それらのディレクトリーに対するフル・コントロール権限を構成する必要があります。

この例では、ディレクトリー `c:\wmq` にローカル mqm グループ許可に与えることによって、データ・ファイルとログ・ファイルに対するフル・コントロール権限をキュー・マネージャーに与えます。

`crtmqm` コマンドを実行して、ワークステーションが IBM MQ サービスを使用し始める時点で自動的に始動するキュー・マネージャーを作成します。

この作業は1つの例にすぎません。具体的な値を使用していますが、変更することも可能です。変更できる値は、斜体で示しています。作業の最終段階で、すべての変更を削除する手順を実行してください。

## 手順

1. コマンド・プロンプトを開きます。
2. 次のコマンドを入力します。

```
md c:\wmq\data, c:\wmq\logs
```

3. 対象のディレクトリーに対する読み取り/書き込み権限をローカル mqm グループに設定します。

```
cacls c:\wmq/T /E /G mqm:F
```

システム応答:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

4. オプション: ローカル mqm グループのメンバーになっているユーザー ID に切り替えます。

この後も管理者として作業を続けてかまいませんが、実際の実動構成では、権限を限定したユーザー ID で作業を進めるようにしてください。そのユーザー ID は、少なくともローカル mqm グループのメンバーでなければなりません。

IBM MQ のインストール環境をドメインの一部として構成する場合は、そのユーザー ID を Domain mqm グループのメンバーにします。"準備 IBM MQ"ウィザードは、Domain mqm グローバル・グループをローカル mqm グループのメンバーにします。したがって、ユーザー ID をローカル mqm グループのメンバーに直接する必要はありません。

5. キュー・マネージャーを作成します。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

システム応答:

```
IBM MQ queue manager created.  
Directory 'c:\wmq\data\QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
Default objects statistics : 74 created. 0 replaced.  
Completing setup.  
Setup completed.
```

6. キュー・マネージャーによって、作成されたディレクトリーが `c:\wmq` ディレクトリー内にあることを確認します。

```
dir c:\wmq/D /B /S
```

7. 対象のファイルに対する読み取り/書き込み権限またはフル・コントロール権限がローカル `mqm` グループにあることを確認します。

```
cacls c:\wmq\*.*
```

## 次のタスク

キューにメッセージを書き込み、キューからメッセージを取り出して、キュー・マネージャーをテストします。

1. キュー・マネージャーを始動します。

```
strmqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using 7.1.0.0.
```

2. テスト・キューを作成します。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

システム応答:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: IBM MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. サンプル・プログラム **amqsput** を使用してテスト・メッセージを書き込みます。

```
echo 'A test message' | amqsput QTEST QMGR
```

システム応答:

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. サンプル・プログラム **amqsget** を使用してテスト・メッセージを取り出します。

```
amqsget QTEST QMGR
```

システム応答:

```
Sample AMQSGET0 start
message A test message
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. キュー・マネージャーを停止させます。

```
endmqm -i QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. キュー・マネージャーを削除してください。

```
dltmqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. 作成したディレクトリーを削除します。

**ヒント:** それぞれのファイルやディレクトリーの削除に関する確認プロンプトが表示されないようにするには、コマンドに /Q オプションを追加します。

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

## 関連概念

441 ページの『[Windows ドメインおよび複数インスタンス・キュー・マネージャー](#)』

Windows で複数インスタンス・キュー・マネージャーを実行する場合は、データとログを共有する必要があります。その共有フォルダーは、それぞれのサーバーやワークステーションで稼働するキュー・マネージャーのすべてのインスタンスからアクセスできる状態でなければなりません。キュー・マネージャーと共有フォルダーを1つの Windows ドメインの中で構成してください。キュー・マネージャーは、ドメイン・ワークステーションでも、ドメイン・サーバーでも、ドメイン・コントローラーでも実行できます。

## 関連タスク

**Windows** [代替ローカル・セキュリティ・グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りと書き込み](#)

この作業では、**crtmqm** コマンドで **-a** フラグを使用する方法を確認します。このフラグによって、ログ・ファイルとデータ・ファイルに対するアクセス権をキュー・マネージャーに与えるための代替ローカル・セキュリティ・グループを指定します。

456 ページの『[代替グローバル・セキュリティ・グループによって許可された共有データ・ファイルおよびログ・ファイルの読み取りと書き込み](#)』

443 ページの『[ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 \(Windows\)](#)』

**Windows** [代替ローカル・セキュリティ・グループによって許可されたデータ・ファイルおよびログ・ファイルの読み取りと書き込み](#)

この作業では、**crtmqm** コマンドで **-a** フラグを使用する方法を確認します。このフラグによって、ログ・ファイルとデータ・ファイルに対するアクセス権をキュー・マネージャーに与えるための代替ローカル・セキュリティ・グループを指定します。

## 始める前に

1. IBM MQ for Windows をプライマリー・インストールとしてインストールします。
2. "IBM MQ " 準備ウィザードを実行します。この作業では、ローカル・ユーザー ID またはドメイン・ユーザー ID のどちらかでインストール環境を実行するように構成します。最終的に、[441 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』](#)のすべての作業を完了するには、ドメインの一部としてインストール環境を構成する必要があります。
3. この作業の最初の部分を実行するために、管理者権限でログオンします。

## このタスクについて

このタスクは、キュー・マネージャーのデータおよびログ・ファイルへのアクセスについて説明して、関連タスクのセットの 1 つです。これらの作業によって、任意のディレクトリーに格納されているデータ・ファイルとログ・ファイルに対する読み取り/書き込み権限のあるキュー・マネージャーを作成する方法を確認します。さらに、[441 ページの『Windows ドメインおよび複数インスタンス・キュー・マネージャー』](#)の作業も必要です。

Windows では、IBM MQ for Windows のデフォルトのデータ・パスとログ・パスを任意のディレクトリーに作成できます。インストール構成ウィザードでは、ローカル **mqm** グループと、キュー・マネージャーのプロセスを実行するユーザー ID に、それらのディレクトリーに対するアクセス権が自動的に与えられます。キュー・マネージャーのデータ・ファイルとログ・ファイルのために別のディレクトリーを指定してキュー・マネージャーを作成する場合は、それらのディレクトリーに対するフル・コントロール権限を構成する必要があります。

この例では、キュー・マネージャーのために、対象のディレクトリーに対するフル・コントロール権限を持った代替セキュリティ・ローカル・グループを設定します。その代替セキュリティ・グループによって、対象のディレクトリーに含まれているファイルを管理する権限をキュー・マネージャーに与えます。代替セキュリティ・グループの主な目的は、代替セキュリティ・グローバル・グループに権限を与えることです。複数インスタンス・キュー・マネージャーをセットアップするときには、代替セキュリティ・グローバル・グループを使用します。この例では、代替セキュリティ・グループの使い方に習熟するためにローカル・グループを構成するだけにして、IBM MQ をドメインにインストールする作業は行いません。通常、代替セキュリティ・グループとしてローカル・グループを構成することはありません。

**crtmqm** コマンドを実行して、ワークステーションが IBM MQ サービスを使用し始める時点で自動的に始動するキュー・マネージャーを作成します。

この作業は 1 つの例にすぎません。具体的な値を使用していますが、変更することも可能です。変更できる値は、斜体で示しています。作業の最終段階で、すべての変更を削除する手順を実行してください。

## 手順

### 1. 代替セキュリティ・グループをセットアップします。

通常、代替セキュリティ・グループは、ドメイン・グループです。この例では、ローカル代替セキュリティ・グループを使用するキュー・マネージャーを作成します。ドメインの一部ではない IBM MQ のインストール環境に関する作業は、ローカル代替セキュリティ・グループでも実行できます。

- a) **lusrmgr.msc** コマンドを実行して、「ローカルユーザーとグループ」ウィンドウを開きます。
- b) 「グループ」 > 「新しいグループ...」を右クリックします。
- c) 「グループ名」フィールドに *altnmqm* と入力し、「作成」 > 「閉じる」をクリックします。
- d) IBM MQ サービスを実行するユーザー ID を識別します。
  - i) **開始 > 実行 ...** をクリックし、*services.msc* と入力し、「OK」をクリックします。
  - ii) サービスのリストから IBM MQ サービスをクリックし、「ログオン」タブをクリックします。
  - iii) ユーザー ID を記録し、「サービス」エクスプローラーを閉じます。
- e) IBM MQ サービスを実行するユーザー ID を *altnmqm* グループに追加します。さらに、キュー・マネージャーを作成して対話式に実行するためにログオン時に使用するユーザー ID も追加します。

Windows は、キュー・マネージャーのプロセスを実行するユーザー ID の権限を検査することによって、データとログのディレクトリーに対するアクセス権がキュー・マネージャーにあるかどうかを確認します。そのユーザー ID は、対象のディレクトリーに対する権限を持った *altnmqm* グループの直接的なメンバーであるか、グローバル・グループに基づく間接的なメンバーでなければなりません。

IBM MQ ドメインの一部としてインストールされ、443 ページの『ドメイン・ワークステーションまたはドメイン・サーバーでの複数インスタンス・キュー・マネージャーの作成 (Windows)』でタスクを実行する場合、446 ページの『Active Directory と DNS のドメインの作成 (Windows)』で作成されたドメインユーザー ID は、*wmquser1* と *wmquser2* で作成されたドメインユーザー ID です。

ドメインの一部としてキュー・マネージャーをインストールしたわけではない環境では、IBM MQ サービスを実行するデフォルトのローカル・ユーザー ID は、*MUSR\_MQADMIN* になります。管理者権限なしでそれらの作業を実行する場合は、ローカル *mqm* グループのメンバーであるユーザーを作成してください。

以下の手順を実行して、*wmquser1* と *wmquser2* を *altnmqm* に追加します。構成内容が異なる場合は、ここで取り上げるユーザー ID とグループの代わりに該当する名前を使用してください。

- i) グループのリストで、**altnmqm > プロパティ > 追加 ...** を右クリックします。
- ii) 「ユーザー、コンピューター、またはグループの選択」ウィンドウで *wmquser1 ; wmquser2* と入力し、「名前の確認」をクリックします。
- iii) 「Windows セキュリティ」ウィンドウにドメイン管理者の名前とパスワードを入力し、「OK」 > 「OK」 > 「適用」 > 「OK」をクリックします。

### 2. コマンド・プロンプトを開きます。

### 3. IBM MQ サービスを再開します。

サービスを実行するユーザー ID で、ここで構成した追加のセキュリティ資格情報を有効にするには、サービスを再始動する必要があります。

次のコマンドを入力します。

```
endmqsvc  
strmqsvc
```

システム応答:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.
```

および:

5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' started successfully.

4. 次のコマンドを入力します。

```
md c:\wmq\data, c:\wmq\logs
```

5. 対象のディレクトリーに対する読み取り/書き込み権限をローカル・ユーザー *user* に設定します。

```
cacls c:\wmq/T /E /G altmqm:F
```

システム応答:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

6. オプション: ローカル mqm グループのメンバーになっているユーザー ID に切り替えます。

この後も管理者として作業を続けてかまいませんが、実際の実動構成では、権限を限定したユーザー ID で作業を進めるようにしてください。そのユーザー ID は、少なくともローカル mqm グループのメンバーでなければなりません。

IBM MQ のインストール環境をドメインの一部として構成する場合は、そのユーザー ID を Domain mqm グループのメンバーにします。"準備 IBM MQ"ウィザードは、Domain mqm グローバル・グループをローカル mqm グループのメンバーにします。したがって、ユーザー ID をローカル mqm グループのメンバーに直接する必要はありません。

7. キュー・マネージャーを作成します。

```
crtmqm -a altmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

システム応答:

```
IBM MQ queue manager created.
Directory 'c:\wmq1\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

8. キュー・マネージャーによって、作成されたディレクトリーが *c:\wmq* ディレクトリー内にあることを確認します。

```
dir c:\wmq/D /B /S
```

9. 対象のファイルに対する読み取り/書き込み権限またはフル・コントロール権限がローカル mqm グループにあることを確認します。

```
cacls c:\wmq\*.*
```

## 次のタスク

キューにメッセージを書き込み、キューからメッセージを取り出して、キュー・マネージャーをテストします。

1. キュー・マネージャーを始動します。

```
stimqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
IBM MQ queue manager 'QMGR' started using 7.1.0.0.
```

2. テスト・キューを作成します。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

システム応答:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2023. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: IBM MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. サンプル・プログラム **amqsput** を使用してテスト・メッセージを書き込みます。

```
echo 'A test message' | amqsput QTEST QMGR
```

システム応答:

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. サンプル・プログラム **amqsget** を使用してテスト・メッセージを取り出します。

```
amqsget QTEST QMGR
```

システム応答:

```
Sample AMQSGET0 start  
message A test message  
Wait 15 seconds ...  
no more messages
```

Sample AMQSGETO end

5. キュー・マネージャーを停止させます。

```
endmqm -i QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' ending.
```

```
IBM MQ queue manager 'QMGR' ended.
```

6. キュー・マネージャーを削除してください。

```
dltmqm QMGR
```

システム応答:

```
IBM MQ queue manager 'QMGR' deleted.
```

7. 作成したディレクトリーを削除します。

**ヒント:**それぞれのファイルやディレクトリーの削除に関する確認プロンプトが表示されないようにするには、コマンドに /Q オプションを追加します。

```
del /F /S C:\wmq\*. *  
rmdir /S C:\wmq
```

## 関連タスク

**Windows** ローカル mqm グループによって許可されたデータ・ファイルとログ・ファイルの読み取りと書き込み

この作業では、キュー・マネージャーを作成し、そのデータ・ファイルとログ・ファイルを任意のディレクトリーに格納する方法を確認します。それらのファイルに対するアクセスを保護するために、ローカル mqm グループを使用します。ディレクトリーは共有しません。

**Linux** Linuxでの複数インスタンス・キュー・マネージャーの作成

例では、Linux 上で複数インスタンス・キュー・マネージャーをセットアップする方法を示しています。セットアップは、関係した概念を例示するために小規模なものとしています。この例は、Linux Red HatEnterprise5 に基づいています。このステップは、他の UNIX プラットフォームで異なります。

## このタスクについて

この例は、3GBRAM が稼働している 2GHz のノートブック・コンピューター(Windows7ServicePack1)にセットアップされています。二つの VMware 仮想マシン(Server1 および Server2)が 640 MB のイメージで Linux Red HatEnterprise5 を実行します。サーバー 1 は、ネットワーク・ファイル・システム (NFS)、キュー・マネージャー・ログおよび HA インスタンスをホストします。通常の手法では、NFS サーバーがキュー・マネージャー・インスタンスのいずれかもホストすることはありません。これは、例を単純化するためのものです。サーバー 2 は、サーバー 1 のキュー・マネージャー・ログをスタンバイ・インスタンスでマウントします。WebSphere MQ MQI クライアントは、Windows 7 Service Pack 1 を実行する、400 MB の追加 VMware イメージ上にインストールされており、サンプルの高可用性アプリケーションを実行します。すべての仮想マシンは、セキュリティ上の理由から、VMware ホスト専用ネットワークの一部として構成されています。

**注:** NFS サーバーにはキュー・マネージャー・データのみを書き込む必要があります。NFS では、mount コマンドで次の 3 つのオプションを使用して、システムを保護します。

- **noexec**

このオプションを使用すると、バイナリー・ファイルを NFS 上で実行できなくなります。こうすることで、リモート・ユーザーがシステム上で望ましくないコードを実行できないようにします。

- **nosuid**

このオプションを使用すると、セット・ユーザー ID ビットとセット・グループ ID ビットを使用できなくなります。こうすることで、リモート・ユーザーが上位の特権を取得できないようにします。

- **nodev**

このオプションを使用すると、文字およびブロック特殊装置が使用または定義できなくなります。こうすることで、リモート・ユーザーが chroot ジェイルから出られないようにします。

## 手順

1. root としてログインします。
2. [IBM MQ のインストール - 概要](#)を参照し、適切なリンクに従って IBM MQ をインストールし、mqm ユーザーおよびグループを作成し、/var/mqm を定義します。
3. [共有ファイル・システムの動作の検証](#)の作業を実行し、ファイル・システムが複数インスタンス・キュー・マネージャーをサポートしていることを確認します。
4. Server1 で、以下の手順を実行します。
  - a. 共有される共通フォルダー /MQHA に、ログおよびデータ用のディレクトリーを作成します。以下に例を示します。
    - i) **mkdir /MQHA**
    - ii) **mkdir /MQHA/logs**
    - iii) **mkdir /MQHA/qmgrs**
5. Server2 で、以下の手順を実行します。
  - a. ファイル共有システムをマウントするフォルダー /MQHA を作成します。このパスをサーバー 1 と同じにしておきます。以下に例を示します。
    - i) **mkdir /MQHA**
6. MQHA ディレクトリーがユーザーおよびグループ mqm により所有されており、ユーザーおよびグループのアクセス権が **rw** に設定されていることを確認します。例えば、**ls -al** は **drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA** を表示します。
  - a. **chown -R mqm:mqm /MQHA**
  - b. **chmod -R ug+rw /MQHA**
7. コマンド **crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM1** を入力して、キュー・マネージャーを作成します。
8. 追加 <sup>2</sup>/MQHA \*(rw, sync, no\_wdelay, fsid=0) を /etc/exports に追加します。
9. Server1 で、以下の手順を実行します。
  - a. NFS デーモンを開始します。/etc/init.d/ **nfs start**
  - b. 以下のように、キュー・マネージャー構成の詳細をサーバー 1 からコピーします。

```
dspmqlinf -o command QM1
```

そして、結果をクリップボードにコピーします。

```
addmqinf -s QueueManager  
-v Name=QM1  
-v Directory=QM1
```

<sup>2</sup> '\*' では、読み取り/書き込み用に一つのマウント/MQHA に到達できるすべてのマシンを使用できます。実動マシンでのアクセスを制限します。

```
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

10. Server2 で、以下の手順を実行します。

- a. コマンド **mount -t nfs4 -o hard,intr Server1:/ /MQHA** を入力して、エクスポートされたファイル・システム /MQHA をマウントします。
- b. 以下のように、キュー・マネージャー構成コマンドをサーバー 2 に貼り付けます。

```
addmqinf -s QueueManager
-v Name=QM1
-v Directory=QM1
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

11. **-x** パラメーターを使用して、キュー・マネージャー・インスタンスをいずれかの順序で開始します (**strmqm -x QM1**)。

キュー・マネージャー・インスタンスの開始に使用されるコマンドは、**addmqinf** コマンドと同じ IBM MQ のインストール済み環境から発行される必要があります。異なるインストール済み環境からキュー・マネージャーを開始および停止するには、まず **setmqm** コマンドを使用してキュー・マネージャーと関連するインストール済み環境を設定する必要があります。詳細については、[setmqm](#) を参照してください。

### Linux Linux での複数インスタンス・キュー・マネージャーの検査

複数インスタンス・キュー・マネージャー構成を検査するには、サンプル・プログラム **amqsghac**、**amqsphac**、および **amqsmhac** を使用します。このトピックでは、Linux Red Hat Enterprise 5 上で複数インスタンス・キュー・マネージャーの構成を検査するためのサンプル構成を記載しています。

高可用性サンプル・プログラムは、クライアントの自動再接続を使用します。接続されたキュー・マネージャーに障害が起きた場合、クライアントは同じキュー・マネージャー・グループ内のキュー・マネージャーへの再接続を試行します。サンプルの説明、高可用性のサンプル・プログラムでは、単純にするために単一インスタンス・キュー・マネージャーを使用してクライアントの再接続を説明しています。複数インスタンス・キュー・マネージャーがある同じサンプルを使用して、複数インスタンス・キュー・マネージャー構成を検査することができます。

サンプルでは、483 ページの『Linux での複数インスタンス・キュー・マネージャーの作成』で説明されている複数インスタンス構成を使用しています。この構成を使用して、複数インスタンス・キュー・マネージャーがスタンバイ・インスタンスに切り替わることを検査します。**endmqm** コマンドを使用してキュー・マネージャーを停止し、**-s** 切り替えオプションを使用します。クライアント・プログラムは、新規キュー・マネージャー・インスタンスに再接続して、少しの遅延の後に新規インスタンスの処理を続行します。

この例では、クライアントは Windows 7 Service Pack 1 システム上で実行されます。システムは、複数インスタンス・キュー・マネージャーを実行する 2 つの VMware Linux サーバーをホスティングします。

### IBM MQ Explorer を使用したフェイルオーバーの検査

フェイルオーバーを検査するためにサンプル・アプリケーションを使用する前に、IBM MQ Explorer を各サーバー上で実行します。「リモート・キュー・マネージャーの追加」>「複数インスタンス・キュー・マネージャーへの直接接続」ウィザードを使用して、両方のキュー・マネージャー・インスタンスを各エクスプローラーに追加します。両方のインスタンスをスタンバイを許可して実行しておきます。アクティブ・インスタンスがある VMware イメージを実行中のウィンドウを閉じて仮想的にサーバーの電源をオフにするか、またはスタンバイ・インスタンスへの切り替えを許可してアクティブ・インスタンスを停止します。

注：サーバーの電源をオフにする場合は、これが /MQHA をホスティングしているものではないことを確認してください。

注：「キュー・マネージャーの停止」ダイアログで「スタンバイ・インスタンスへの切り替えを許可」オプションが表示されない場合があります。このオプションがないのは、キュー・マネージャーが単一インスタンス・キュー・マネージャーとして実行されているためです。それを開始したときに、「スタンバイ・インスタンスの許可」オプションを指定しませんでした。キュー・マネージャーの停止要求が拒否された場

合は、「詳細」ウィンドウを参照してください。実行中のスタンバイ・インスタンスがないことが原因である可能性があります。

## サンプル・プログラムを使用したフェイルオーバーの検査

### アクティブ・インスタンスを実行するサーバーの選択

MQHA ディレクトリーまたはファイル・システムのホストになるサーバーが1つ選択されている場合があります。アクティブ・サーバーを実行中のVMware ウィンドウを閉じることによってフェイルオーバーをテストする予定であれば、それがMQHAのホストになるサーバーではないことを確認してください。

### アクティブ・キュー・マネージャー・インスタンスを実行するサーバー上

注：SVRCONN チャンネルを、MCAUSER をmqm に設定して実行することは、サンプルの構成ステップの数を減らすことに役立ちます。別のユーザー ID を選択し、システムがサンプルで使用されているものは異なるセットアップとなっている場合、アクセス権問題が発生する可能性があります。mqm を公開システム上で MCAUSER として使用しないでください。これはセキュリティが大きく危険に晒される可能性があります。

1. *ipaddr1* および *ipaddr2* を変更し、以下のコマンドを/MQHA/hasamples.tst. に保管する

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER('mqm') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME(' ipaddr1 (1414), ipaddr2
(1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
START LISTENER(LISTENER.TCP)
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

2. 端末ウィンドウをパス /MQHA で開き、以下のコマンドを実行します。

```
runmqsc -m QM1 < hasamples.tst
```

3. **runmqsc** コマンドの出力を検査することで、リスナーが実行中であり、キュー・マネージャー制御があることを確認します。

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

または、IBM MQ Explorer を使用して、TCPIP リスナーが実行中で、Control = Queue Manager があることを確認します。

### クライアント上

1. クライアント接続テーブル AMQCLCHL.TAB を、サーバー上の/MQHA/qmgrs/QM1.000/@ipcc からクライアント上の C:\ にコピーします。
2. パス C:\ を指定してコマンド・プロンプトを開き、クライアント・チャンネル定義テーブル (CCDT) を指すように環境変数 MQCHLLIB を設定します。

```
SET MQCHLLIB=C:\
```

3. コマンド・プロンプトで、以下のコマンドを入力します。

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

## アクティブ・キュー・マネージャー・インスタンスを実行するサーバー上

### 1. 次のいずれかの場合:

- アクティブ・サーバー・インスタンスがある VMware イメージを実行するウィンドウを閉じます。
- IBM MQ Explorer を使用して、アクティブ・キュー・マネージャー・インスタンスを停止して、スタンバイ・インスタンスへの切り替えを許可し、再接続可能クライアントに再接続を指示します。

### 2. 最終的には 3 つのクライアントで接続が切断していることが検出され、次いで再接続されます。この構成では、サーバー・ウィンドウを閉じた場合、3 つの接続がすべて再確立されるには約 7 分かかります。他の接続よりも速く再確立される接続もあります。

## 結果

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start

17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsghac TARGET QM1
Sample AMQSGHAC start
message Message 1
message Message 2
message Message 3
message Message 4
message Message 5
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message Message 6
message Message 7
message Message 8
message Message 9
```

## Multi 複数インスタンス・キュー・マネージャーの削除

Multiplatforms で複数インスタンス・キュー・マネージャーを完全に削除するには、**dltmqm** コマンドを使用してキュー・マネージャーを削除してから、**rmvmqinf** コマンドか **dltmqm** コマンドを使用して他のサーバーからインスタンスを削除します。

**dltmqm** コマンドを実行して、他のサーバー (キュー・マネージャーが定義されているすべてのサーバー) 上で定義されているインスタンスがあるキュー・マネージャーを削除します。**dltmqm** コマンドは、それを作成した同じサーバー上で実行する必要はありません。次いで、**rmvmqinf** または **dltmqm** コマンドを、キュー・マネージャーの定義がある他のすべてのサーバー上で実行します。

キュー・マネージャーは停止しているときにのみ削除できます。削除時には実行中のインスタンスはなく、厳密に言えばキュー・マネージャーは単一または複数のどちらのインスタンス・キュー・マネージャーでもありません。それはリモート・シェアにキュー・マネージャー・データとログを持つ単なるキュー・マネージャーです。キュー・マネージャーの削除時には、そのキュー・マネージャー・データとログは削除され、キュー・マネージャー・スタンプは **dltmqm** コマンドを発行したサーバー上の **mqm.ini** ファイルから除去されます。キュー・マネージャーを削除するときには、キュー・マネージャー・データとログを含むネットワーク・シェアへのアクセス権限が必要です。

以前にキュー・マネージャーのインスタンスを作成した他のサーバー上の **mqm.ini** ファイル内にも項目があります。各サーバーに順にアクセスし、**rmvmqinf Queue manager stanza name** コマンドを実行して、キュー・マネージャー・スタンプを除去する必要があります。

**Linux** **UNIX** UNIX and Linux システム上では、**mqm.ini** 共通ファイルをネットワーク・ストレージ内に置いていて、各サーバー上で **AMQ\_MQS\_INI\_LOCATION** 環境変数を設定してすべてのサーバーからそれを参照していた場合は、更新するファイルは **mqm.ini** ファイル 1 つしかないので、そのサーバーの 1 つだけからキュー・マネージャーを削除すれば済みます。

## 例

最初のサーバー

```
dltmqm QM1
```

インスタンスが定義されている他のサーバー

```
rmvmqinf QM1、または
```

```
dltmqm QM1
```

## **Multi** 複数インスタンス・キュー・マネージャーの開始および停止

**Multiplatforms** で単一インスタンスまたは複数インスタンスのキュー・マネージャーとして構成されたキュー・マネージャーの開始と停止。

1 組みのサーバー上に複数インスタンス・キュー・マネージャーを定義した場合は、いずれかのサーバー上で、キュー・マネージャーを単一インスタンス・キュー・マネージャー、または複数インスタンス・キュー・マネージャーのいずれかとして実行することができます。

複数インスタンス・キュー・マネージャーを実行するには、**strmqm -x QM1** コマンドを使用して、一方のサーバー上でキュー・マネージャーを開始します。-x は、インスタンスのフェイルオーバーを許可するオプションです。これがアクティブ・インスタンスになります。同じ **strmqm -x QM1** コマンドを使用して、もう一方のサーバー上でスタンバイ・インスタンスを開始します。-x は、インスタンスのスタンバイとしての開始を許可するオプションです。

キュー・マネージャーは現在、すべての要求を処理している 1 つのアクティブ・インスタンスと、アクティブ・インスタンスに障害が起きた場合の引き継ぎに備えている 1 つのスタンバイ・インスタンスがある状態で実行されています。アクティブ・インスタンスには、キュー・マネージャーのデータおよびログに対する排他的アクセス権限が付与されています。スタンバイ・インスタンスは、キュー・マネージャーのデータおよびログに対する排他的アクセス権限が付与されるのを待っています。排他的アクセス権限が付与されたスタンバイ・インスタンスは、アクティブ・インスタンスになります。

アクティブ・インスタンス上で **endmqm -s** コマンドを発行することにより、手動でスタンバイ・インスタンスに制御を切り替えることもできます。**endmqm -s** コマンドは、スタンバイ・インスタンスをシャットダウンすることなく、アクティブ・インスタンスをシャットダウンします。キュー・マネージャーのデータおよびログに対する排他的アクセス・ロックが解除され、スタンバイ・インスタンスに引き継がれます。

異なるサーバー上に複数のインスタンスを置いて構成したキュー・マネージャーを、単一インスタンス・キュー・マネージャーとして開始および停止することもできます。**strmqm** コマンドで -x オプションを使用せずにキュー・マネージャーを開始した場合、他のマシン上に構成されたキュー・マネージャーのインスタンスをスタンバイ・インスタンスとして開始できなくなります。別のインスタンスを開始しようとすると、そのキュー・マネージャー・インスタンスをスタンバイ・インスタンスとして実行することは認められていないという応答を受け取ります。

**endmqm** コマンドで **-s** オプションを使用せずに複数インスタンス・キュー・マネージャーのアクティブ・インスタンスを停止すると、アクティブ・インスタンスとスタンバイ・インスタンスの両方が停止します。**-x** オプションを指定した **endmqm** コマンドを実行してスタンバイ・インスタンスを停止すると、そのインスタンスのスタンバイ状態は解除されますが、アクティブ・インスタンスの実行は続行します。**-x** オプションを使用していない **endmqm** をスタンバイ・インスタンスに対して発行することはできません。

同時に実行できるのは、アクティブ・インスタンスとスタンバイ・インスタンスの、2つのキュー・マネージャー・インスタンスだけです。2つのインスタンスを同時に開始した場合、IBM MQ は、どちらがアクティブ・インスタンスになるかを制御しません。どちらがアクティブ・インスタンスになるかはネットワーク・ファイル・システムにより決定されます。キュー・マネージャーのデータに対する排他的アクセス権限を最初に獲得したインスタンスが、アクティブ・インスタンスになります。

**注:** 失敗したキュー・マネージャーを再始動する前に、アプリケーションをキュー・マネージャーのそのインスタンスから切断する必要があります。そうしない場合、キュー・マネージャーは正しく再始動されない場合があります。

## Multi 共有ファイル・システム

Multipatforms の場合、複数インスタンス・キュー・マネージャーは、ネットワーク・ファイル・システムを使用してキュー・マネージャーのインスタンスを管理します。

複数インスタンス・キュー・マネージャーは、ファイル・システム・ロック、およびキュー・マネージャーのデータおよびログの共有を組み合わせて使用して、フェイルオーバーを自動化します。キュー・マネージャーのインスタンスのうち、キュー・マネージャーの共有データおよびログに対する排他的アクセス権限を持つことができるのは1つのみです。アクセス権限を取得したインスタンスが、アクティブ・インスタンスになります。排他的アクセス権限を取得できなかったもう一方のインスタンスは、キュー・マネージャーのデータおよびログが使用可能になるまで、スタンバイ・インスタンスとして待機します。

ネットワーク・ファイル・システムは、キュー・マネージャーのアクティブ・インスタンス用に保持しているロックの解除を担当します。アクティブ・インスタンスに何らかの障害が起こった場合、ネットワーク・ファイル・システムはアクティブ・インスタンス用に保持しているロックを解除します。排他ロックが解除されるとすぐに、ロックを待機していたスタンバイ・キュー・マネージャーがロックを獲得しようとします。獲得できた場合は、それがアクティブ・インスタンスになり、ファイル共有システム上のキュー・マネージャーのデータおよびログに対する排他的アクセス権限を保持します。その後、このインスタンスは開始段階に進みます。

関連トピックの ファイル・システム・サポートの計画 では、セットアップ方法、およびファイル・システムが複数インスタンス・キュー・マネージャーをサポートするかどうかを確認する方法を説明します。

複数インスタンス・キュー・マネージャーは、ファイル・システムを障害から保護するわけではありません。データを保護する方法はいくつかあります。

- 新磁気ディスク制御機構 (RAID) などの高信頼ストレージに投資して、それらをネットワーク回復力のあ  
るネットワーク・ファイル・システム内に組み込む。
- IBM MQ のリニア・ログを代替メディアにバックアップして、1次ログ・メディアに障害が起きた場合  
に、その代替メディア上のログを使用してリカバリーする。このプロセスは、バックアップ・キュー・  
マネージャーを使用して管理できます。

## Multi 複数のキュー・マネージャー・インスタンス

複数インスタンス・キュー・マネージャーは、障害発生後にスタンバイ・キュー・マネージャー・イン  
スタンスを使用してキュー・マネージャーの可用性を取り戻すことができるため、回復力があります。

キュー・マネージャー・インスタンスの複製は、キュー・マネージャー・プロセスの可用性を向上させる  
上で非常に有効な方法です。単に説明を目的とした単純な可用性モデルの場合で考えると、キュー・マ  
ネージャーのインスタンスが1つである場合の信頼性が99% (年間の累積ダウン時間が3.65日) である  
とき、キュー・マネージャーのインスタンスをもう1つ追加することで、可用性が99.99% (年間の累積ダ  
ウン時間が約1時間) に上昇します。

このモデルは、実際の可用性の数値見積もりを得るには単純過ぎます。可用性を現実的にモデル化する  
には、平均故障間隔 (MTBF) および平均修理時間 (MTTR) の統計、および故障間隔と修理時間の確率分布を  
収集する必要があります。

複数インスタンス・キュー・マネージャーという用語は、キュー・マネージャーのデータおよびログを共有する、キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの組み合わせを意味しています。複数インスタンス・キュー・マネージャーでは、一方のサーバーにあるキュー・マネージャー・インスタンスはアクティブにし、もう一方のサーバーにあるもう一方キュー・マネージャー・インスタンスはスタンバイにすることで、アクティブ・インスタンスで障害が起きた場合に自動的に引き継ぎが行われるようにして、キュー・マネージャー・プロセスを障害から保護します。

## Multi フェイルオーバーまたは切り替え

スタンバイ・キュー・マネージャー・インスタンスは、要求(切り替え)時またはアクティブ・インスタンスで障害が起きたとき(フェイルオーバー)に、アクティブ・インスタンスの処理を引き継ぎます。

- 切り替えは、アクティブ・キュー・マネージャー・インスタンスに対して発行された **endmqm -s** コマンドに回答して、スタンバイ・インスタンスが開始されたときに実行されます。**endmqm** のパラメーター **-c**、**-i**、または **-p** を指定することで、キュー・マネージャーをどれほど急速に停止するかを制御できます。

**注:** 切り替えは、スタンバイ・キュー・マネージャー・インスタンスが既に開始されている場合にのみ実行されます。**endmqm -s** コマンドは、アクティブ・キュー・マネージャーのロックを解除し、切り替えを許可します。ただしスタンバイ・キュー・マネージャー・インスタンスの開始は行いません。

- フェイルオーバーは、アクティブ・インスタンスが突然停止した可能性があるために、そのインスタンスがキュー・マネージャー・データに対して保持していたロックが解除されたときに(つまり、**endmqm** コマンドの発行なしに)、実行されます。

スタンバイ・インスタンスは、アクティブ・インスタンスとして処理を引き継ぐと、キュー・マネージャーのエラー・ログにメッセージを書き込みます。

再接続可能クライアントは、キュー・マネージャーの障害発生時または切り替え時に自動的に再接続します。クライアント再接続を要求するために **endmqm** コマンドに **-e** フラグを含める必要はありません。

IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

フェイルオーバーが発生し、スタンバイ・インスタンスがアクティブになったにもかかわらず、失敗したインスタンスを再開できない場合は、失敗したインスタンスにローカルで接続しているアプリケーションがその失敗したインスタンスから切断されていないかどうかを確認してください。

失敗したインスタンスを再開できるようにするには、ローカルで接続しているアプリケーションを終了するか、失敗したキュー・マネージャー・インスタンスから切断する必要があります。共有バインディングを使用しており(デフォルトの設定です)、失敗したインスタンスへの接続を保持しているローカル接続のアプリケーションが存在すると、そのインスタンスを再開する妨げになります。

ローカル接続のアプリケーションを終了できない場合、あるいはローカル・キュー・マネージャーの障害発生時にそれらのアプリケーションが確実に切断されるようにできない場合は、隔離されたバインディングの使用を検討してください。隔離されたバインディングを使用するローカル接続のアプリケーションは、切断されなくても、ローカル・キュー・マネージャー・インスタンスの再開を妨げません。

## Multi チャネルおよびクライアントの再接続

チャネルおよびクライアントの再接続は、スタンバイ・キュー・マネージャー・インスタンスがアクティブになった後、メッセージ処理を回復させるためには不可欠の作業です。

複数インスタンス・キュー・マネージャー・インスタンスは、異なるネットワーク・アドレスを持つ複数のサーバー上にインストールされています。すべてのキュー・マネージャー・インスタンスの接続情報を使用して IBM MQ のチャネルおよびクライアントを構成する必要があります。スタンバイが引き継ぎを行うと、クライアントおよびチャネルは、新規のネットワーク・アドレスにある新たにアクティブになったキュー・マネージャー・インスタンスに自動的に再接続します。IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

このような設計は、HA-CMP 処理などの高可用性環境の方法とは異なります。HA-CMP は、クラスターの仮想 IP アドレスを指定して、そのアドレスをアクティブ・サーバーに転送します。IBM MQ の再接続では、IP アドレスを変更または転送しません。これはチャネル定義およびクライアント接続で定義したネットワーク・アドレスを使用して再接続することにより機能します。管理者は、複数インスタンス・キュー・マネージャーのすべてのインスタンスに対して、チャネル定義およびクライアント接続でネットワーク・

アドレスを定義する必要があります。複数インスタンス・キュー・マネージャーに対してネットワーク・アドレスを構成するための最善の方法は、以下の接続に応じて異なります。

#### キュー・マネージャー・チャンネル

チャンネルの CONNAME 属性は、接続名のコンマ区切りリストです (例: CONNAME ('127.0.0.1(1234), 192.0.2.0(4321)')). 接続は、正常に確立されるまで、この接続リストに指定された順序に従って試行されます。正常に確立された接続がない限り、チャンネルは再接続を試みます。

#### クラスター・チャンネル

一般に、複数インスタンス・キュー・マネージャーがクラスター内で機能するために必要な追加の構成はありません。

キュー・マネージャーがリポジトリ・キュー・マネージャーに接続する場合は、リポジトリがキュー・マネージャーのネットワーク・アドレスを検出します。リポジトリは、キュー・マネージャーの CLUSRCVR チャンネルの CONNAME を参照します。TCP/IP では、CONNAME が省略されたりブランクに構成されたりしている場合は、キュー・マネージャーがそれを自動的に設定します。スタンバイ・インスタンスが引き継ぎを行う場合、その IP アドレスが CONNAME として以前のアクティブ・インスタンスの IP アドレスと置き換わります。

必要な場合、キュー・マネージャー・インスタンスのネットワーク・アドレスのリストを使用して、CONNAME を手動で構成することができます。

#### クライアント接続

クライアント接続では、接続リストを使用するか、キュー・マネージャー・グループを使用して代替接続を選択することができます。クライアントは IBM WebSphere MQ 7.0.1 以上のクライアント・ライブラリーで実行するようにコンパイルされる必要があります。また、それらを IBM WebSphere MQ 7.0.1 以上のキュー・マネージャーと接続する必要があります。

フェイルオーバー実行時の再接続には、ある程度の時間がかかります。スタンバイ・キュー・マネージャーは、始動を完了する必要があります。障害が起きたキュー・マネージャーに接続されていたクライアントは、接続障害を検出し、新規クライアント接続を開始する必要があります。新規クライアント接続が、新たにアクティブになったスタンバイ・キュー・マネージャーを選択する場合は、クライアントは同じキュー・マネージャーに再接続されます。

再接続時にクライアントが MQI 呼び出しの途中である場合は、その呼び出しが完了するまで待つ必要があります。

メッセージ・チャンネルでバッチ転送中に障害が発生した場合、そのバッチはロールバック後、リスタートされます。

切り替えはフェイルオーバーよりも短時間であり、1つのキュー・マネージャー・インスタンスを停止して、別の1つを開始するだけの時間で済みます。再生するログ・レコード数がわずかしくないキュー・マネージャーの場合、切り替えにかかる時間は多くても数秒程度です。フェイルオーバーにかかる時間を見積もるには、障害が検出されるまでの時間を加算する必要があります。検出にかかる時間は、速ければ10秒程度ですが、ネットワークやファイル・システムによっては、数分かかる場合もあります。

### Multi アプリケーションの復旧

アプリケーションの復旧は、フェイルオーバーの後に自動で継続されるアプリケーション処理です。フェイルオーバー後のアプリケーションの復旧は、慎重に設計する必要があります。一部のアプリケーションでは、フェイルオーバーが実行されたことを認識する必要があります。

アプリケーションの復旧の目的は、短い遅延時間でアプリケーション処理を続行することです。新しい処理を続行する前に、アプリケーションは障害時に処理していた作業単位をバックアウトして再実行依頼する必要があります。

アプリケーションの復旧における問題は、IBM MQ MQI client とキュー・マネージャーが共有する、キュー・マネージャーに保管されたコンテキストが失われることです。IBM MQ MQI client はコンテキストの大部分を復元しますが、忠実には復元できない部分があります。以下の各セクションでは、アプリケーションの復旧に関するいくつかのプロパティ、およびそれらが複数インスタンス・キュー・マネージャーに接続したアプリケーションの復旧にどう影響するかについて説明します。

## トランザクション・メッセージング

メッセージ送信の観点からすると、フェイルオーバーによって IBM MQ メッセージングの持続プロパティが変更されることはありません。メッセージが持続メッセージで、作業単位内で正しく管理されている場合、フェイルオーバー中にそれらのメッセージが失われることはありません。

トランザクション処理の観点からすると、フェイルオーバー後にトランザクションはバックアウトされるかまたはコミットされます。

コミットされていないトランザクションはロールバックされます。フェイルオーバーの後に、再接続可能アプリケーションではトランザクションが失敗したことを示す MQRC\_BACKED\_OUT 理由コードを受け取ります。次いでトランザクションを再開する必要があります。

コミット済みトランザクションとは、2 フェーズ・コミットの 2 番目のフェーズに達しているトランザクション、または MQCMIT を開始した単一フェーズ (メッセージのみ) トランザクションのことです。

キュー・マネージャーがトランザクション・コーディネーターであり、障害前に MQCMIT がその 2 フェーズ・コミットの第 2 フェーズを開始済みであった場合、トランザクションは正常に完了します。その完了はキュー・マネージャーの制御下にあり、キュー・マネージャーが再び実行されると続行します。再接続可能アプリケーションでは、MQCMIT 呼び出しは正常に完了します。

メッセージだけが関係する単一フェーズ・コミットでは、コミット処理を開始したトランザクションは、キュー・マネージャーが再度実行すると、その制御下で正常に完了します。再接続可能アプリケーションでは、MQCMIT は正常に完了します。

再接続可能クライアントは、トランザクション・コーディネーターとして機能するキュー・マネージャーの制御下では、単一フェーズのトランザクションを使用できます。拡張トランザクション・クライアントは再接続をサポートしません。トランザクション・クライアントの接続時に再接続が要求される場合、接続は成功しますが、再接続することはできません。その接続は、再接続不可として動作します。

## アプリケーションの再始動または再開

フェイルオーバーはアプリケーションを中断させます。障害解決後、アプリケーションは最初から再始動することもできますし、中断後の処理を再開することもできます。後者の機能のことをクライアントの自動再接続といいます。IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

IBM MQ MQI client・アプリケーションでは、クライアントが自動的に再接続されるように接続オプションを設定できます。該当するオプションは、MQCNO\_RECONNECT または MQCNO\_RECONNECT\_Q\_MGR です。オプションを設定しなければ、クライアントは、自動的に再接続しようとはしません。キュー・マネージャーの障害で、MQRC\_CONNECTION\_BROKEN がクライアントに返されます。新しい MQCONN または MQCONNX 呼び出しを発行することによって新しい接続を試行して開始するようにクライアントを設計することもできます。

サーバー・プログラムは再始動する必要があります。キュー・マネージャーが、これらをキュー・マネージャーまたはサーバーで障害が発生した際に処理していたポイントで自動的に再接続することはできません。複数インスタンス・キュー・マネージャー・インスタンスで障害が発生した場合、IBM MQ サーバー・プログラムは通常、スタンバイ・キュー・マネージャー・インスタンス上では再始動しません。

IBM MQ サーバー・プログラムをスタンバイ・サーバー上で自動的に再始動するには、次の 2 つの方法があります。

1. サーバー・アプリケーションをキュー・マネージャー・サービスとしてパッケージする。これはスタンバイ・キュー・マネージャーの再始動時に再始動します。
2. 例えば、スタンバイ・キュー・マネージャー・インスタンスの開始時に書き込まれるフェイルオーバーのログ・メッセージなどによってトリガーされる独自のフェイルオーバー・ロジックを作成する。その後、アプリケーション・インスタンスは、開始後に MQCONN または MQCONNX を呼び出して、キュー・マネージャーへの接続を作成する必要があります。

## フェイルオーバーの検出

フェイルオーバーを確実に認識する必要があるアプリケーションもあれば、そうでないアプリケーションもあります。以下の 2 つの例について考えてみます。

1. メッセージング・チャンネルを介してメッセージを取得または受信するメッセージング・アプリケーションでは、通常、そのチャンネルの反対側にあるキュー・マネージャーが稼働している必要はありません。したがって、チャンネルの反対側にあるキュー・マネージャーがスタンバイ・インスタンスで再始動しても、影響が出る可能性はほとんどありません。
2. IBM MQ MQI client アプリケーションは、1つのキューからの持続メッセージ入力を処理し、持続メッセージ応答を1つの作業単位の一部として別のキューに書き込みます。作業単位を再始動することによって、同期点内でMQPUT、MQGET、またはMQCMITからのMQRC\_BACKED\_OUT理由コードを処理する場合、メッセージは失われません。さらに、このアプリケーションでは接続障害に対応して何らかの特別な処理を実行する必要もありません。

しかし、2番目の例で、アプリケーションがMQGETのオプションMQGMO\_MSG\_UNDER\_CURSORを使用して、処理すべきメッセージを選択するためにキューを参照しているとします。再接続によってブラウザ・カーソルがリセットされるため、MQGET呼び出しは正しいメッセージを返しませんが、この例の場合、アプリケーションは、フェイルオーバーが実行されたことを認識する必要があります。さらにアプリケーションは、カーソルのあるメッセージに対して別のMQGETを発行する前に、ブラウザ・カーソルを復元する必要があります。

ブラウザ・カーソルの消失は、再接続後にアプリケーション・コンテキストが変化する例の1つです。他のケースは、493ページの『自動再接続されるクライアントの復旧』に記載されています。

フェイルオーバー後のIBM MQ MQI client・アプリケーションには、3つの代替設計パターンがあります。フェイルオーバーを検出する必要がないのは、1つだけです。

### 再接続なし

このパターンでは、接続が失敗した場合に、アプリケーションは現行接続上のすべての処理を停止します。アプリケーションが処理を続行するには、キュー・マネージャーとの新規接続を確立する必要があります。新規接続での処理を続行するために必要な、すべての状態情報の転送全体を受け持つのは、アプリケーションです。接続が失われた後にキュー・マネージャーと再接続する既存のクライアント・アプリケーションについては、この記述が当てはまります。

クライアントは、接続が失われた後に次のMQI呼び出しからMQRC\_CONNECTION\_BROKENまたはMQRC\_Q\_MGR\_NOT\_AVAILABLEなどの理由コードを受け取ります。アプリケーションは、キュー・ハンドルなどのすべてのIBM MQ状態情報を破棄し、新しい接続を確立するために新しいMQCONNまたはMQCONNX呼び出しを発行してから、処理する必要があるIBM MQオブジェクトを再オープンする必要があります。

デフォルトのMQIの動作では、キュー・マネージャーとの接続が失われた後には、キュー・マネージャーの接続ハンドルは使用不可になります。デフォルトは、MQCONNXにMQCNO\_RECONNECT\_DISABLEDオプションを設定して、フェイルオーバー後のアプリケーション再接続が行われないようにすることと同じです。

### フェイルオーバーへの耐性

フェイルオーバーの影響を受けないアプリケーションを作成します。場合によっては、注意深くエラー処理を行うだけでフェイルオーバーを処理できます。

### 再接続の認識

キュー・マネージャーにMQCBT\_EVENT\_HANDLER イベント・ハンドラーを登録します。クライアントがサーバーへの再接続の試行を開始したときにはMQRC\_RECONNECTING、再接続が正常に行われた後にはMQRC\_RECONNECTEDを使用してイベント・ハンドラーに通知が行われます。ユーザーはその後、クライアント・アプリケーションが処理を続行できるように、予測可能な状態を再度確立するためのルーチンを実行できます。

## 自動再接続されるクライアントの復旧

フェイルオーバーは予期しないイベントであり、自動的に再接続されたクライアントが再接続の結果を設計できるように設計されている場合は、予測可能なものでなければなりません。

予測不可の障害を、予測可能で確かなリカバリーに変える主要な要素は、トランザクションの使用です。

前のセクションで、ローカル・トランザクションを使用してMQGETおよびMQPUTを調整するIBM MQ MQI clientの例(493ページの『2』)を示しました。クライアントはMQRC\_BACKED\_OUTエラーに対する応答

として MQCMIT または MQBACK 呼び出しを発行した後、バックアウトされたトランザクションを再実行依頼します。キュー・マネージャーの障害が発生するとトランザクションはバックアウトされ、クライアント・アプリケーションの動作により、トランザクションもメッセージも失われることはありません。

すべてのプログラム状態がトランザクションの一部として管理されるわけではないため、再接続の結果がさらに分かりにくくなります。キュー・マネージャーのフェイルオーバーの影響を受けないクライアント・アプリケーションを設計するには、再接続によって IBM MQ MQI client の状態がどのように変化するかを知る必要があります。

特殊なフェイルオーバー・コードを一切含まないアプリケーションを設計し、再接続エラーを他のエラーと同じロジックで処理するようにすることもできます。あるいは、再接続に特殊なエラー処理を必要とすることが分かった場合は、IBM MQ にイベント・ハンドラーを登録して、フェイルオーバーを処理するルーチンを実行することもできます。ルーチンで再接続処理そのものを扱うこともできますし、メインプログラム・スレッドが処理を再開するときに、復旧処理を実行する必要があることを指示するフラグを設定することもできます。

IBM MQ MQI client 環境はフェイルオーバー自体を認識し、再接続後に可能な限りコンテキストを復元します。そのために、この環境では、クライアント内に何らかの状態情報を格納して、その IBM MQ 状態を回復するためにクライアント・アプリケーションに代わって追加の MQI 呼び出しを発行します。例えば、障害発生時点でオープンされていたオブジェクトへのハンドルが復元され、一時動的キューが同じ名前でもオープンされます。しかし、避けられない変化があるため、こうした変化に対応できる設計が必要です。これらの変化は、以下の 5 種類に分類できます。

1. アプリケーション・プログラムが一貫性のある新規のコンテキスト状態を復元するまで、MQI 呼び出しから新規の (つまり以前に診断されていない) エラーが返される。

新規のエラーを受け取る例として、コンテキストを保存してから再接続するまでの間にコンテキストを渡そうとしたときの戻りコード MQRC\_CONTEXT\_NOT\_AVAILABLE があります。セキュリティー・コンテキストは無許可のクライアント・プログラムには渡されないため、再接続後にコンテキストを復元することはできません。復元してしまうと、悪意のあるアプリケーション・プログラムがセキュリティー・コンテキストを取得してしまう可能性があります。

通常、アプリケーションは、一般的で予測可能なエラーは綿密に設計された方法で処理し、一般的でないエラーについては汎用エラー・ハンドラーに任せます。あらゆるエラーに対応するエラー・ハンドラーは、IBM MQ から切断して再接続したり、場合によってはプログラムを完全に停止したりします。継続性を高めるには、いくつかのエラーを別の方法で処理しなければならない場合もあります。

2. 非持続メッセージが失われることがある。
3. トランザクションがロールバックされる。
4. 同期点外で使用された MQGET 呼び出しまたは MQPUT 呼び出しが中断され、メッセージが失われることがある。
5. MQI 呼び出しでの待機時間の延長により、タイミングに起因したエラーが発生する。

コンテキストの消失に関する詳細を以下のセクションにリストします。

- NPMCLASS(HIGH) オプションを指定してキューに入れていて、かつ、キュー・マネージャーの障害が、シャットダウン時に非持続メッセージを保管するオプションを妨げなかった場合を除き、非持続メッセージは廃棄されます。
- 非持続サブスクリプションは、接続が切断されると失われます。再接続時に再確立されます。永続サブスクリプションを使用することを検討してください。
- get-wait 間隔が再計算され、その限度を超えている場合には、MQRC\_NO\_MSG\_AVAILABLE が返されます。同様に、同じ総有効期限時間となるようにサブスクリプションの有効期限が再計算されます。
- キュー内のブラウズ・カーソルの位置が失われますが、通常、これは最初のメッセージの前に再設定されます。
  - MQGMO\_BROWSE\_MSG\_UNDER\_CURSOR または MQGMO\_MSG\_UNDER\_CURSOR を指定する MQGET 呼び出しが失敗し、理由コード MQRC\_NO\_MSG\_AVAILABLE が返されます。
  - ブラウズのためにロックされたメッセージがアンロックされます。

- ハンドルの有効範囲を持つブラウズ・マーク付きメッセージのマークが解除され、再度ブラウズされることがあります。
- 多くの場合、共有ブラウズ・マーク付きメッセージのマークが解除されます。
- セキュリティー・コンテキストが失われます。MQPMO\_PASS\_ALL\_CONTEXT を指定してメッセージをキューに入れる場合など、メッセージ・コンテキストの保存を使用しようとするとう失敗し、MQRC\_CONTEXT\_NOT\_AVAILABLE が返されます。
- メッセージ・トークンが失われます。メッセージ・トークンを使用した MQGET で、理由コード MQRC\_NO\_MSG\_AVAILABLE が返されます。

注: *MsgId* および *CorrelId* はメッセージの一部であるため、フェイルオーバー時にメッセージとともに保存されます。そのため、*MsgId* または *CorrelId* を使用した MQGET は、予期したとおりに機能します。

- コミットされていないトランザクションの同期点でキューに書き込まれたメッセージは、使用できなくなります。
- 論理順序に従ってメッセージを処理したり、メッセージ・グループ内のメッセージを処理したりすると、再接続後に戻りコード MQRC\_RECONNECT\_INCOMPATIBLE が返されます。
- MQI 呼び出しによって、クライアントが現在通常受け取る、より一般的な MQRC\_CONNECTION\_BROKEN ではなく、MQRC\_RECONNECT\_FAILED が返される場合があります。
- メッセージがキュー・マネージャーに正常に配信されたかどうかを IBM MQ MQI client が認識しない場合、同期点外での MQPUT 呼び出し中の再接続は MQRC\_CALL\_INTERRUPTED を返します。MQCMIT の際の再接続も同様に動作します。
- MQRC\_CALL\_INTERRUPTED が正常な再接続の後に戻されるのは、以下の成功または失敗に関して、IBM MQ MQI client がキュー・マネージャーから応答を受け取らない場合です。
  - 同期点の外部で MQPUT 呼び出しを使用することによる持続メッセージの配信。
  - 同期点の外部で MQPUT1 呼び出しを使用することによる持続メッセージまたはデフォルトで持続となるメッセージの配信。
  - MQCMIT 呼び出しを使用することによるトランザクションのコミット。再接続が正常に行われた後にのみ、応答が戻されます。
- チャンネルは新規インスタンスとして再始動するため(異なるチャンネルになる場合もあります)、チャンネル出口の状態は保持されません。
- 一時動的キューは、一時動的キューが開かれていた再接続可能クライアントの復旧プロセスの一部として復元されます。一時動的キュー上のメッセージは復元されませんが、キューが開かれていた、つまりキューの名前を保存していたアプリケーションは、処理を続行できます。

キューがそれを作成したアプリケーション以外のアプリケーションによって使用されている場合、キューは復元が間に合わないために次に参照される時点でまだ存在しない可能性があります。例えば、クライアントが応答先キューとして一時動的キューを作成し、そのキューにチャンネルが応答メッセージを置く場合は、キューの復旧が間に合わない可能性があります。この場合、チャンネルは通常、応答先メッセージを送達不能キューに置くことになります。

再接続可能なクライアント・アプリケーションが一時動的キューを(別のアプリケーションが既にそのキューを作成したために)名前を指定して開いた場合、再接続の発生時に、IBM MQ MQI client が一時動的キューを再作成できなくなります。これは、作成の基になるモデルがないためです。MQI では、モデルで一時動的キューを開けるのは単一のアプリケーションのみです。一時動的キューの使用を必要とする他のアプリケーションは、MQPUT1 またはサーバー・バイndィングを使用する必要があります。また、失敗した場合に、再接続を再び試行できるようにする必要があります。

一時動的キューに入れることができるのは非持続メッセージのみで、それらのメッセージはフェイルオーバー中に失われます。失われるのは、再接続中に MQPUT1 を使用して一時動的キューに入れられたメッセージです。MQPUT1 中にフェイルオーバーが発生した場合、MQPUT1 が成功してもメッセージは入れられません。この問題の回避策の1つは、永続動的キューを使用することです。サーバー・バイndィング・アプリケーションはすべて、再接続できないため、一時動的キューを名前で開くことができます。

## Multi データ・リカバリーおよび高可用性

複数インスタンス・キュー・マネージャーを使用した高可用性ソリューションには、ストレージの故障後にデータをリカバリーするメカニズムが含まれている必要があります。

複数インスタンス・キュー・マネージャーを使用した場合、キュー・マネージャー・プロセスの可用性は向上しますが、キュー・マネージャーがメッセージや他の情報を保管するために使用するファイル・システムなどの他のコンポーネントの可用性は向上しません。

データの可用性を高める1つの方法は、回復力のあるネットワーク・データ・ストレージを使用することです。ネットワーク・ファイル・システムおよび回復力のあるデータ・ストレージを使用して独自のソリューションを構築することも、統合ソリューションを購入することもできます。回復力と災害復旧を組み合わせたものとしては、数十または数百キロメートル離れた場所でのディスク複製を可能にする非同期ディスク複製があります。

ストレージ・メディアを最大限に活用できるように、IBM MQ ディレクトリーをメディアに異なる方法でマップするように構成することができます。複数インスタンス・キュー・マネージャーでは、IBM MQ のディレクトリーおよびファイルの2つのタイプの間には重要な区別があります。

### キュー・マネージャーのインスタンス間で共有する必要があるディレクトリー。

キュー・マネージャーの各インスタンスの間で共有しなければならない情報は、qmgrs ディレクトリーおよび logs ディレクトリーという2つのディレクトリーの中にあります。これらのディレクトリーは、共有ネットワーク・ファイル・システム上になければなりません。メッセージの作成や削除に伴ってデータが絶えず変わるので、持続的な高可用性と優れたパフォーマンスを備えたストレージ・メディアを使用することをお勧めします。

### キュー・マネージャーのインスタンス間で共有する必要がないディレクトリーおよびファイル。

他のディレクトリーの中には、キュー・マネージャーの各インスタンス間で共有する必要がないために、ミラーリングされたファイル・システムを使用する以外の方法で短時間でリストアされるものもあります。

- IBM MQ 実行可能ファイルおよびツール用ディレクトリー。再インストールするか、バックアップを行いバックアップされたファイル・アーカイブからリストアすることによって置き換えます。
- インストール済み環境全体に関して変更される構成情報。構成情報は、IBM MQ (Windows、UNIX and Linux システム上の mqs.ini ファイルなど) によって管理されるか、独自の構成管理の一部 (MQSC 構成スクリプトなど) によって管理されます。バックアップを行い、ファイル・アーカイブを使用してリストアします。
- トレース、エラー・ログ、FFDC ファイルなど、インストール済み環境全体の出力。これらのファイルは、デフォルトのデータ・ディレクトリーの errors および trace サブディレクトリーに保管されます。UNIX and Linux システム上のデフォルトのデータ・ディレクトリーは、/var/mqm です。Windows の場合、デフォルトのデータ・ディレクトリーは IBM MQ のインストール・ディレクトリーです。

バックアップ・キュー・マネージャーにより、複数インスタンス・キュー・マネージャーのメディア・バックアップをリニア・ログを使用して定期的に行うこともできます。バックアップ・キュー・マネージャーによるリカバリーは、ミラーリングされたファイル・システムからのリカバリーほど高速ではありません。また、最新バックアップ以降に行われた変更はリカバリーされません。バックアップ・キュー・マネージャー・メカニズムは、ローカルのストレージ故障の後でキュー・マネージャーをリカバリーする場合よりも、オフサイトの災害復旧シナリオで使用する場合に適しています。

## IBM MQ 可用性ソリューションの結合

アプリケーションでは、可用性を向上させるために IBM MQ の他の機能を使用します。複数インスタンス・キュー・マネージャーは、他の高可用性機能を補完します。

### IBM MQ クラスターによるキューの可用性の向上

1つのクラスター・キュー (クラスター内の各マネージャー上のすべてのキューのうちの最大で1つ) に複数の定義を作成することで、キューの可用性を向上させることができます。

クラスターの1メンバーに障害が起こった後に新しいメッセージがクラスター・キューに送信されるとします。障害が起きたキュー・マネージャーにメッセージが送信される必要がない限り、そのメッセージは、キューの定義がある、クラスター内の別の実行中のキュー・マネージャーに送信されます。

クラスターにより可用性は大幅に向上しますが、結果的にメッセージの遅延となる、関連する2つの障害シナリオがあります。複数インスタンス・キュー・マネージャーを使用してクラスターを構築すると、メッセージが遅延する可能性が減ります。

### 孤立メッセージ

クラスター内のキュー・マネージャーに障害が起きた場合、クラスター内の他のキュー・マネージャーに送信可能なメッセージは、障害が起きたキュー・マネージャーにはそれ以上送信されません。既に送信済みのメッセージは、障害が起きたキュー・マネージャーが再始動するまで孤立します。

### アフィニティー

アフィニティーとは、本来は別のものである2つの処理の間で共有される情報を表すために使用される用語です。例えば、サーバーに要求メッセージを送信するアプリケーションと、応答を処理することになっている同じアプリケーションの間には、アフィニティーが存在します。その他の例としては、各メッセージの処理が先行するメッセージに依存する、メッセージのシーケンスなどがあります。

メッセージをクラスター・キューに送信する場合は、アフィニティーを考慮する必要があります。つまり、連続するメッセージを同じキュー・マネージャーに送信する必要があるか、またはそれぞれのメッセージをクラスター内のどのメンバーにも送信できるか、などです。

クラスター内の同じキュー・マネージャーにメッセージを送信する必要があり、そのキュー・マネージャーに障害が発生した場合には、障害が起きたクラスター・キュー・マネージャーが再び実行されるまで、そのメッセージは送信元の伝送キュー内で待機します。

クラスターが複数インスタンス・キュー・マネージャーを使用して構成されている場合は、障害が起きたキュー・マネージャーの再始動を待機する遅延はほんの数分程度に制限され、スタンバイにより引き継がれます。スタンバイが実行中になっていれば、孤立メッセージの処理が再開され、新たにアクティブ化されたキュー・マネージャー・インスタンスへのチャンネルが開始され、伝送キュー内で待機していたメッセージが流れ始めます。

障害が起きたキュー・マネージャーによるメッセージ遅延を克服するようにクラスターを構成する可能な1つの方法は、クラスター内の各サーバーに対して2つの異なるキュー・マネージャーをデプロイし、その異なるキュー・マネージャーの1つはアクティブ・インスタンスとして、もう1つはスタンバイ・インスタンスとして配置することです。これはアクティブ/スタンバイ構成であり、クラスターの可用性を向上させます。

クラスターは、管理の手間が減り、スケーラビリティが向上するという利点があることに加え、複数インスタンス・キュー・マネージャーを補完する可用性の追加要素を継続的に提供します。クラスターには、キュー・マネージャーのアクティブ・インスタンスとスタンバイ・インスタンスの両方に影響する他のタイプの障害からの保護機能があります。

### 無中断サービス

クラスターは無中断サービスを提供します。クラスターが受信した新規のメッセージは、アクティブ・キュー・マネージャーに送信されて処理されます。無中断サービスの提供を複数インスタンス・キュー・マネージャーに依存しないようにしてください。なぜなら、スタンバイ・キュー・マネージャーが障害を検出して開始を完了し、そのチャンネルが再接続され、失敗したメッセージのバッチが再実行依頼されるまでに時間を要するからです。

### 局所的障害

許容可能なパフォーマンスを実現するには、アクティブ・サーバー、スタンバイ・サーバー、およびファイル・システム・サーバーはミリ秒単位の速度で対話する必要があるため、それらのサーバーの物理的距離には実用面での制約があります。

クラスター・キュー・マネージャーで必要になる対話速度はほぼ秒単位であり、これらのキュー・マネージャーは地理的に世界のどこにでも分散させて配置できます。

### 操作上のエラー

異なる2つのメカニズムを使用して可用性を向上させることにより、人為的エラーなどの操作上のエラーが発生して、可用性の実現が危うくなる可能性を減らすことができます。

## キュー共有グループによるメッセージ処理の可用性の向上

**z/OS** キュー共有グループ (z/OS 上でのみ提供) を使用すると、キュー・マネージャーのグループがキューのサービス提供を共有できるようになります。あるキュー・マネージャーで障害が起きた場合には、引き続き他のキュー・マネージャーがそのキューにあるすべてのメッセージを処理します。複数インスタンス・キュー・マネージャーは、z/OS ではサポートされておらず、幅広いメッセージング・アーキテクチャーの中の限られた一部分としてのみキュー共有グループを補完します。

## IBM MQ クライアントによるアプリケーションの可用性の向上

IBM MQ MQI client プログラムは、キュー・マネージャーの可用性、接続の重み、およびアフィニティーに基づいて、1つのキュー・マネージャー・グループ内の複数の異なるキュー・マネージャーに接続できます。キュー・マネージャーが実行しているマシンとは異なるマシン上でアプリケーションを実行することで、ソリューションの総合的な可用性を向上させることができます。ただしこれは、接続先のキュー・マネージャー・インスタンスに障害が起きた場合に、アプリケーションを再接続する方法があることが前提です。

キュー・マネージャー・グループを使用すると、停止したキュー・マネージャーとクライアントの結合が解除され、IP スプレイヤーのように、キュー・マネージャーのグループ内でクライアント接続のロード・バランシングが行われるため、クライアントの可用性が向上します。クライアント・アプリケーションには、障害が起きたキュー・マネージャーとのアフィニティー (特定のキューへの依存など) があってはなりません。そうでないと、処理を再開できません。

クライアントの自動再接続および複数インスタンス・キュー・マネージャーを使用すると、いくつかのアフィニティーの問題が解決されるため、クライアントの可用性は向上します。IBM MQ classes for Java は自動クライアント再接続をサポートしていません。

MQCNO のオプション MQCNO\_RECONNECT\_Q\_MGR を設定して、クライアントを同じキュー・マネージャーに強制的に再接続させることができます。

1. 以前に接続していた単一インスタンスのキュー・マネージャーが実行していない場合には、そのキュー・マネージャーが再度実行中になるまで接続が再試行されます。
2. キュー・マネージャーが複数インスタンス・キュー・マネージャーとして構成されている場合、クライアントはアクティブであるいずれかのインスタンスに再接続します。

同じキュー・マネージャーに自動的に再接続することで、クライアントの代わりにキュー・マネージャーが保持していた多くの状態情報 (開いたキューや、サブスクライブしていたトピック) は、復元されます。要求への応答を受け取るためにクライアントが動的応答先キューを開いていた場合、応答先キューへの接続も復元されます。

## Linux V9.0.4 RDQM 高可用性

RDQM (複製データ・キュー・マネージャー) は、Linux プラットフォーム上で使用できる高可用性ソリューションです。

RDQM 構成は、高可用性 (HA) グループとして構成されている3つのサーバーから成り、その個々にキュー・マネージャーのインスタンスがあります。インスタンスの1つは実行中のキュー・マネージャーで、データを他の2つのインスタンスに同期的に複製します。このキュー・マネージャーを実行しているサーバーに障害が発生すると、別のキュー・マネージャーのインスタンスが開始され、このインスタンスには操作に使用する現行データがあります。キュー・マネージャーのインスタンスは3つとも浮動 IP アドレスを共有しているので、クライアントの構成時に使用する必要がある IP アドレスは1つだけです。ネットワークの問題のために HA グループがパーティション化されていても、同時に実行できるキュー・マネージャーのインスタンスは1つだけです。キュー・マネージャーを実行しているサーバーは「1次」と呼ばれ、その他の2つのサーバーはどちらも「2次」と呼ばれます。

3つのノードを使用すると、スプリット・ブレンという状態になる可能性が大幅に減少します。2ノードの高可用性システムでは、2つのノード間の接続が切断されるとスプリット・ブレンが発生する可能性があります。接続のない状態だと、両方のノードでキュー・マネージャーが同時に実行され、別々のデータが累積される可能性があります。そうすると、接続が復元された時に2つの別々のバージョンのデータ (「スプリット・ブレン」) が存在することになるので、保持するデータ・セットと破棄するデータ・セットを決定するために手操作による介入が必要になります。

RDQM では、スプリット・ブレイン状態を回避するために、クォーラムを設定した 3 ノード・システムを使用します。1つのノードが他の1つ以上のノードと通信できる場合は、その通信可能なノード同士がクォーラムを形成します。キュー・マネージャーは、クォーラムを形成するノードでしか実行できません。他の1つ以上のノードと接続していないノードではキュー・マネージャーを実行できないので、接続の切れた2つのノードでキュー・マネージャーが同時に実行されることは絶対にありません。

- 1つのノードで障害が発生しても、他の2つのノードのいずれかでキュー・マネージャーを実行できます。2つのノードで障害が発生すると、残りの1つのノードではキュー・マネージャーを実行できません。そのノードはクォーラムを形成していないからです(残りのノードは、他の2つのノードで障害が失敗したのか、それとも他の2つのノードは実行中だけれども接続が失われたのかを判別できません)。
- 1つのノードが接続を失うと、そのノードではキュー・マネージャーを実行できません。そのノードはクォーラムを形成していないからです。残りの2つのノードのいずれかではキュー・マネージャーを実行できます。その2つがクォーラムを形成しているからです。すべてのノードが接続を失うと、どのノードでもキュー・マネージャーを実行できません。どのノードもクォーラムを形成していないからです。

注：IBM MQ Console は、複製データ・キュー・マネージャーをサポートしていません。IBM MQ Explorer は複製データ・キュー・マネージャーに使用できますが、RDQM 機能に固有の情報は表示されません。

3つのノードのグループ構成は、Pacemaker が実施します。3つのノード間の複製は、DRBD が実施します。(Pacemaker については <https://clusterlabs.org/pacemaker/> を、DRBD については <https://docs.linbit.com/docs/users-guide-9.0/> を参照してください。)

複製データ・キュー・マネージャーは、556 ページの『キュー・マネージャー・データのバックアップ』で説明しているプロセスを使用してバックアップできます。このキュー・マネージャーを停止してバックアップしても、RDQM 構成で実行されているノード・モニターには影響を与えません。

以下の図に、HA グループ内の3つの各ノード上で1つずつ RDQM が実行されている標準的なデプロイメントを示します。

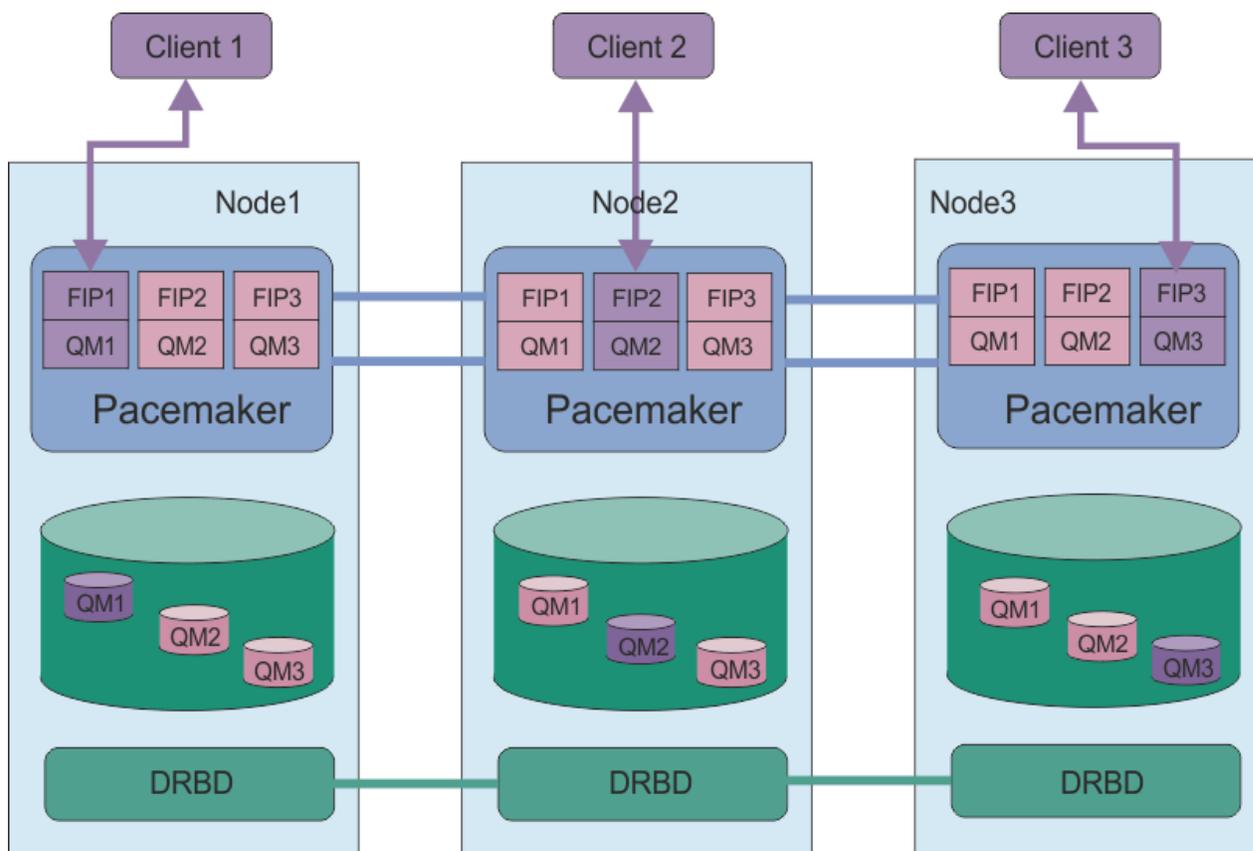


図 80. 3つの RDQM がある HA グループの例

以下の図では、Node3 で障害が発生し、Pacemaker リンクが失われ、キュー・マネージャー QM3 が Node2 で実行されています。

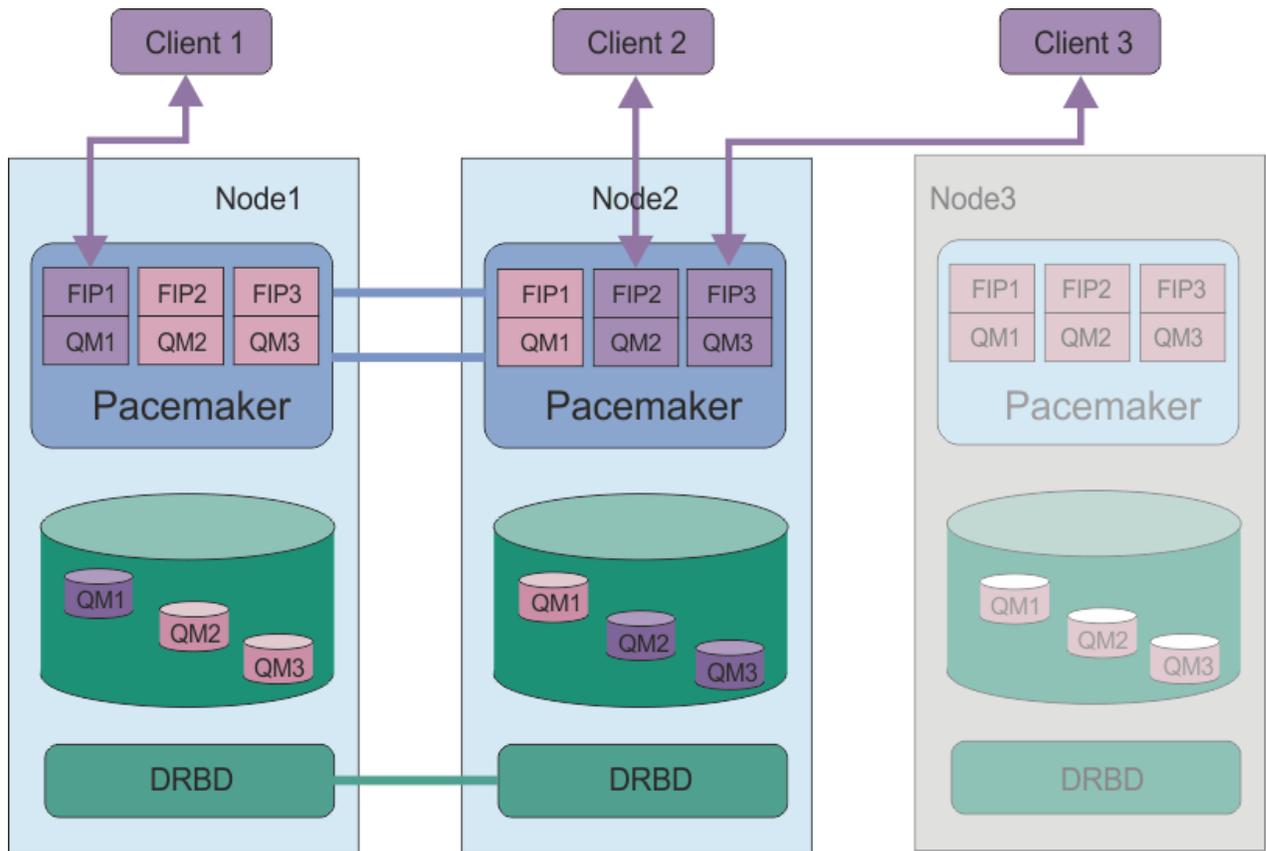


図 81. Node3 で障害が発生した後の例

### 関連情報

[RDQM \(複製データ・キュー・マネージャー\) のインストール](#)

[複製データ・キュー・マネージャーのマイグレーション](#)

### Linux V 9.0.4 RDQM HA ソリューションの要件

RDQM 高可用性 (HA) グループを構成する前に、いくつかの要件を満たしている必要があります。

### システム要件

RDQM HA グループを構成する前に、HA グループの一部となる 3 つのサーバーのそれぞれで、いくつかの構成を完了する必要があります。

- 各ノードには、`drbdpool` という名前のボリューム・グループが必要です。各複製データ・キュー・マネージャーのストレージは、このボリューム・グループからのキュー・マネージャーごとに別個の論理ボリュームとして割り振られます。最高のパフォーマンスを得るためには、このボリューム・グループを内部ディスク・ドライブ (SSD) に対応する 1 つ以上の物理ボリュームで構成する必要があります。RDQM HA ソリューションをインストールした後、実際に RDQM を作成する前に、`drbdpool` を作成する必要があります。 `vgs` コマンドを使用してボリューム・グループ構成を確認します。出力は次のようになります。

```

VG      #PV #LV #SN Attr   VSize  VFree
drbdpool 1   9   0 wz--n- <16.00g <7.00g
rhe1    1   2   0 wz--n- <15.00g  0

```

特に、属性の 6 番目の列に `c` 文字 (つまり、`wz--nc`) がいないことを確認してください。 `c` は、クラスタリングが有効になっていることを示します。有効になっている場合は、ボリューム・グループを削除して、クラスタリングなしで再作成する必要があります。

- `drbdpool` ボリューム・グループを作成したら、それに対してなにも行わないでください。 IBM MQ が `drbdpool` に作成された論理ボリュームのマウント方法とマウント場所を管理します。

- 各ノードには、RDQM サポートの構成に使用されるインターフェースが最大で 3 つ必要です。
    - HA グループをモニターするための Pacemaker の基本インターフェース。
    - HA グループをモニターするための Pacemaker の代替インターフェース。
    - 同期データ複製用のインターフェース。複製インターフェースと呼ばれます。HA グループで実行されるすべての複製データ・キュー・マネージャーの予期されるワークロードを前提として、複製要件をサポートするために十分な帯域幅が必要です。
- 3 つのすべてのインターフェースに同じ IP アドレスを使用するか、各インターフェースに別個の IP アドレスを使用するか、または基本インターフェースと代替インターフェースに同じ IP アドレスを使用して、複製インターフェースに別個の IP アドレスを使用するように HA グループを構成できます。
- 最大限の耐障害性を実現するためには、これらのインターフェースは独立ネットワーク・インターフェース・カード (NIC) である必要があります。
- RFC 1123 で修正された RFC 952 で定義されているように、DRBD では、HA グループの各ノードに有効なインターネット・ホスト名 (uname -n によって返された値) がある必要があります。
  - HA グループのノード間にファイアウォールがある場合は、ファイアウォールでポートの範囲のノード間のトラフィックが許可される必要があります。RHEL で標準ファイアウォールを実行している場合に必要ポートを開くサンプル・スクリプトが用意されています (/opt/mqm/samp/rdqm/firewalld/configure.sh)。このスクリプトは root として実行する必要があります。他のファイアウォールを使用している場合は、サービス定義 /usr/lib/firewalld/services/rdqm\* を調べて、開く必要があるポートを確認してください。
  - SELinux を permissive 以外のモードで使用しているシステムでは、以下のコマンドを実行する必要があります。

```
semanage permissive -a drbd_t
```

## ネットワーク要件

RDQM HA グループの 3 つのノードを同じデータ・センターに配置することをお勧めします。

そうしたノードを別々のデータ・センターに配置する場合は、以下の制約に注意する必要があります。

- データ・センター間の待ち時間が長くなると、パフォーマンスが急速に低下します。IBM では最大 5 ミリ秒の待ち時間をサポートしていますが、アプリケーション・パフォーマンスの観点からすれば、待ち時間が 1 から 2 ミリ秒より長くなるのは許容できないかもしれません。
- 複製リンクで送信するデータは、IBM MQ AMS での暗号化は別として、それ以外の暗号化の対象になりません。

浮動 IP アドレスを構成すれば、HA グループ内のどのノードでクライアントが実行されているかに関わりなく、クライアントが複製データ・キュー・マネージャー (RDQM) に関して同じ IP アドレスを使用することが可能になります。浮動アドレスは、RDQM の 1 次ノードの指定した物理インターフェースにバインドします。RDQM がフェイルオーバーし、別のノードが 1 次ノードになると、浮動 IP は、新しい 1 次上の同じ名前前のインターフェースにバインドされます。3 つのノードの物理インターフェースは、すべて同じ名前前でなければなりません。また、浮動 IP アドレスと同じサブネットに属していることが必要です。

## クラスターを構成するためのユーザー要件

RDQM HA グループは、ユーザー root として構成できます。root として構成しない場合は、代わりに mqm グループのユーザーとして構成します。mqm ユーザーが RDQM クラスターを構成するには、以下の要件を満たす必要があります。

- mqm ユーザーは、sudo を使用して、RDQM HA グループを構成する 3 つの各サーバーでコマンドを実行できる必要があります。
- mqm ユーザーが RDQM HA グループを構成する 3 つの各サーバーでパスワードなしの SSH を使用してコマンドを実行できる場合、このユーザーは 1 つのサーバーでのみコマンドを実行する必要があります。
- mqm ユーザー用にパスワードなしの SSH を構成する場合、このユーザーの UID が 3 つのサーバーすべてで同じである必要があります。

mqm ユーザーが root 権限で以下のコマンドを実行できるように sudo を構成する必要があります。

```
/opt/mqm/bin/crtmqm  
/opt/mqm/bin/dltmqm  
/opt/mqm/bin/rdqmadm  
/opt/mqm/bin/rdqmstatus
```

## キュー・マネージャーで作業するためのユーザー要件

複製データ・キュー・マネージャー (RDQM) を作成、削除、または構成するには、mqm グループと haclient グループの両方に属するユーザー ID を使用する必要があります (haclient グループは Pacemaker のインストール中に作成されます)。

Linux V 9.0.4 パスワードなしの SSH のセットアップ

HA グループの 1 つのノードで構成コマンドを発行するだけで済むように、パスワードなしの SSH をセットアップできます。

## このタスクについて

パスワードなしの SSH をセットアップするには、各ノードで mqm ID を構成して、そのユーザーに対して各ノードで鍵を生成する必要があります。次に、鍵を他のノードに配布して、接続をテストし、既知のホストのリストに各ノードを追加します。最後に、mqm ID をロックダウンします。

注: この説明は、個別の 1 次、代替、および複製のインターフェースで HA グループを定義していることが前提となっています。したがって、1 次および代替のインターフェースを介してパスワードなしの SSH アクセスを定義します。単一 IP アドレスでシステムを構成する予定の場合は、その単一インターフェースを介してパスワードなしの SSH アクセスを定義します。

## 手順

1. 3 つのノードそれぞれで以下の手順を実行し、mqm ユーザーをセットアップし、SSH 鍵を生成します。

a) mqm ホーム・ディレクトリーを /home/mqm に変更します。

```
usermod -d /home/mqm mqm
```

b) /home/mqm ディレクトリーを作成します。

```
mkhomedir_helper mqm
```

c) mqm パスワードを追加します。

```
passwd mqm
```

d) 対話式シェルを mqm として実行します。

```
su mqm
```

e) mqm 認証鍵を生成します。

```
ssh-keygen -t rsa -f /home/mqm/.ssh/id_rsa -N ''
```

2. 3 つのノードそれぞれで以下の手順を実行し、ノードの鍵を他の 2 つのノードに追加し、各ノードの 1 次および代替 (使用している場合) のアドレスの接続をテストします。

a) リモート・ノードに鍵を追加します。

```
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_primary_address  
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_alternate_address  
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_primary_address  
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_alternate_address
```

b) パスワードなしの SSH を確認して、リモート・ノードの known\_hosts を更新します。

```
ssh remote_node1_primary_address uname -n
ssh remote_node1_alternate_address uname -n
ssh remote_node2_primary_address uname -n
ssh remote_node2_alternate_address uname -n
```

接続ごとに、続行するかどうかを確認するプロンプトが出されます。それぞれが `known_hosts` を更新できるように確認します。パスワードなしの SSH を使用して HA グループを構成する前に、これを完了する必要があります。

- c) `mqm` として対話式シェルを終了します。

```
exit
```

3. 各ノードで `root` として以下の手順を実行し、`mqm` パスワードを削除して ID をロックします。

- a) `mqm` パスワードを削除します。

```
passwd -d mqm
```

- b) `mqm` をロックします。

```
passwd -l mqm
```

4. 各ノードで `root` として次の手順を実行して、`mqm` ユーザーの `sudo` アクセス権限をセットアップします。

- a) 次のようにしてディレクトリーを `/etc` に変更し、`sudoers` ファイルを編集します。

```
cd /etc
vi sudoers
```

- b) 「`### Allows people in group wheel to run all commands`」という行を探し、その行の下に次のテキストを追加します。

```
##mqm ALL=(ALL) ALL
```

- c) 「`### Same thing without a password`」という行を探し、その行の下に次のテキストを追加します。

```
%mqm ALL=(ALL) NOPASSWD: ALL
```

## Linux 9.0.4 Pacemaker クラスタ (HA グループ) の定義

HA グループは、Pacemaker クラスタです。Pacemaker クラスタを定義するには、`/var/mqm/rdqm.ini` ファイルを編集し、`rdqmadm` コマンドを実行します。

### このタスクについて

Pacemaker については、<https://clusterlabs.org/pacemaker/>を参照してください。ユーザーが `sudo` を使用できる場合は、`mqm` グループのユーザーとして Pacemaker クラスタを作成できます。ユーザーが各サーバーに対してパスワードなしの SSH を使用することもできる場合は、`rdqm.ini` ファイルを編集し、いずれかのサーバーで `rdqmadm` を実行するだけで Pacemaker クラスタを作成できます。それ以外の場合は、ファイルを作成し、ノードとなる各サーバー上で `root` としてコマンドを実行する必要があります。

`rdqm.ini` ファイルは、Pacemaker クラスタ内のすべてのノードの IP アドレスを提供します。Pacemaker クラスタで IP アドレスを 1 つ、2 つ、または 3 つ使用するように指定できます。同期データ複製に使用されるインターフェースの名前は、「複製インターフェース」です。HA グループ内で実行されているすべての RDQM の予期されるワークロードを前提として、複製要件をサポートするためには、インターフェースに十分な帯域幅がある必要があります。1 次インターフェースおよび 2 次インターフェースは、Pacemaker でシステムをモニターするために使用されますが、Pacemaker は、必要に応じて複製インターフェースをこの目的のために使用することができます。

以下のサンプル・ファイルは、各インターフェースに別個の IP アドレスを使用するサンプル Pacemaker クラスターの構成を示しています。

```
Node:
  HA_Primary=192.168.4.1
  HA_Alternate=192.168.5.1
  HA_Replication=192.168.6.1
Node:
  HA_Primary=192.168.4.2
  HA_Alternate=192.168.5.2
  HA_Replication=192.168.6.2
Node:
  HA_Primary=192.168.4.3
  HA_Alternate=192.168.5.3
  HA_Replication=192.168.6.3
```

以下のサンプル・ファイルは、各インターフェースに同じ IP アドレスを使用するサンプル Pacemaker クラスターの構成を示しています。この場合、複製インターフェースのみを指定します。

```
Node:
  HA_Replication=192.168.4.1
Node:
  HA_Replication=192.168.4.2
Node:
  HA_Replication=192.168.4.3
```

2つの IP アドレスを使用する場合、rdqm.ini ファイルには、各ノードの Primary フィールドと Replication フィールドがありますが、Alternate フィールドはありません。

```
Node:
  HA_Primary=192.168.4.1
  HA_Replication=192.168.5.1
Node:
  HA_Primary=192.168.4.2
  HA_Replication=192.168.5.2
Node:
  HA_Primary=192.168.4.3
  HA_Replication=192.168.5.3
```

## 手順

- Pacemaker クラスターを mqm ユーザーとして定義するには、次の手順を実行します。
  - a) ユーザー mqm が **sudo** を使用してコマンドを実行できること、およびオプションでパスワードなしの SSH を使用して各サーバーに接続できることを確認してください。
  - b) /var/mqm/rdqm.ini ファイルで Pacemaker クラスターが定義されるように、3つのサーバーのいずれかでこのファイルを編集します。
  - c) 以下のコマンドを実行します。

```
rdqmadm -c
```

(パスワードなしの SSH を使用できない場合は、.ini ファイルを各サーバーにコピーし、各サーバーでコマンドを実行する必要があります。)

- Pacemaker クラスターを root ユーザーとして定義するには、次の手順を実行します。
  - a) ファイルでクラスターが定義されるように、3つのサーバーのいずれかで /var/mqm/rdqm.ini ファイルを編集します。
  - b) このファイルを、Pacemaker クラスター内のノードとなる他の2つのサーバーにコピーします。
  - c) 3つのサーバーそれぞれで、以下のコマンドを root として実行します。

```
rdqmadm -c
```

## 関連情報

[rdqmadm \(複製データ・キュー・マネージャー・クラスターの管理\)](#)

Linux

V 9.0.4

Pacemaker クラスタ (HA グループ) の削除

HA グループは、Pacemaker クラスタです。-u オプションを指定して **rdqmadm** コマンドを実行すると、Pacemaker クラスタ構成を削除できます。

## このタスクについて

いずれかのノードに複製データ・キュー・マネージャーが存在する場合は、Pacemaker クラスタ構成を削除できません。

## 手順

- Pacemaker クラスタ構成を削除するには、任意のノードから以下のコマンドを入力します。

```
rdqmadm -u
```

## 関連情報

[rdqmadm \(複製データ・キュー・マネージャー・クラスターの管理\)](#)

Linux

V 9.0.4

HA RDQM の作成

**crtmqm** コマンドを使用して、高可用性複製データ・キュー・マネージャー (RDQM) を作成します。

## このタスクについて

mqm ユーザーが **sudo** を使用できる場合は、mqm グループのユーザーとして高可用性複製データ・キュー・マネージャー (RDQM) を作成できます。ユーザーが各サーバーに対してパスワードなしの SSH を使用することもできる場合は、1つのノードで **create RDQM** コマンドを実行するだけで3つすべてのノードで RDQM を作成できます。その他の場合、RDQM を作成するには **root** である必要があります、3つのすべてのノードでコマンドを実行する必要があります。

## 手順

- mqm グループのユーザーとして RDQM を作成するには、次のようにします。
  - a) mqm ユーザーが **sudo** を使用してコマンドを実行できること、およびパスワードなしの SSH を使用して各サーバーに接続できることを確認します。
  - b) 次のコマンドを入力します。

```
crtmqm -sx [-fs FilesystemSize] qmname
```

ここで、*qmname* は、複製データ・キュー・マネージャーの名前です。オプションでキュー・マネージャーのファイル・システム・サイズ (drbdpool ポリウム・グループで作成される論理ポリウムのサイズ) を指定できます。

このコマンドは、mqm ユーザーとして SSH を使用して、クラスター内の他のノードへの接続を試行します。接続が成功すると、ノードでキュー・マネージャーの2次インスタンスが作成されます。成功しなかった場合は、2次インスタンスを作成して、**crtmqm -sx** コマンドを実行する必要があります (ユーザー **root** 用の説明に従います)。

- **root** ユーザーとして RDQM を作成するには、次のようにします。
  - a) RDQM の2次インスタンスをホストする各ノードで以下のコマンドを入力します。

```
crtmqm -sxs [-fs FilesystemSize] qmname
```

ここで、*qmname* は、複製データ・キュー・マネージャーの名前です。オプションでキュー・マネージャーのファイル・システム・サイズ (drbdpool ポリウム・グループで作成される論理ポリウ

ムのサイズ)を指定できます。HAグループ内の3つのノードすべてのRDQMに同じファイル・システム・サイズを指定する必要があります。

このコマンドは、RDQMの2次インスタンスを作成します。

b) 残りのノードで、以下のコマンドを入力します。

```
crtmqm -sx [-fs FilesystemSize] qmname
```

ここで、*qmname* は、複製データ・キュー・マネージャーの名前です。オプションでキュー・マネージャーのファイル・システム・サイズを指定できます。

このコマンドは、キュー・マネージャーの2次インスタンスが他の2つのノードに存在するかどうかを判別します。2次インスタンスが存在する場合、コマンドは、1次キュー・マネージャーを作成し、開始します。2次が存在しない場合は、各ノードで **crtmqm -sxs** コマンドを実行するように指示されます。

DataPath (**-md**) 引数および LogPath (**-ld**) 引数を除き、標準 Linux キュー・マネージャーの作成に有効なすべての引数は、1次複製データ・キュー・マネージャーにも有効です。

## 関連情報

[crtmqm](#)

Linux V 9.0.4 HA RDQM の削除

**dltmqm** コマンドを使用して、高可用性複製データ・キュー・マネージャー (RDQM) を削除します。

## このタスクについて

RDQMの1次ノードからRDQMを削除するには、コマンドを実行する必要があります。最初にRDQMを終了する必要があります。mqmユーザーが必要なsudo特権を持っている場合は、mqmユーザーとしてコマンドを実行できます。その他の場合は、rootとしてコマンドを実行する必要があります。1次キュー・マネージャーに関連したリソースが削除された後、コマンドは他のノードに接続するためのsshを使用して、2次キュー・マネージャーの削除を試みます。この削除が失敗した場合は、他のノードでdltmqmを手動で実行してプロセスを完了する必要があります。1次キュー・マネージャーがまだ削除されていない場合は、2次ノードでコマンドが失敗します。

## 手順

- RDQMを削除するには、以下のコマンドを入力します。

```
dltmqm RDQM_name
```

## 関連情報

[dltmqm](#)

Linux V 9.0.4 RDQM の優先ロケーションの設定

複製データ・キュー・マネージャー (RDQM) の優先ロケーションは、そのノードが使用可能な場合にRDQMが実行されるノードを識別します。

## このタスクについて

優先ロケーションは、HAグループが通常の状態(すべてのノードおよび接続が使用可能)である場合に、Pacemakerがキュー・マネージャーを実行する必要があるノードの名前です。優先ロケーションは、キュー・マネージャーの作成時に1次ノードの名前に初期化されます。コマンドを実行して、3つのノードのいずれかに優先ロケーションを設定できます。この場合、mqmグループおよびhaclientグループに属するユーザーである必要があります。

## 手順

- ローカル・ノードまたは指定したノードを、指定したキュー・マネージャーの優先ロケーションとして割り当てるには、以下のコマンドを入力します。

```
rdqmadm -p -m qmname [ -n nodename[,nodename ]
```

ここで、*qmname* は優先ロケーションを指定する RDQM の名前で、*nodename* はオプションの優先ノードの名前です。

HA グループが通常の状態、優先ロケーションが現在の 1 次ノードではない場合、キュー・マネージャーが停止し、新しい優先ロケーションで再始動されます。2 つのノード名のコンマ区切りリストを指定して、優先ロケーションの 2 つ目の設定を割り当てることができます。

- キュー・マネージャーがリストア時に自動的にノードに戻ることがないように、優先ロケーションをクリアするには、以下のコマンドを入力します。

```
rdqmadm -p -m qmname -d
```

## 関連情報

[rdqmadm \(複製データ・キュー・マネージャー・クラスターの管理\)](#)

Linux

V 9.0.4

### 浮動 IP アドレスの作成および削除

浮動 IP アドレスを使用すると、HA グループ内のどのノードでクライアントが実行されているかにかかわらず、複製データ・キュー・マネージャー (RDQM) と同じ IP アドレスを使用することができます。

## このタスクについて

浮動 IP アドレスを作成または削除するには、**rdqmint** コマンドを使用します。浮動アドレスは、RDQM の 1 次ノードの指定した物理インターフェースにバインドします。RDQM がフェイルオーバーし、別のノードが 1 次ノードになると、浮動 IP は、新しい 1 次上の同じ名前のインターフェースにバインドされます。3 つのノード上の物理インターフェースは、浮動 IP アドレスと同じサブネットに属している必要があります。以下の図は、浮動 IP アドレスの使用を示しています。

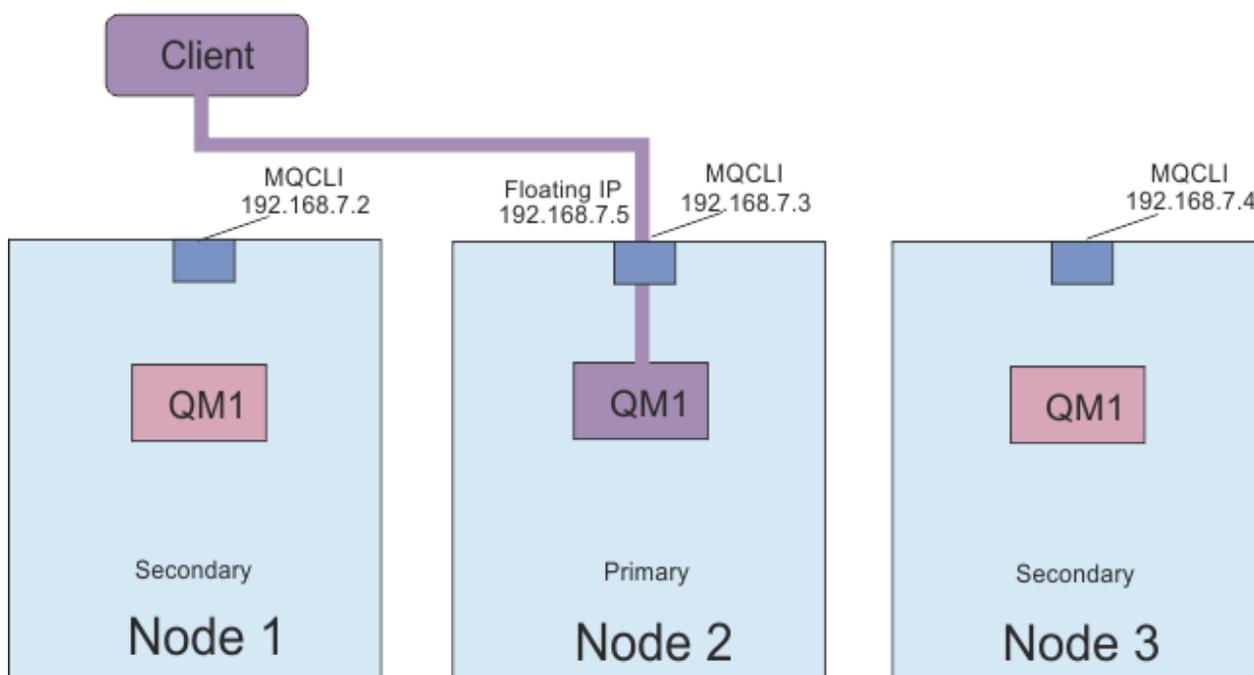
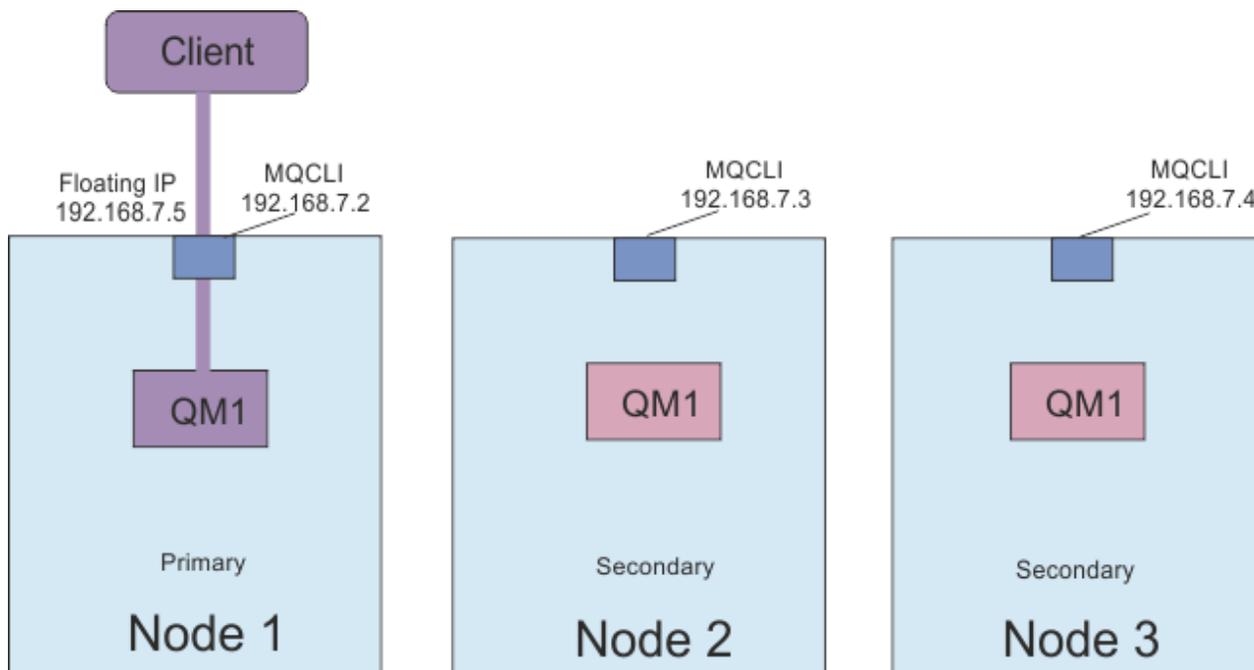


図 82. 浮動 IP アドレス

**rdqmint** コマンドを実行するには、mqm グループと haclient グループの両方のユーザーである必要があります。RDQM の 1 次ノードまたは 2 次ノードのいずれかで浮動 IP アドレスを作成または削除できます。

注：複数の RDQM に同じ浮動 IP アドレスを使用することはできません。それぞれの RDQM の浮動 IP アドレスは固有でなければなりません。

## 手順

- RDQM の浮動 IP アドレスを作成するには、以下のコマンドを入力します。

```
rdqmint -m qmname -a -f ipv4address -l interfacename
```

ここで、

#### **qmname**

浮動 IP アドレスを作成する RDQM の名前。

#### **ipv4address**

ipv4 形式の浮動 IP アドレス。

この浮動 IP アドレスは、どちらのアプライアンス (装置) にもまだ定義されていない有効な IPv4 アドレスでなければならず、またローカル・インターフェースに定義されている静的 IP アドレスと同じサブネットに属していなければなりません。

#### **interfacename**

1 次ノードがバインドする物理インターフェースの名前。

以下に例を示します。

```
rdqmint -m QM1 -a -f 192.168.7.5 -l MQCLI
```

- 既存の浮動 IP アドレスを削除するには、以下のコマンドを入力します。

```
rdqmint -m qmname -d
```

## 関連情報

[rdqmint \(RDQM の浮動 IP アドレスの追加または削除\)](#)

### Linux V 9.0.4 HA RDQM の開始、停止、および状態の表示

標準 IBM MQ 制御コマンドのバリエーションを使用して、複製データ・キュー・マネージャー (RDQM) を開始または停止したり、現在の状態を表示したりできます。

## このタスクについて

mqm グループと haclient グループの両方に属するユーザーとして、複製データ・キュー・マネージャー (RDQM) を開始または停止したり、現在の状態を表示したりするコマンドを実行する必要があります。

そのキュー・マネージャーの 1 次ノードでキュー・マネージャーを開始および停止するには、コマンドを実行する必要があります。

## 手順

- RDQM を開始するには、RDQM の 1 次ノードで以下のコマンドを入力します。

```
strmqm qmname
```

ここで、*qmname* は、開始する RDQM の名前です。

RDQM が開始され、Pacemaker が RDQM の管理を開始します。その他の *strmqm* オプションを指定する場合は、*strmqm* と共に *-ns* オプションを指定する必要があります。

- RDQM を停止するには、RDQM の 1 次ノードで以下のコマンドを入力します。

```
endmqm qmname
```

ここで、*qmname* は、停止する RDQM の名前です。

Pacemaker は、RDQM の管理を停止し、次に RDQM が終了します。他のすべての *endmqm* パラメーターは、RDQM を停止するときに使用できます。

- RDQM の状態を表示するには、以下のコマンドを入力します。

```
dspmq
```

出力される状態情報は、コマンドを RDQM の 1 次ノードまたは 2 次ノードのいずれかで実行するかによって異なります。1 次ノードで実行する場合は、**dspmq** によって返された正常状況メッセージの 1 つが表示されます。2 次ノードでコマンドを実行する場合は、状況 **running elsewhere** が表示されます。例えば、**dspmq** がノード RDQM7 で実行されている場合、以下の情報が返されることがあります。

```
QMNAME(RDQM8)          STATUS(Running elsewhere)
QMNAME(RDQM9)          STATUS(Running elsewhere)
QMNAME(RDQM7)          STATUS(Running)
```

1 次ノードが使用不可の場合、または **dspmq** が **root** ではないユーザーまたは **haclient** グループのメンバーによって実行された場合、**Unavailable** 状態が報告されます。以下に例を示します。

```
QMNAME(RDQM8)          STATUS(Unavailable)
QMNAME(RDQM9)          STATUS(Unavailable)
QMNAME(RDQM7)          STATUS(Unavailable)
```

コマンド **dspmq -o ha** (または **dspmq -o HA**) を入力して、ノードで認識されているキュー・マネージャーのリストやキュー・マネージャーが RDQM であるかどうかなどを表示できます。

```
dspmq -o ha

QMNAME(RDQM8)          HA(Replicated)
QMNAME(RDQM9)          HA(Replicated)
QMNAME(RDQM7)          HA(Replicated)
QMNAME(QM7)            HA()
```

## 関連情報

[dspmq \(キュー・マネージャーの表示\)](#)

[endmqm \(キュー・マネージャーの終了\)](#)

[strmqm \(キュー・マネージャーの始動\)](#)

## Linux V 9.0.4 RDQM および HA グループの状況の表示

HA グループおよび個々の複製データ・キュー・マネージャー (RDQM) の状況を表示できます。

## このタスクについて

個々の RDQM および HA グループ全体の状況を表示するには、**rdqmstatus** コマンドを使用します。

**rdqmstatus** コマンドを実行するには、mqm グループと **haclient** グループのユーザーである必要があります。このコマンドは、3 つのノードのいずれでも実行できます。

## 手順

- HA 構成の一部であるノードおよび RDQM の状況を表示するには、次のようにします。

```
rdqmstatus
```

コマンドを実行したノードの ID と HA 構成の RDQM の状況が、以下のように表示されます。

```
Node:                  mqhavam07.exampleco.com

Queue manager name:   RDQM8
Queue manager status: Running elsewhere
HA current location:  mqhavam08.exampleco.com

Queue manager name:   RDQM9
Queue manager status: Running elsewhere
HA current location:  mqhavam09.exampleco.com

Queue manager name:   RDQM7
Queue manager status: Running
HA current location:  This node
```

- HA グループの 3 つのノードの状況を表示するには、以下のコマンドを入力します。

```
rdqmstatus -n
```

各ノードのオンラインまたはオフライン状況が報告されます。以下に例を示します。

```
Node mqha04(mqhavm04.example.com) is online
Node mqha05(mqhavm05.example.com) is offline
Node mqha06(mqhavm06.example.com) is online
```

- HA グループのすべてのノードの特定のキュー・マネージャーの状況を表示するには、以下のコマンドを入力します。

```
rdqmstatus -m qmname
```

ここで、*qmname* は、状況を表示する RDQM の名前です。現行ノードの RDQM の状況が表示され、その後現行ノードの観点からの他の 2 つのノードの状況の要約が表示されます。

以下の表に、RDQM に対する **rdqmstatus** コマンドによって返される可能性がある現行ノードに関する情報の要約を示します。

表 30. 現行ノードの状況		
状況属性	可能な値	表示される時期
ノード名	<i>nodename</i>	常に表示されます。
キュー・マネージャー状況	実行中 別の場所で実行中 Ended 利用不可	常に表示されます。
CPU	<i>n.nn%</i>	現行ノードに 1 次の役割がある場合にのみ表示されます (つまり、RDQM がそのノードで実行されています)。
メモリー	<i>nnn</i> MB used、 <i>y.y</i> GB allocated	現行ノードに 1 次の役割がある場合にのみ表示されます (つまり、RDQM がそのノードで実行されています)。
キュー・マネージャー・ファイル・システム	<i>nnn</i> MB used、 <i>y.y</i> GB allocated [ <i>z%</i> ]	現行ノードに 1 次の役割がある場合にのみ表示されます (つまり、RDQM がそのノードで実行されています)。
HA 役割	Primary Secondary Unknown	常に表示されます。
HA 状況	すべてのノードがスタンバイ状態です このノードはスタンバイ状態です Remote nodes in standby 混合  リモート・ノードの状況	すべてのノードがスタンバイ状態です 現行ノードがスタンバイ状態の場合 両方のリモート・ノードがスタンバイ状態の場合 リモート・ノードごとに状況が異なる場合 (個々の状況については次の表を参照)  両方のリモート・ノードの状況が同じ場合 (すべての値については次の表を参照)
HA 制御	有効 無効 不明	常に表示されます。RDQM が Pacemaker 制御下にあるかどうかを示されます。

状況属性	可能な値	表示される時期
HA 優先ロケーション	なし このノード 不明 <i>nodename</i>	常に表示されます。
HA 浮動 IP インターフェース	<i>Interface_name</i>	常に表示されます。
HA 浮動 IP アドレス	<i>IPV4_address</i>	常に表示されます。

以下の表に、HA グループのその他のノードに対する **rdqmstatus** コマンドによって返される情報の要約を示します。

状況属性	可能な値	表示される時期
ノード名	<i>nodename</i>	常に表示されます。
HA 状況	通常 同期が進行中 リモートは使用不可です 不整合 休止 リモート・ノードはスタンバイ状態 です 不明	ノードが互いに同期している場合 リモート・ノードと同期している場合 リモート・ノードと通信できない場合 リモート・ノードとの同期化が行われておらず、同期していない場合 複製が一時停止している場合 リモート・ノードはスタンバイ状態 です
HA 同期が進行中	<i>n.n%</i>	同期化が進行中であり、 <b>root</b> としてコマンドを実行した場合に表示されます。
HA 予想される同期時間	<i>yyyy-mm-dd hh:mm:ss.nnn</i>	同期化が進行中の場合に表示されません。
HA 非同期データ	<i>nKB</i>	リモート・ノードが使用できないか、または不整合である場合に表示されます。

## 例

### 1 次ノードの正常状況の例:

```

Node:                               mqhavam07.exampleco.com
Queue manager status:              Running
CPU:                               0.00
Memory:                             123MB
Queue manager file system:         606MB used, 1.0GB allocated [60%]
HA role:                            Primary
HA status:                          Normal
HA control:                         Enabled
HA current location:               This node
HA preferred location:              This node
HA floating IP interface:           Eth4
HA floating IP address:             192.0.2.4

```

```

Node:                               mqhavam08.exampleco.com
HA status:                          Normal

```

```
Node: mqhavam09.exampleco.com
HA status: Normal
```

2 次ノードの正常状況の例:

```
Node: mqhavam08.exampleco.com
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: mqhavam07.exampleco.com
HA preferred location: mqhavam07.exampleco.com
HA floating IP interface: Eth4
HA floating IP address: 192.0.2.4
```

```
Node: mqhavam07.exampleco.com
HA status: Normal
```

```
Node: mqhavam09.exampleco.com
HA status: Normal
```

同期化が進行中の場合の 1 次ノードの状況の例:

```
Node: mqhavam07.exampleco.com
Queue manager status: Running
CPU: 0.53
Memory: 124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Synchronization in progress
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: Eth4
HA floating IP address: 192.0.2.4
```

```
Node: mqhavam08.exampleco.com
HA status: Synchronization in progress
HA synchronization progress: 11.0%
HA estimated time to completion: 2017-09-06 14:55:05
```

```
Node: mqhavam09.exampleco.com
HA status: Synchronization in progress
HA synchronization progress: 11.0%
HA estimated time to completion: 2017-09-06 14:55:06
```

複数の状態を示す 1 次ノードの例:

```
Node: mqhavam07.exampleco.com
Queue manager status: Running
CPU: 0.02
Memory: 124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Mixed
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: Eth4
HA floating IP address: 192.0.2.4
```

```
Node: mqhavam08.exampleco.com
HA status: Normal
```

```
Node: mqhavam09.exampleco.com
HA status: Inconsistent
```

## 関連情報

 [rdqmstatus](#)

  **障害が発生したノードの置き換え**

HA グループ内のノードの 1 つに障害が発生した場合は、それを置き換えることができます。

## このタスクについて

ノードを置き換えるためのステップは、シナリオによって異なります。

- 障害が発生したノードを同じ構成のノードで置き換える場合は、HA グループを中断せずにノードを置き換えることができます。
- 新しいノードの構成が異なる場合は、HA グループを削除して再構築する必要があります。

## 手順

- 障害が発生したノードのように交換ノードが構成されている場合 (同じホスト名、同じ IP アドレスなど) は、新規ノードで以下のステップを実行します。
  - a) その他のノードのファイルと一致する `rdqm.ini` ファイルを作成し、`rdqmadm -c` コマンドを実行します (503 ページの『[Pacemaker クラスタ \(HA グループ\) の定義](#)』を参照)。
  - b) `crtmqm -sxs qmanager` コマンドを実行して、各複製データ・キュー・マネージャーを再作成します (505 ページの『[HA RDQM の作成](#)』を参照)。
- 交換ノードの構成が、障害が発生したノードとは異なる場合:
  - a) `dltmqm` コマンドを使用して、HA グループ内の他のノードから複製データ・キュー・マネージャーを削除します (506 ページの『[HA RDQM の削除](#)』を参照)。
  - b) `rdqmadm -u` コマンドを使用して、Pacemaker クラスタを構成解除します (505 ページの『[Pacemaker クラスタ \(HA グループ\) の削除](#)』を参照)。
  - c) `rdqmadm -c` コマンドを使用して、新規ノードの情報を含む Pacemaker クラスタを再構成します (503 ページの『[Pacemaker クラスタ \(HA グループ\) の定義](#)』を参照)。
  - d) `crtmqm -sxs qmanager` コマンドを実行して、各複製データ・キュー・マネージャーを再作成します (505 ページの『[HA RDQM の作成](#)』を参照)。

### V 9.0.4 スプリット・ブレイク状態の解決

HA グループ内で特定の障害シーケンスが発生すると、スプリット・ブレイク状態が報告される場合があります。

例えば、3つのノードがすべて接続を失ったとします。1次ノードが接続を再確立する前に2つの2次ノードが両方とも接続を再確立すると、その2つの2次ノードが新しいクォーラムを形成し、そのどちらかがキュー・マネージャーを実行することになります。その状態で元の1次ノードが接続を再確立すると、スプリット・ブレイク状態が報告される可能性があります。

この状態で、`rdqmstatus -m QMname` を元の1次ノードで実行すると、HA 状況が `Inconsistent` と表示されます。

```
Node: node1
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Inconsistent
HA control: Enabled
HA current location: hanode2
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None

Node: node2
HA status: Inconsistent
HA out of sync data: 8KB

Node: node3
HA status: Inconsistent
HA out of sync data: 8KB
```

この場合は、新しいクォーラムを形成した元の2次ノードの両方でデータを保持する必要があります。以下のステップを完了します。

1. 元の1次ノードで、`root` として以下のコマンドを実行します。

```
drbdadm connect --discard-my-data QMname
```

2. それぞれの 2 次ノードで、root として以下のコマンドを実行します。

```
drbdadm connect QMname:first-node-name
```

**drbdadm** は、DRBD で提供されるコマンドです。これは、`/usr/sbin/drbdadm` 内に `drbd-utils` パッケージの一部としてインストールされます。

## Linux V 9.0.5 RDQM 災害復旧

Linux プラットフォームのサブセットでは、RDQM (複製データ・キュー・マネージャー) を利用して、災害復旧ソリューションを構築できます。

詳細については、[Software Product Compatibility Reports](#) を参照してください。

1 つのサーバーで災害復旧キュー・マネージャーの 1 次インスタンスを作成し、リカバリー・ノードとして実行する別のサーバーでキュー・マネージャーの 2 次インスタンスを作成できます。キュー・マネージャーのインスタンス間でデータを複製します。1 次キュー・マネージャーを失った場合は、手動で 2 次インスタンスを 1 次インスタンスにしてキュー・マネージャーを開始し、同じ場所から処理を再開できます。2 次の役割になっているキュー・マネージャーを開始することはできません。2 つのノード間でのデータの複製は DRBD によって処理されます。

1 次キュー・マネージャーと 2 次キュー・マネージャーの間のデータ複製では、同期複製と非同期複製のいずれかを選択できます。非同期オプションを選択すると、IBM MQ の PUT や GET などの操作が完了してアプリケーションに戻ってから、そのイベントが 2 次キュー・マネージャーに複製されます。非同期で複製されるので、リカバリー状態となった後に、一部のメッセージング・データが失われる可能性があります。しかし、2 次キュー・マネージャーは一貫性のある状態を保ち、メッセージ・ストリームの少し前の部分で開始する場合でも、即時に実行を開始することができます。

既存のキュー・マネージャーに災害復旧構成を追加することはできません。また、1 つのキュー・マネージャーで RDQM 災害復旧と RDQM 高可用性の両方を構成することもできません。

さまざまなサーバーで実行する RDQM キュー・マネージャーについては、いろいろな組み合わせが可能です。例えば、6 つの 1 次 DR キュー・マネージャーを同じノードで実行し、2 次 DR キュー・マネージャーを 6 つのデータ・センターの 6 つのノードで構成する、といったことが可能です。その逆に、1 次災害復旧キュー・マネージャーを別々のノードで実行し、すべての 2 次災害復旧キュー・マネージャーを同じノードで実行することもできます。いくつかの構成例を以下の図に示します。

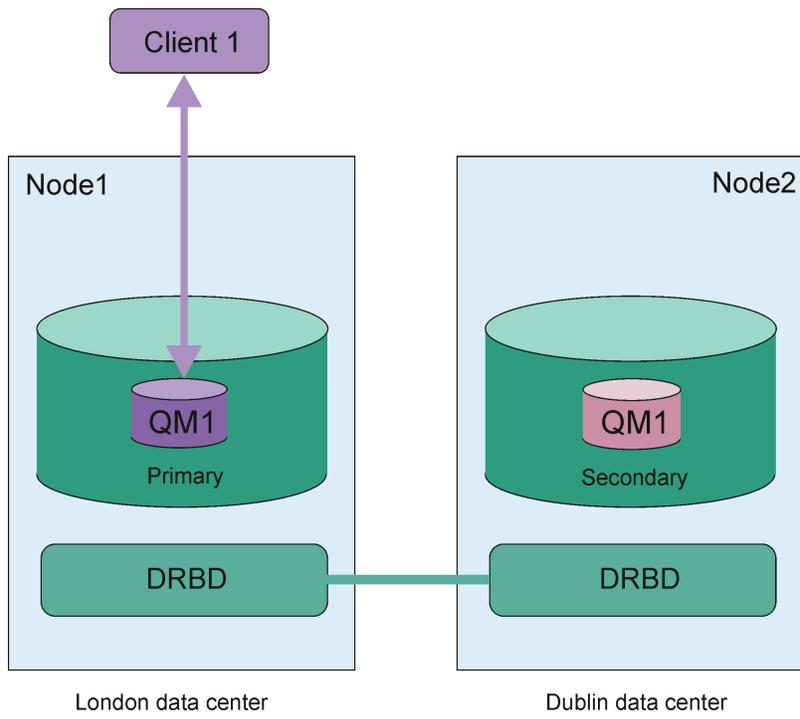


図 83. 1 つの RDQM ペア

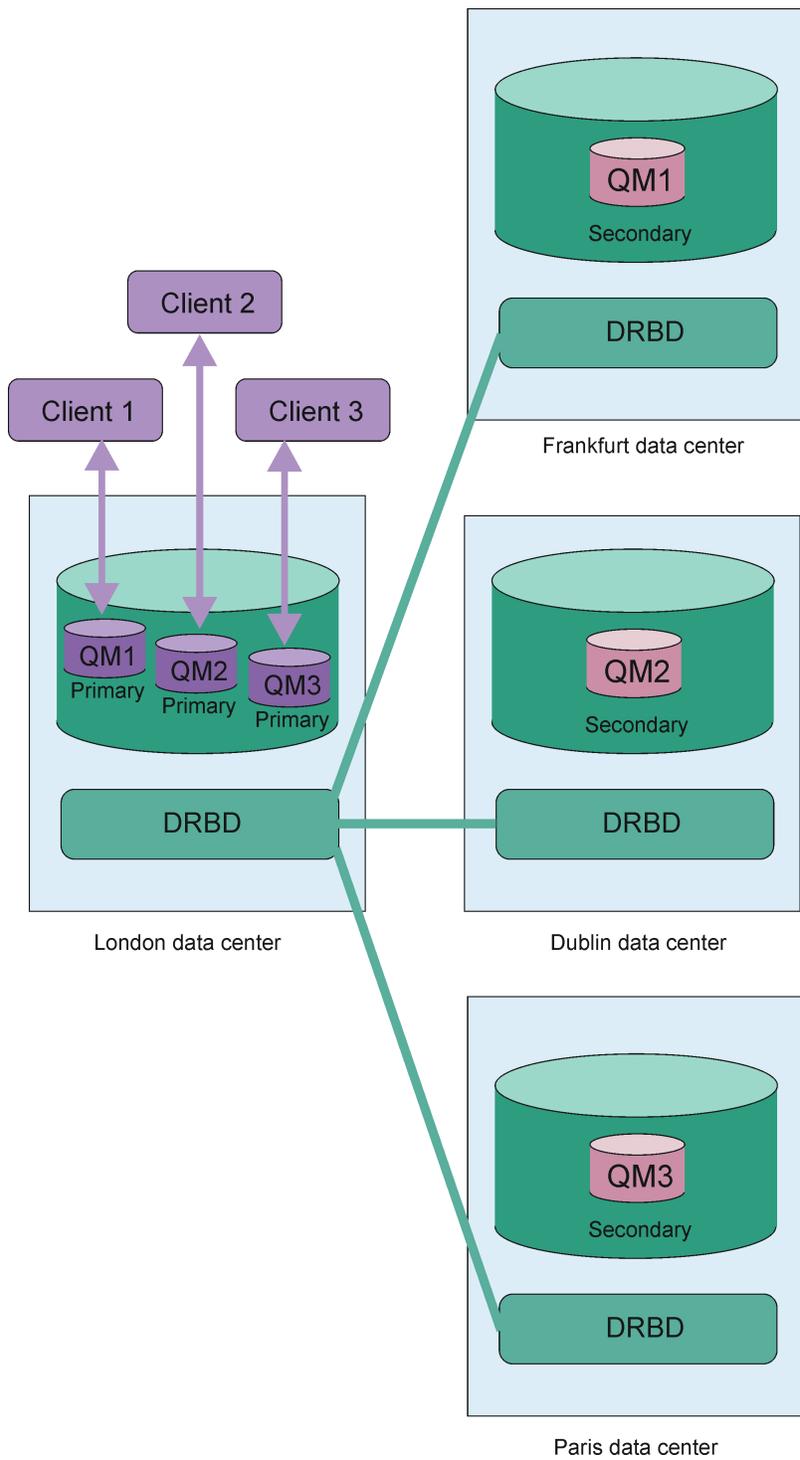


図 84. 1 次キュー・マネージャーを同じノードで構成する例

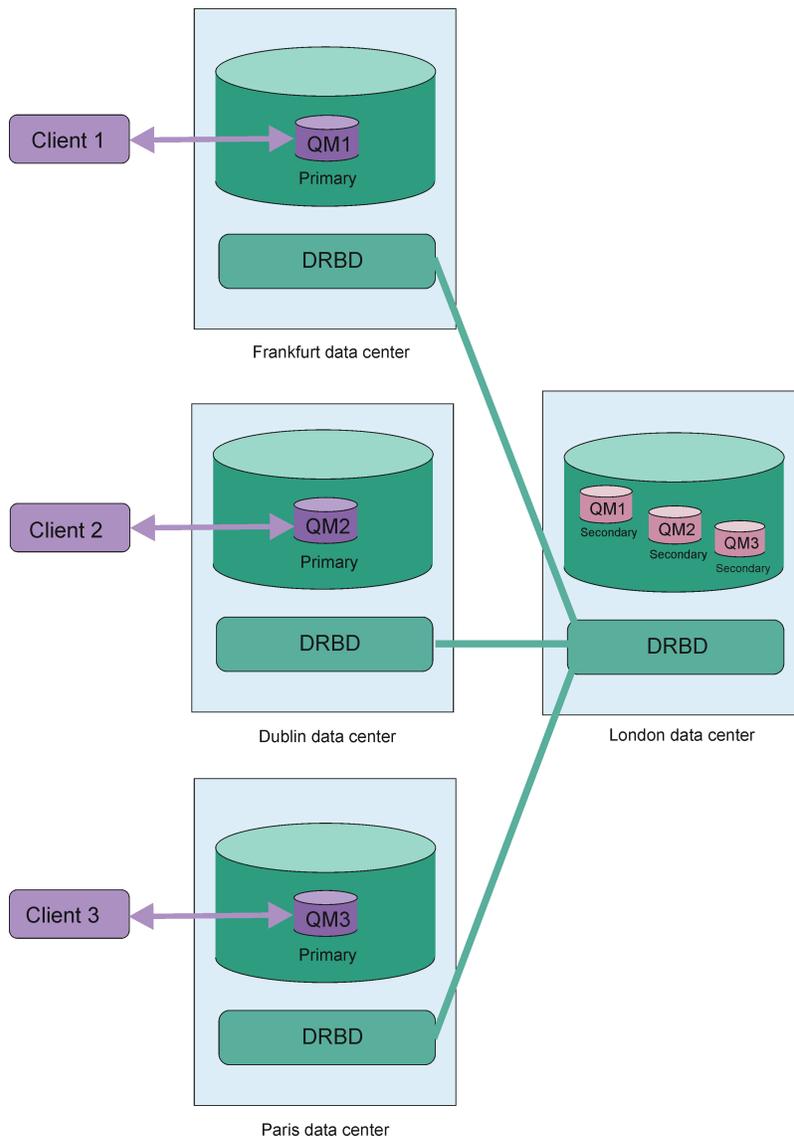


図 85. 2 次キュー・マネージャーを同じノードで構成する例

## 複製、同期、スナップショット

災害復旧構成にした 2 つのノードを接続すると、災害復旧キュー・マネージャーの永続データに対するすべての更新がキュー・マネージャーの 1 次インスタンスから 2 次インスタンスに転送されます。これを複製といいます。

ノード間のネットワーク接続が失われると、キュー・マネージャーの 1 次インスタンスの永続データに対する変更が追跡されます。ネットワーク接続が復旧すると、2 次インスタンスの内容の取り込みができるだけ早く追いつくようにするために、別のプロセスが使用されます。これを同期といいます。

同期が進行中のとき、2 次インスタンス内のデータは不整合の状態になっています。2 次キュー・マネージャー・データの状態のスナップショットが取得されます。同期中にメイン・ノードまたはネットワーク接続で障害が発生すると、2 次インスタンスがそのスナップショットに戻って、キュー・マネージャーを開始できるようになります。ただし、元のネットワーク障害の発生後に行われたすべての更新情報は失われます。

### Linux V 9.0.5 RDQM DR ソリューションの要件

RDQM 災害復旧 (DR) キュー・マネージャー・ペアを構成する前に、いくつかの要件を満たさなければなりません。

## システム要件

RDQM DR を構成する前に、RDQM DR キュー・マネージャーをホストする各サーバーでいくつかの構成を完了する必要があります。

- 各ノードには、drbdpool という名前のボリューム・グループが必要です。各災害復旧複製データ・キュー・マネージャー (DR RDQM) のストレージは、このボリューム・グループのキュー・マネージャーごとに2つの別個の論理ボリュームとして割り振られます。(それぞれのキュー・マネージャーは、スナップショットへの復帰操作をサポートするために2つの論理ボリュームを必要とするので、各 DR RDQM に、作成時の指定値の2倍のストレージが割り振られます。) 最高のパフォーマンスを得るためには、このボリューム・グループを内部ディスク・ドライブ (SSD) に対応する1つ以上の物理ボリュームで構成する必要があります。
- 各ノードにデータ複製用のインターフェースが必要です。すべての複製データ・キュー・マネージャーの予期されるワークロードを前提として、複製要件をサポートするために十分な帯域幅が必要です。  
フォールト・トレランスを最大にするために、このインターフェースとして、独立したネットワーク・インターフェース・カード (NIC) を使用してください。
- RFC 1123 で修正された RFC 952 で定義されているように、DRBD では、有効なインターネット・ホスト名 (uname -n によって返される値) が RDQM 用の各ノードで必要になります。
- DR RDQM 用のノード同士の間にはファイアウォールがある場合は、そのファイアウォールで、ノード間のトラフィックを複製用のポートで許可する必要があります。
- SELinux を permissive 以外のモードで使用しているシステムでは、以下のコマンドを実行する必要があります。

```
semanage permissive -a drbd_t
```

## ネットワーク要件

災害復旧用のノードをさまざまなデータ・センターに配置することをお勧めします。

次の制約に注意してください。

- データ・センター間の待ち時間が長くなると、パフォーマンスが急速に低下します。IBM では、同期複製で最大5ミリ秒、非同期複製で最大50ミリ秒の待ち時間をサポートしています。
- 複製リンクで送信するデータは、IBM MQ AMS での暗号化は別として、それ以外の暗号化の対象になりません。
- RDQM キュー・マネージャーを災害復旧用に構成すると、2つの RDQM ノード間のデータ複製の要件によるオーバーヘッドが発生します。同期複製の方が非同期複製よりもオーバーヘッドが大きくなります。同期複製では、データが両方のノードに書き込まれるまで、ディスク入出力操作がブロックされます。非同期複製では、データを1次ノードに書き込むだけで処理の継続が可能になります。

## キュー・マネージャーで作業するためのユーザー要件

複製データ・キュー・マネージャー (RDQM) の作成/削除/構成の作業は、root ユーザーとして実行するか、以下のコマンドに対する sudo 権限を持った mqm グループに属するユーザー ID で実行する必要があります。

- **crtmqm**
- **dltmqm**
- **rdqmdr**

mqm グループに属するユーザーは、以下のコマンドを使用して DR RDQM の状態と状況を表示できます。

- **dspmq**
- **rdqmstatus**

**crtmqm** コマンドを使用して、災害復旧構成の 1 次キュー・マネージャーまたは 2 次キュー・マネージャーとして機能する複製データ・キュー・マネージャー (RDQM) を作成します。

## このタスクについて

ユーザーが **sudo** を使用できる場合は、**mqm** グループのユーザーとして複製データ・キュー・マネージャー (RDQM) を作成できます。そうでない場合は、RDQM を **root** として作成する必要があります。

1 つのノードで 1 次 RDQM DR キュー・マネージャーを作成する必要があります。次に、別のノードで同じキュー・マネージャーの 2 次インスタンスを作成する必要があります。1 次インスタンスと 2 次インスタンスの名前は同じでなければならず、同じ量のストレージを割り振る必要があります。

## 手順

- 1 次 DR RDQM を作成するには、以下のようになります。

- a) 次のコマンドを入力します。

```
crtmqm -rr p [-rt (a | s)] -rl Local_IP -ri Recovery_IP -rn Recovery_Name -rp Port  
[other_crtmqm_options] [-fs size] QMname
```

ここで、

### **-rr p**

キュー・マネージャーの 1 次インスタンスを作成することを指定します。

### **-rt a | s**

**-rt s** では、DR 構成で同期複製を使用することを指定し、**-rt a** では、DR 構成で非同期複製を使用することを指定します。非同期複製がデフォルトです。

### **-rl Local\_IP**

このキュー・マネージャーの DR 複製で使用するローカル IP アドレスを指定します。

### **-ri Recovery\_IP**

キュー・マネージャーの 2 次インスタンスをホスティングするサーバーでの複製に使用するインターフェースの IP アドレスを指定します。

### **-rn Recovery\_Name**

キュー・マネージャーの 2 次インスタンスをホスティングするシステムの名前を指定します。この名前は、そのサーバーで **uname -n** を実行した時に返される値です。そのサーバーで 2 次キュー・マネージャーを明示的に作成する必要があります。

### **-rp Port**

DR レプリケーションに使用するポートを指定します。

### **other\_crtmqm\_options**

(オプション) 以下の一般的な **crtmqm** オプションを 1 つ以上指定することもできます。

- -z
- -q
- -c Text
- -d DefaultTransmissionQueue
- -h MaxHandles
- -g ApplicationGroup
- -oa user|group
- -t TrigInt
- -u DeadQ
- -x MaxUMsgs
- -lp LogPri

- -ls LogSec
- -lc | -l
- -lla | -lln
- -lf LogFileSize
- -p Port

#### **-fs size**

(オプション) キュー・マネージャー用に作成するファイル・システムのサイズ、つまり drbdpool ボリューム・グループ内で作成する論理ボリュームのサイズを指定します。スナップショットへの復帰操作をサポートするために、そのサイズの別の論理ボリュームも作成するので、DR RDQM の合計ストレージは、ここで指定した値の 2 倍になります。

#### **QMname**

複製データ・キュー・マネージャーの名前を指定します。この名前には大/小文字の区別があります。

コマンドが完了すると、キュー・マネージャーの 2 次インスタンスを作成するために 2 次ノードで入力する必要のあるコマンドが出力されます。1 次ノードで **rdqmdr** コマンドを使用して、2 次キュー・マネージャーを作成するために 2 次ノードで実行する必要がある **crtmqm** コマンドを取得することもできます。522 ページの『DR RDQM の 1 次特性と 2 次特性の管理』を参照してください。

- 2 次 DR RDQM を作成するには、以下のようになります。

- a) RDQM の 2 次インスタンスをホストするノードで、以下のコマンドを入力します。

```
crtmqm -rr s [-rt (a | s)] -rl Local_IP -ri Primary_IP -rn Primary_Name -rp Port
[other_crtmqm_options] [-fs size] QMname
```

#### 説明

##### **-rr s**

キュー・マネージャーの 2 次インスタンスを作成することを指定します。

##### **-rt a | s**

**-rt s** は、DR 構成が同期複製を使用することを指定し、**-rt a** は DR 構成が非同期複製を使用することを指定します。

##### **-rl Local\_IP**

このキュー・マネージャーの DR 複製で使用するローカル IP アドレスを指定します。

##### **-ri Primary\_IP**

キュー・マネージャーの 1 次インスタンスをホスティングするサーバーでの複製に使用するインターフェースの IP アドレスを指定します。

##### **-rn Primary\_Name**

キュー・マネージャーの 1 次インスタンスをホスティングするシステムの名前を指定します。この名前は、そのサーバーで `uname -n` を実行した時に返される値です。

##### **-rp Port**

DR レプリケーションに使用するポートを指定します。

##### **other\_crtmqm\_options**

(オプション) 以下の一般的な **crtmqm** オプションを 1 つ以上指定することもできます。

- -z

#### **-fs size**

キュー・マネージャー用に作成するファイル・システムのサイズ、つまり drbdpool ボリューム・グループ内で作成する論理ボリュームのサイズを指定します。1 次キュー・マネージャーの作成時にデフォルト以外のサイズを指定した場合は、ここで同じ値を指定する必要があります。

#### **QMname**

複製データ・キュー・マネージャーの名前を指定します。キュー・マネージャーの 1 次インスタンスで指定した名前と同じ名前ではなければなりません。名前には大/小文字の区別があります。

## 次のタスク

キュー・マネージャーの1次インスタンスと2次インスタンスを作成したら、両方のノードの状況を調べて、両方が正しいことを確認する必要があります。両方のノードで **rdqmstatus** コマンドを使用します。どちらのノードでも正常状況が表示されるはずです (524 ページの『DR RDQM 状況の表示』を参照)。その状況が表示されない場合は、2次インスタンスをいったん削除して再作成してください。その時に、正しい引数を使用するように注意してください。

### 関連情報

[crtmqm](#)

Linux V 9.0.5 DR RDQM の削除

**dltmqm** コマンドを使用して、災害復旧複製データ・キュー・マネージャー (RDQM) を削除します。

## このタスクについて

RDQM を削除するために、RDQM の1次ノードと2次ノードの両方でこのコマンドを実行する必要があります。最初に RDQM を終了する必要があります。mqm ユーザーが必要な sudo 特権を持っている場合は、mqm ユーザーとしてコマンドを実行できます。その他の場合は、root としてコマンドを実行する必要があります。

### 手順

- DR RDQM を削除するには、以下のコマンドを入力します。

```
dltmqm RDQM_name
```

### 関連情報

[dltmqm](#)

Linux V 9.0.5 DR RDQM の1次特性と2次特性の管理

2次災害復旧複製データ・キュー・マネージャー (DR RDQM) を1次 DR RDQM に変更できます。1次インスタンスを2次インスタンスに変更することもできます。

## このタスクについて

**rdqmdr** コマンドを使用して、RDQM の2次インスタンスを1次インスタンスに変更できます。何かの理由で1次インスタンスが失われた時に、このアクションを実行しなければならない場合があります。その後、キュー・マネージャーを開始し、リカバリー・ノードで実行できます。

**rdqmdr** コマンドを使用して、RDQM の1次インスタンスを2次インスタンスに変更することもできます。システムを再構成する時などに、このアクションを実行しなければならない場合があります。

1次キュー・マネージャーで **rdqmdr** を使用して、リカバリー・ノードでそのキュー・マネージャーの2次インスタンスを作成するのに必要な正確なコマンドを取得することもできます。

mqm グループのユーザーが **sudo** を使用できるのであれば、そのユーザーとして **rdqmdr** コマンドを使用できます。そうでなければ、root としてログインする必要があります。

### 手順

- DR RDQM の2次インスタンスを1次インスタンスに変更するには、以下のコマンドを入力します。

```
rdqmdr -m QMname -p
```

キュー・マネージャーの1次インスタンスが実行を続けていて、DR 複製リンクがまだ機能している場合、このコマンドは失敗します。

- キュー・マネージャーの1次インスタンスを2次インスタンスに変更するには、以下のコマンドを入力します。

```
rdqmdr -m QMname -s
```

- キュー・マネージャーの2次インスタンスを構成するのに必要な **crtmqm** コマンドを表示するには、1次ノードで以下のコマンドを入力します。

```
rdqmdr -d -m QMname
```

返された **crtmqm** コマンドを2次ノードで入力すれば、RD RDQM の2次インスタンスを作成できます。

## Linux V9.0.5 DR RDQM の開始、停止、および状態の表示

IBM MQ の標準的な制御コマンドのバリエーションを使用して、災害復旧複製データ・キュー・マネージャー (DR RDQM) を開始/停止したり、現在の状態を表示したりできます。

### このタスクについて

複製データ・キュー・マネージャー (RDQM) を開始/停止したり、現在の状態を表示したりするコマンドは、mqm グループに属するユーザーとして実行する必要があります。

そのキュー・マネージャー (つまり、キュー・マネージャーが現在実行されているノード) の1次ノードでキュー・マネージャーを開始および停止するには、コマンドを実行する必要があります。

### 手順

- DR RDQM を開始するには、RDQM の1次ノードで以下のコマンドを入力します。

```
stmqm qmname
```

ここで、*qmname* は、開始する RDQM の名前です。

- RDQM を停止するには、RDQM の1次ノードで以下のコマンドを入力します。

```
endmqm qmname
```

ここで、*qmname* は、停止する RDQM の名前です。

- RDQM の状態を表示するには、以下のコマンドを入力します。

```
dspm q -m QMname
```

出力される状態情報は、コマンドを RDQM の1次ノードまたは2次ノードのいずれかで実行するかによって異なります。1次ノードで実行する場合は、**dspm q** によって返された正常状況メッセージの1つが表示されます。このコマンドを2次ノードで実行すると、状況 **Ended immediately** が表示されます。例えば、**dspm q** がノード RDQM7 で実行されている場合、以下の情報が返されることがあります。

```
QMNAME(DRQM8)          STATUS(Ended immediately)
QMNAME(DRQM7)          STATUS(Running)
```

**dspm q** の引数を使用して、RDQM が災害復旧用として構成されているかどうか、現時点で1次インスタンスになっているか2次インスタンスになっているかを確認できます。

```
dspm q -m QMname -o (dr | DR)
```

以下のいずれかの応答が表示されます。

#### **DRROLE()**

キュー・マネージャーが災害復旧用に構成されていないことを示しています。

#### **DRROLE(Primary)**

キュー・マネージャーが DR 1次として構成されていることを示しています。

#### **DRROLE(Secondary)**

キュー・マネージャーが DR 2次として構成されていることを示しています。

## 関連情報

[dspmq](#)

[endmqm](#)

[strmqm](#)

## Linux V 9.0.4 DR RDQM 状況の表示

1つのノードにあるすべての災害復旧複製データ・キュー・マネージャー (DR RDQM) の状況や、指定した DR RDQM の詳細情報を表示できます。

## このタスクについて

**rdqmstatus** コマンドで、すべての DR RDQM の状況や個々の RDQM の状況を表示できます。

**rdqmstatus** コマンドを実行するには、mqm グループのユーザーである必要があります。このコマンドは、DR RDQM ペアのどちらのノードでも実行できます。

## 手順

- 1つのノードにあるすべての DR RDQM の状況を表示するには、そのノードで以下のコマンドを実行します。

```
rdqmstatus
```

そのノードにある DR RDQM の状況が表示されます。以下に例を示します。

```
Queue manager name:      DRQM8
Queue manager status:    Ended immediately
DR role:                  Secondary

Queue manager name:      DRQM7
Queue manager status:    Running
DR role:                  Primary
```

- 1つの RDQM の状況を表示するには、以下のコマンドを入力します。

```
rdqmstatus -m qmname
```

以下の表に、返される情報の要約を示します。

状況属性	可能な値	表示される時期
キュー・マネージャー状況	状態 (dspmq で表示される値)	常に表示されます。
CPU	<i>n.nn%</i>	現行ノードの RDQM に 1 次の役割がある場合に限り表示されます
メモリー	<i>nnnMB</i>	現行ノードの RDQM に 1 次の役割がある場合に限り表示されます
キュー・マネージャー・ファイル・システム	<i>nnnMB used, n.nGB allocated [n%]</i>	現行ノードの RDQM に 1 次の役割がある場合に限り表示されます
DR 役割	1 次 2 次 不明	常に表示されます。
DR 状況	通常	通常のコマンド

表 32. Status attributes (続き)		
状況属性	可能な値	表示される時期
	同期が進行中	同期が進行中です
	パーティション	DR 複製ネットワークが使用不可の状態になっている時に、両方のノードでキュー・マネージャーが開始されました。
	リモート・システムが使用不可	もう一方のノードへの接続が失われました
	不整合	同期が進行中でしたが、中断されました
	スナップショットへの復帰	キュー・マネージャーが不整合状態になった時に取られたスナップショットに復帰することをユーザーが選択しました。
	リモート・システムが未構成	RDQM の 1 次インスタンスは構成されていますが、2 次インスタンスが構成されていません
	ネゴシエーションの失敗	一方のノードで同期複製が設定され、もう一方のノードで非同期複製が設定されています
DR タイプ	同期または非同期	常に表示されます。
DR ポート	<i>port_number</i> (このキュー・マネージャーのデータを複製するための TCP/IP ポート)	常に表示されます。
DR ローカル IP アドレス	このキュー・マネージャーが DR の複製を実行している複製元のローカル IP アドレス	常に表示されます。
DR リモート IP アドレス	このキュー・マネージャーが DR の複製を実行している複製先のリモート IP アドレス	常に表示されます。
DR 非同期データ	nKB	リモート・ノードが使用できないか、または不整合である場合に表示されます。
DR 同期の進行状況	n%	同期の進行中に表示されます
DR 完了までの推定時間	YYYY-MM-DD HH:MM:SS	同期の進行中に表示されます
スナップショットへの復帰の進行中	n%	DR 状況が Reverting to snapshot の場合に表示されます。状況がカウントダウンで表示されるので、0% になったら完了です。

## 例

1 次ノードの正常状況の例:

```
Queue manager status: Running
CPU: 0.00
Memory: 123MB
```

```
Queue manager file system:      51MB used, 1.0GB allocated [5%]
DR role:                        Primary
DR status:                      Normal
DR type:                        Synchronous
DR port:                        3000
DR local IP address:            192.168.20.1
DR remote IP address:          192.168.20.2
```

2 次ノードの正常状況の例:

```
Queue manager status:          Ended immediately
DR role:                       Secondary
DR status:                      Normal
DR port:                        3000
DR local IP address:            192.168.20.2
DR remote IP address:          192.168.20.1
```

同期化が進行中の場合の 1 次ノードの状況の例:

```
Queue manager status:          Running
CPU:                            0.53
Memory:                         124MB
Queue manager file system:      51MB used, 1.0GB allocated [5%]
DR role:                        Primary
DR status:                      Synchronization in progress
DR type:                        Synchronous
DR port:                        3000
DR local IP address:            192.168.20.1
DR remote IP address:          192.168.20.2
DR synchronization progress:    11.0%
DR estimated time to completion: 2017-09-06 14:55:05
```

1 次ノードがパーティション化されていることを示す例:

```
Queue manager status:          Running
CPU:                            0.02
Memory:                         124MB
Queue manager file system:      51MB used, 1.0GB allocated [5%]
DR role:                        Primary
DR status:                      Partitioned
DR type:                        Synchronous
DR port:                        3000
DR local IP address:            192.168.20.1
DR remote IP address:          192.168.20.2
```

## 関連情報

Linux [rdqmstatus](#)

Linux [V 9.0.5](#) **災害復旧環境での操作**

災害復旧構成の 2 次キュー・マネージャーに切り替えたい状況がいくつかあります。

### 災害時回復の場合

メイン・サイトで 1 次キュー・マネージャーが完全に失われた場合は、リカバリー・サイトで 2 次キュー・マネージャーを開始します。アプリケーションは、リカバリー・サイトのキュー・マネージャーに再接続し、その 2 次キュー・マネージャーがアプリケーション・メッセージを処理します。以前の構成に戻すための手順は、失敗の原因によって異なります。例えば、メイン・ノードが完全に失われた場合と一時的に失われた場合とでは、手順が異なります。

メイン・サイトが一時的に失われた後に実行する手順については、[527 ページの『リカバリー・ノードへの切り替え』](#)を参照してください。永久的な障害が発生した場合に実行する手順については、[527 ページの『災害復旧構成の障害ノードの交換』](#)を参照してください。

### 災害復旧テストのサポート

災害復旧構成をテストするには、一時的に 2 次インスタンスに切り替えて、アプリケーションが正常に接続できるかどうかを確認します。1 次ノードで一時的な障害が発生した場合の切り替えと同じ手順を実行します。[527 ページの『リカバリー・ノードへの切り替え』](#)を参照してください。

## スナップショットへの復帰

同期の進行中に 1 次ノードで障害が発生した場合は、同期の開始直前の 2 次キュー・マネージャー・データのスナップショットに復帰できます。2 次サーバーを整合状態に復元できたら、1 次として実行します。527 ページの『リカバリー・ノードへの切り替え』の説明に沿って 2 次を 1 次にすれば、スナップショットに戻ります。キュー・マネージャーを開始する前に、スナップショットへの復帰が完了したことを (`rdqmstatus` コマンドを使用して) 確認する必要があります。

Linux V 9.0.5 リカバリー・ノードへの切り替え

メイン・サイトで障害が発生した場合、リカバリー・サイトへの切り替え手順を実行します。

## このタスクについて

メイン・サイトで 1 次キュー・マネージャーが失われた場合は、リカバリー・サイトの 2 次キュー・マネージャーを 1 次にして開始します。アプリケーションは、リカバリー・サイトのキュー・マネージャーに再接続し、そのキュー・マネージャーがアプリケーション・メッセージを処理します。この手順を使用して、リカバリー・ノードをテストすることもできます。

root としてログインするか、mqm グループに属していて必要な sudo 構成を持っているユーザーとしてログインする必要があります。

## 手順

1. この手順を使用して 2 次キュー・マネージャーをテストする場合 (つまり、1 次インスタンスがまだ実行中の場合) は、1 次インスタンスを停止して 2 次インスタンスとして再指定する必要があります。

```
endmqm qmname
rdqmdr -m qmname -s
```

2. リカバリー・ノードで以下のコマンドを入力して、2 次キュー・マネージャーを 1 次にします。

```
rdqmdr -m qmname -p
```

3. 次のコマンドを入力して、キュー・マネージャーを開始します。

```
strmqm qmname
```

4. アプリケーションがリカバリー・キュー・マネージャー上のキュー・マネージャーに再接続していることを確認します。代替接続名のリストに 1 次と 2 次のキュー・マネージャーを指定してチャンネルを定義しておく、アプリケーションは自動的に新しい 1 次キュー・マネージャーに接続します。

## 関連情報

[strmqm](#)

[rdqmdr](#)

Linux V 9.0.5 災害復旧構成の障害ノードの交換

災害復旧構成のいずれかのノードを失った場合は、以下の手順を実行し、ノードを置き換えて災害復旧構成を復元できます。

## このタスクについて

メインサイトのノードの修復が不可能になるような障害が発生した場合は、キュー・マネージャーがリカバリー・ノードで実行されている間に、障害が発生したノードを置き換えて、元の災害復旧構成を復元します。置換後のノードでは、障害が発生したノードの ID を使用する必要があります。同じ名前と同じ IP アドレスを使用しなければなりません。

root としてログインするか、mqm グループに属していて必要な sudo 構成を持っているユーザーとしてログインする必要があります。

## 手順

メイン・サイトのキュー・マネージャーが失われた場合は、以下の手順を実行します。

1. リカバリー・ノードで以下のコマンドを実行して、2次キュー・マネージャーに1次の役割を与えます。

```
rdqmdr -m QMname -p
```

*QMname* は、キュー・マネージャーの名前です。

2. 災害復旧を再構成するために置換後の1次ノードで実行する必要があるコマンドを取得します。

```
rdqmdr -m QMname -d
```

このコマンドの出力をコピーします。

3. 以下のコマンドを実行して、キュー・マネージャーを開始します。

```
strmqm QMname
```

4. アプリケーションがリカバリー・ノード上のキュー・マネージャーに再接続したことを確認します。代替接続名のリストに1次と2次のキュー・マネージャーを指定してチャンネルを定義しておくこと、アプリケーションは自動的に新しい1次キュー・マネージャーに接続します。
5. メイン・サイトで障害が発生したノードを置き換え、元のノードの災害復旧構成で使用していた同じ名前と同じIPアドレスを構成します。次に、ステップ2でコピーした **crtmqm** コマンドを実行して、災害復旧を構成します。これで、キュー・マネージャーの二次インスタンスが作成され、プライマリーインスタンスはデータを二次インスタンスと同期化します。
6. 現在の1次インスタンスを終了します。
7. 同期が完了したら、リカバリー・ノードで実行している1次インスタンスをもう一度2次インスタンスに戻します。

```
rdqmdr -m QMname -s
```

8. 置換後の1次ノードで、キュー・マネージャーの2次インスタンスを1次インスタンスにします。

```
rdqmdr -m QMname -p
```

9. 置換後の1次ノードで、キュー・マネージャーを開始します。

```
strmqm QMname
```

これで、メイン・サイトで障害が発生する前の構成を復元できました。

### 関連情報

[strmqm](#)

[rdqmdr](#)

[endmqm](#)

## ロギング: メッセージが失われないようにするための機能

IBM MQ は、キュー・マネージャーが制御する持続データに対する重要な変更をすべてリカバリー・ログに記録します。

ログには、オブジェクトの作成および削除、持続メッセージの更新、トランザクション状態、オブジェクト属性の変更、およびチャンネル活動が含まれます。ログは、メッセージ・キューのすべての更新をリカバリーするために必要な情報を、次の方法によって格納します。

- キュー・マネージャーの変更の記録を保持する

- 再始動プロセス用としてキューの更新の記録を保持する
- ハードウェア障害またはソフトウェア障害の後にデータを復元できるようにする

しかし、IBM MQ は、ログ・ファイルを含め、MQ のファイルをホストするディスク・システムにも依存しています。ディスク・システムそれ自体の信頼性がなければ、依然としてログ情報などの情報が消失するおそれがあります。

## ログの概要

ログは、1 次ファイルと 2 次ファイル、および制御ファイルで構成されます。ログ・ファイルの数とサイズ、およびファイル・システム内の保管場所を定義します。

IBM MQ のログは、次の 2 つのコンポーネントからなっています。

1. 1 つ以上のログ・データ・ファイル。
2. ログ制御ファイル

ログ・データ・ファイルは、ログ・エクステントとも呼ばれます。

記録するデータは、複数のログ・エクステントに含められます。その数とサイズは、(108 ページの『[IBM MQ のログのデフォルト](#)』で説明しているように) 定義することも、システム・デフォルト (1 次エクステントが 3 つで 2 次エクステントが 2 つ) を取ることもできます。

この 3 つの 1 次エクステントと 2 つの 2 次エクステントのデフォルトは 16 MB です。

キュー・マネージャーを作成するときに、割り当てた 1 次 ログ・エクステントの数が、事前に割り振られるログ・エクステントの数になります。数を指定しないと、デフォルト値が使用されます。

IBM MQ では、以下の 2 つのタイプのロギングを使用します。

- 循環
- リニア

リニア・ロギングで使用されるログ・エクステントの数は、メディア・イメージの記録頻度に応じて、非常に大きくなる可能性があります。

詳しくは、[530 ページの『ログのタイプ』](#)を参照してください。

IBM MQ for Windows では、ログのパスを変更していなければ、ログ・エクステントは次のディレクトリーに作成されます。

```
C:\ProgramData\IBM\MQ\log\QMGrName
```

IBM MQ for UNIX and Linux システムでは、ログのパスを変更していなければ、ログ・エクステントは次のディレクトリーに作成されます。

```
/var/mqm/log/QMGrName
```

IBM MQ は、これらの 1 次ログ・エクステントで開始しますが、1 次ログ・スペースが足りなくなった場合、2 次 ログ・エクステントを割り振ります。WebSphere MQ はこれを動的に行い、ログ・スペースの要求が少なくなったら、それを除去します。デフォルトでは、最高 2 個の 2 次ログ・エクステントを割り振ることができます。このデフォルトの割り振りを、[85 ページの『IBM MQ およびキュー・マネージャー構成情報の変更』](#)で説明されている方法で変更することができます。

ログ・エクステントの接頭部は、文字 S または文字 R のいずれかです。アクティブ、非アクティブ、および不必要なエクステントの接頭部は S、再利用エクステントの接頭部は R となります。

キュー・マネージャーのバックアップまたは復元を行うときは、すべてのアクティブ、非アクティブ、および不必要なエクステントをログ制御ファイルと共にバックアップおよび復元します。

注: 再利用エクステントをバックアップおよび復元する必要はありません。

## ログ制御ファイル

ログ制御ファイルには、サイズや場所、次に使用できるエクステンツの名前など、ログ・エクステンツの状態を表すのに必要な情報が含まれています。

**重要:** ログ制御ファイルは、専らキュー・マネージャーの内部使用向けです。

キュー・マネージャーは、リカバリー・ログの状態に関連付けられた制御データを、ログ制御ファイルに保持します。ログ制御ファイルの内容を変更しないでください。

ログ制御ファイルは、ログ・パスにあり、ファイル名は `amqhlctl.lfh` です。キュー・マネージャーのバックアップまたは復元を行うときは、ログ・エクステンツと共に、ログ制御ファイルがバックアップおよび復元されることを確認してください。

## ログのタイプ

IBM MQ では、キュー・マネージャー・アクティビティの記録を維持する方法として、循環ロギングとリニア・ロギングの2つがあります。

### 循環ロギング

ログを使用してシステムの停止時に処理中であったトランザクションをロールバックするだけの再始動リカバリーでよい場合には、循環ロギングを使用してください。

循環ロギングでは、すべての再始動データをログ・ファイルのリングに保持します。ロギングでは、リングの最初のファイルに出力され、それから次のファイルに進むといった方法で、すべてのファイルに出力されます。その後、リングの最初のファイルに戻って、再び出力が開始されます。この操作は、プロダクトが使用されている間ずっと行われるので、ログ・ファイルが不足することは決してありません。

キュー・マネージャーをデータの消失なしに再始動するために必要なログ項目は、キュー・マネージャーのデータ・リカバリーの必要性がなくなるまで IBM MQ によって保持されます。再利用のためにログ・ファイルを解放するためのメカニズムについては、[532 ページの『チェックポイント機能を使用してリカバリーの完了を確認する』](#)で説明されています。

### リニア・ロギング

再始動リカバリーおよびメディア・リカバリーの両方が必要な場合には、リニア・ロギングを使用します。メディア・リカバリーは、ログの内容を再生することにより、消失または損傷したデータを再作成するものです。リニア・ロギングでは、ログ・データを連続したログ・ファイルに保持します。

オプションで、ログ・ファイルを以下のようにすることもできます。

- 再利用する。ただし、これはリスタート・リカバリーやメディア・リカバリー用に必要ではなくなったときだけです。
- 手でアーカイブして、より長い期間保存したり分析したりする。

メディア・イメージの頻度により、リニア・ログ・ファイルの再利用が可能となる時期が決まります。これは、リニア・ログ・ファイル用に使用可能にする必要のあるディスク・スペースのサイズを判別するための主な要素となります。

時間またはログの使用量に基づいて定期的なメディア・イメージの取得を自動的に行うようにキュー・マネージャーを構成すること、またはメディア・イメージを手動でスケジュールすることができます。

管理者は、実装するポリシーを決定し、ディスク・スペース使用量に与える影響を判断します。再始動リカバリーに必要なログ・ファイルは常に使用可能でなければなりません。メディア・リカバリーだけに必要なログ・ファイルは長期保管用にアーカイブすることが可能です(磁気テープなど)。

管理者が自動ログ管理および自動メディア・イメージを有効にした場合、リニア・ロギングは非常に大きな循環ログと同様の方法で動作しますが、メディア・リカバリーによってメディア障害に対する冗長性は向上します。

**V 9.0.4** IBM MQ 9.0.4 以降、`migmqlog` コマンドを使用して、キュー・マネージャーの既存のログ・タイプをリニアから循環に、または循環からリニアに変更できます。

## ロガーの変更

V 9.0.2

Multi

IBM MQ 9.0.2 から、アーカイブ作成を含む自動ログ管理を使用している場合、ロガーは非アクティブなりニア・ログ・エクステントを追跡します。



**重要:** アーカイブなしの自動ログ管理を使用している場合、バックアップ・キュー・マネージャーの使用はこのプロセスではサポートされません。

ULW

ログ・エクステントがリカバリーに必要ななくなったときに、必要に応じてアーカイブされると、ロガーは適切な時点でログ・エクステントを削除するか、またはそのログ・エクステントを再利用します。

再利用されるログ・エクステントは、ログ・シーケンスで次の位置になるように名前変更されます。メッセージ AMQ7490 が定期的に出力されて、作成、削除、または再利用されたエクステントの数を知らせます。

ロガーは、再利用に備えて保持するエクステントの数を選択し、それらのエクステントを削除するタイミングを選択します。

## アクティブ・ログ

リニア・ロギングでも循環ロギングでも、アクティブと呼ばれるいくつかのファイルがあります。循環ロギングとリニア・ロギングのどちらを使用するかにかかわらず、アクティブ・ログはログ・スペースの最大量であり、再始動リカバリーで参照される可能性があります。

アクティブ・ログ・ファイルの数は、通常は、構成ファイルに定義された 1 次ログ・ファイルの数より少なくなります。(数の定義方法の詳細については、[535 ページの『ログのサイズの計算』](#)を参照してください。)

アクティブ・ログ・スペースにはメディア・リカバリーに必要なスペースが含まれないこと、またリニア・ロギングで使用されるログ・ファイルの数はメッセージ・フローやメディア・イメージの頻度によっては非常に多くなることに注意してください。

## 非アクティブ・ログ

再始動リカバリーにログ・ファイルが不要になると、ログ・ファイルは `inactive` になります。再始動リカバリーにもメディア・リカバリーにも必要ないログ・ファイルは、不必要なログ・ファイルと見なすことができます。

自動ログ管理を使用しているとき、それらの不必要なログ・ファイルの処理はキュー・マネージャーによって制御されます。手動のログ管理を選択した場合、操作の対象でなくなったときに不必要なログ・ファイルを管理 (削除やアーカイブなど) することは、管理者の責任となります。

ログ・ファイルの属性指定の詳細については、[541 ページの『ログの管理』](#)を参照してください。

## 2 次ログ・ファイル

2 次ログ・ファイルはリニア・ロギング用として定義されますが、このファイルは通常の操作では使用されません。持続期間の長いトランザクションのために、アクティブ・プールからファイルを解放できないような状態が生じた場合には (そのファイルが再始動に必要とされる可能性がまだ存在するため)、2 次ファイルがフォーマットされて、アクティブ・ログ・ファイル・プールに追加されます。

使用可能な 2 次ファイルの数が使い果たされた場合、ログ活動を必要とするそれ以降のほとんどの操作の要求は拒否されて、MQRC\_RESOURCE\_PROBLEM 戻りコードがアプリケーションに戻されます。また、長期実行トランザクションは非同期ロールバックの対象と見なされます。



**重要:** いずれのタイプのロギングも、不意の停電に対処できます。ただし、ハードウェア障害が起こらないことを想定した場合です。

## チェックポイント機能を使用してリカバリーの完了を確認する

循環ロギングとリニア・ロギングの両方のキュー・マネージャーが再始動リカバリーをサポートしています。直前のキュー・マネージャーのインスタンスがどれほど突然に終了した場合でも (例えば電源異常の場合など)、再始動の際に、キュー・マネージャーはその永続的な状態を終了時点での正しいトランザクション状態に復元します。

再始動リカバリーは、ディスクの保全性が維持されていることに依存します。同様に、オペレーティング・システムには、オペレーティング・システムがどれほど突然に終了する場合でもディスクの保全性が維持される機能が必要です。

ディスクの保全性が維持されない稀なケースでは、リニア・ロギング (およびメディア・リカバリー) により、追加の冗長性と回復可能性のオプションが提供されます。普及を続ける RAID などのテクノロジーによって、ディスクの保全性の問題が生じる可能性はますます小さくなっているため、多くの企業では循環ロギングを構成して、再始動リカバリーだけを使用しています。

IBM MQ は、従来型のログ先行書き込みリソース・マネージャーとして設計されています。メッセージ・キューに対する持続更新は、2段階で行われます。

1. 更新を表すログ・レコードがリカバリー・ログに確実に書き込まれます。
2. キュー・ファイルやバッファは、システムにとって最も効率的な方法で更新されますが、一貫して同じ方法であるとは限りません。

そのため、ログ・ファイルの方が、基礎となるキュー・バッファやファイル状態よりも新しい状態になることがあります。

この状態が収まることなく続くと、クラッシュ・リカバリーの後、キューの状態を整合性のあるものとするために大量のログ再生が必要となります。

IBM MQ は、クラッシュ・リカバリーの後に必要なログ再生の量を制限するために、checkpoints を使用します。ログ・ファイルをアクティブと呼ぶかどうかを制御するキー・イベントは、checkpoint です。

IBM MQ チェックポイントは、以下のようなポイントです。

- リカバリー・ログとオブジェクト・ファイル間の整合点。
- 後続のログ・レコードの順方向再生により、キュー・マネージャーが終了した可能性のある時点での正しい論理状態にキューが復元されることが保証される、ログ内の場所を示す点。

チェックポイントの際に、IBM MQ は必要に応じて以前の更新内容をキュー・ファイルに移動し、クラッシュ・リカバリーの後にキューを整合性のある状態に戻すために再生しなければならないログ・レコードの量を制限します。

最新の完全なチェックポイントは、クラッシュ・リカバリーの際に再生を実行する開始点とする、ログ内のポイントになります。したがって、チェックポイントの頻度は、チェックポイントを記録するためのオーバーヘッドと、それらのチェックポイントによる潜在的なリカバリー時間の改善との間のトレードオフによって決まります。

最新の完全なチェックポイントが開始するログ内の位置は、ログ・ファイルがアクティブか非アクティブかを判別する主要な要因の1つとなります。もう1つの主要な要因は、現行のアクティブ・トランザクションによる最初の持続更新に関連した最初のログ・レコードのログ内での位置です。

新しいチェックポイントが2番目以降のログ・ファイルに記録された場合、現在のトランザクションが最初のログ・ファイル内のログ・レコードを参照していなければ、最初のログ・ファイルは非アクティブになります。循環ロギングの場合、これにより最初のログ・ファイルの再利用が可能になります。リニア・ロギングの場合、メディア・リカバリーのために通常は最初のログ・ファイルが引き続き必要となります。

循環ロギングまたは自動ログ管理のいずれかを構成した場合、キュー・マネージャーが非アクティブ・ログ・ファイルを管理します。手動ログ管理によってリニア・ロギングを構成した場合、操作の要件に従って非アクティブ・ファイルを管理することは、管理タスクの1つになります。

IBM MQ は、自動的にチェックポイントを生成します。チェックポイント取得のタイミングは以下のとおりです。

- キュー・マネージャーの開始時

- シャットダウン時
- ログ・スペースの不足時
- **Multi** 直前のチェックポイントが取得されて以来 50,000 件の操作がログに記録された後
- **z/OS** 前回のチェックポイント取得時以降、*number\_of\_operations* 件の操作がログに記録された後 (*number\_of\_operations* は、**LOGLOAD** プロパティで設定された操作数です)。

IBM MQ は、再始動時に、ログの中の最新のチェックポイント・レコードを見つけます。その情報は、すべてのチェックポイントの最後に更新されるチェックポイント・ファイルに保持されています。チェックポイント以降に発生した操作すべてが、順方向に再生されます。これを、やり直しフェーズと呼びます。

やり直しフェーズによって、キューがシステム障害やシャットダウンの前の論理的状态に戻ります。やり直しフェーズの間に、システム障害やシャットダウンが発生したときに未完了であったトランザクションのリストが作成されます。

**Multi** 再生フェーズの進行状況を示すメッセージ AMQ7229 および AMQ7230 が発行されます。

IBM MQ は、バックアウトまたはコミットする操作を知るために、未完了トランザクションに関連付けられた各アクティブ・ログ・レコードにアクセスします。これを、リカバリー・フェーズと呼びます。

**Multi** リカバリー・フェーズの経過を示すために、メッセージ AMQ7231、AMQ7232、および AMQ7234 が出されます。

リカバリー・フェーズの間に必要なすべてのログ・レコードにアクセスした後、次に各アクティブ・トランザクションが解決され、トランザクションに関連付けられた各操作がバックアウトまたはコミットされます。これを、解決フェーズと呼びます。

**Multi** 解決フェーズの経過を示すために、メッセージ AMQ7233 が出されます。

**z/OS** z/OS では、再始動処理はさまざまなフェーズで構成されます。

1. ページ・セットに必要なメディア・リカバリー、および作業単位をバックアウトするためと未確定の作業単位のロックを取得するために必要な最も古いログ・レコードに基づいて、リカバリー・ログの範囲が確立されます。
2. ログ範囲が決定されると、順方向のログの読み取りが行われて、ページ・セットが最新の状態になり、未確定または未完了の作業単位に関連したすべてのメッセージがロックされます。
3. 順方向のログ読み取りが完了すると、ログは逆方向に読み取られて、障害発生時に未確定または未完了だった作業単位がバックアウトされます。

**z/OS** 表示される可能性のあるメッセージの例を以下に示します。

```
CSQR001I +MQOX RESTART INITIATED
CSQR003I +MQOX RESTART - PRIOR CHECKPOINT RBA=00000001E48C0A5E
CSQR004I +MQOX RESTART - UR COUNTS - 806
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN BACKOUT=0
CSQR030I +MQOX Forward recovery log range 815
from RBA=00000001E45FF7AD to RBA=00000001E48C1882
CSQR005I +MQOX RESTART - FORWARD RECOVERY COMPLETE - 816
IN COMMIT=0, INDOUBT=0
CSQR032I +MQOX Backward recovery log range 817
from RBA=00000001E48C1882 to RBA=00000001E48C1882
CSQR006I +MQOX RESTART - BACKWARD RECOVERY COMPLETE - 818
INFLIGHT=0, IN BACKOUT=0
CSQR002I +MQOX RESTART COMPLETED
```

注：読み取るログが大量にある場合は、メッセージ CSQR031I (順方向リカバリー) および CSQR033I (逆方向リカバリー) が定期的に発行されて、進捗が示されます。

534 ページの図 86 では、最新のチェックポイント (チェックポイント 2) より前のすべてのレコードは、IBM MQ にはもう必要ありません。このチェックポイント情報およびその後のログ項目から、キューをリカバリーすることができます。循環ロギングでは、チェックポイントより前に解放されたファイルは再利用できます。リニア・ロギングでは、解放されたログ・ファイルは、通常の操作ではアクセスする必要がなくなり、非アクティブになります。この例では、キュー・ヘッド・ポインターが最新のチェックポイン

トであるチェックポイント2をポイントするように移動されます。Checkpoint 2は、新しいキュー・ヘッド、ヘッド2になります。ログ・ファイル1を再使用できるようになりました。

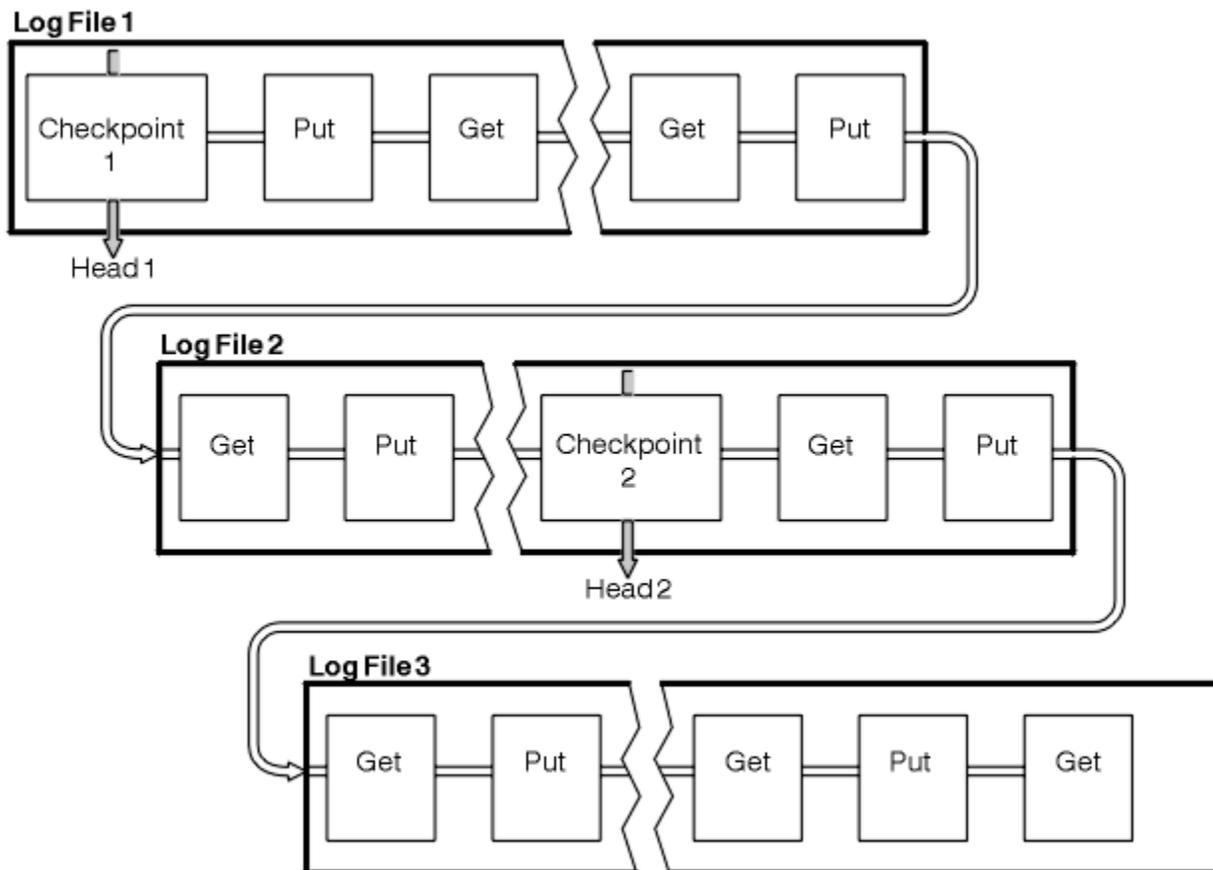


図 86. チェックポイント機能

### 持続期間の長いトランザクションを使用したチェックポイント機能

持続期間の長いトランザクションが、ログ・ファイルの再利用にどんな影響を与えるかを示したものです。

535 ページの図 87 は、持続期間の長いトランザクションが、ログ・ファイルの再利用にどんな影響を与えるかを示したものです。例では、持続期間の長いトランザクションによって、示されている最初のチェックポイントの後に LR 1 で示されているログへの入力が始まりました。3 番目のチェックポイントの後までトランザクションは完了しません (LR 2 の地点)。LR 1 以降のすべてのログ情報は、このトランザクションのリカバリーを行うために、必要であればトランザクションが完了するまで保持されます。

持続期間の長いトランザクションが完了した後に (LR 2)、ログのヘッドがチェックポイント 3 (ログに記録された最も新しいチェックポイント) に論理的に移されます。チェックポイント 3 (ヘッド 2) より前のログ・レコードの入ったファイルは、不要になります。循環ロギングを使用している場合には、スペースは再利用できます。

持続期間の長いトランザクションが完了する前に 1 次ログ・ファイルがすべて満杯になった場合は、ログフルを防止するために、2 次ログ・ファイルが使用される可能性があります。

完全にキュー・マネージャーの制御下にあるアクティビティー (チェックポイントなど) については、そのアクティビティーを 1 次ログ内に保持する試行が行われるようにスケジュールされています。

ただし、キュー・マネージャーの制御下でない動作 (いずれかのトランザクションの期間など) をサポートするために 2 次ログ・スペースが必要なとき、そのアクティビティーが完了するように、キュー・マネージャーは定義されたいずれかの 2 次ログ・スペースの使用を試みます。

合計ログ・スペースの 80% を使用してもそのアクティビティーが完了しない場合、キュー・マネージャーは、それがアプリケーションに影響を与えることになるとしても、ログ・スペースを再要求するアクションを開始します。

循環ロギングを使用している場合に、ログのヘッドが移されたときは、1次ログ・ファイルが再利用できます。ロガーは、カレント・ファイルが満杯になった後に、使用できる最初の1次ファイルを利用します。リニア・ロギングを使用している場合は、ログのヘッドはアクティブ・プールのさらに後方に移されて、最初のファイルが非アクティブになります。新しい1次ファイルがフォーマットされて、将来のロギング活動に備えてプールの下部に追加されます。

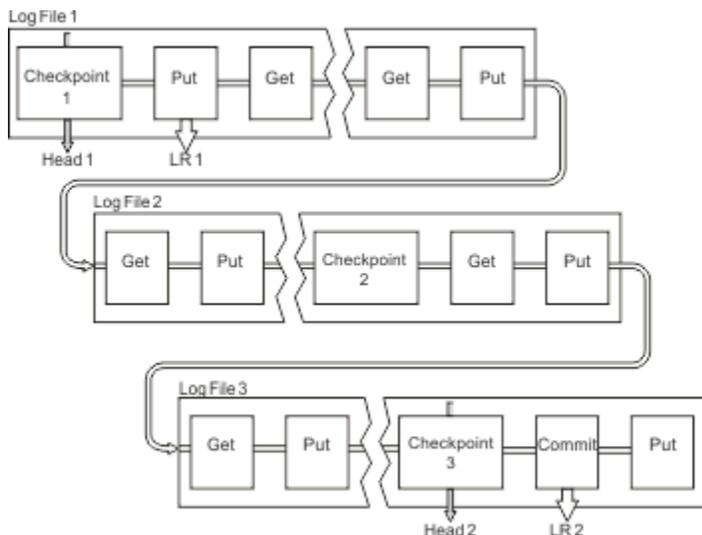


図 87. 持続期間の長いトランザクションでのチェックポイント機能

## ログのサイズの計算

キュー・マネージャーが必要とするログのサイズを見積もります。

キュー・マネージャーで循環ロギングとリニア・ロギングのどちらを使用するかを決めた後、キュー・マネージャーに必要なアクティブ・ログのサイズを見積もる必要があります。アクティブ・ログのサイズは、次のログ構成パラメーターによって決まります。

### LogFilePages

4K ページを 1 単位として表した、個々の 1 次ログ・ファイルまたは 2 次ログ・ファイルのサイズ

### LogPrimaryFiles

事前に割り振る 1 次ログ・ファイルの数

### LogSecondaryFiles

1 次ログ・ファイルが満杯になってきたときに使用するために作成可能な 2 次ログ・ファイルの数

注:

- 1 次ログ・ファイルと 2 次ログ・ファイルの数は、キュー・マネージャーを始動するたびに変更することができます。ただし、2 次ログに対して行った変更の効果はすぐにはわからない可能性があります。
- ログ・ファイル・サイズは変更できません。キュー・マネージャーの作成前に決定する必要があります。
- 1 次ログ・ファイルの数およびログ・ファイル・サイズによって、キュー・マネージャーの作成時に事前に割り振るログ・スペースの量が決まります。
- 1 次ログ・ファイルと 2 次ログ・ファイルの合計数は、UNIX and Linux システムでは 511 以下、Windows では 255 以下にする必要があります。このため、長時間続くトランザクションでは、キュー・マネージャーが再始動リカバリーに使用できるログ・スペースの最大量が制限されます。メディア・リカバリーのためにキュー・マネージャーが使用するログ・スペースの量は、この制限には拘束されません。
- 循環ロギングを使用している場合は、キュー・マネージャーは 1 次および 2 次ログ・スペースを再利用します。ログ・ファイルが満杯になり、シーケンス内の次の 1 次ログ・ファイルが使用できない場合には、キュー・マネージャーは 2 次ログ・ファイルを割り振り、限度に達するまでこれを続けます。

割り振る必要があるログの数については、536 ページの『アクティブ・ログの大きさの決定方法』を参照してください。1 次ログのエクステンツは順番に使用され、その順番は変わりません。

例えば、3つの1次ログ0、1、および2があるとすると、0、1、2の順に使用され、続いて1、2、0、2、0、1が使用され、0、1、2に戻ります。割り振った2次ログが、必要に応じてこの間に使用されません。

6. チェックポイントの時点で、1次ログ・ファイルは再利用できるようになります。ログ・スペースの量は徐々に小さくなるので、キュー・マネージャーは、1次ログ・スペースおよび2次ログ・スペースの両方を考慮に入れてチェックポイントをとります。

**V9.0.2** IBM MQ 9.0.2からは、キュー・マネージャーは、ログの使用が1次エクステント内に保持されるようにチェックポイントをスケジュールすることを試みます。

詳しくは、[108 ページの『IBM MQ のログのデフォルト』](#)を参照してください。

## アクティブ・ログの大きさの決定方法

キュー・マネージャーが必要とするアクティブ・ログのサイズを見積もります。

アクティブ・ログのサイズの上限を次に示します。

```
logsize = (primaryfiles + secondaryfiles) * logfilepages * 4096
```

ログは、キュー・マネージャーがディスクに書き込む1秒あたりのデータ量が最大の状態で、実行時間が最長のトランザクションを処理できる十分な大きさにする必要があります。

実行時間が最長のトランザクションの実行時間がN秒で、キュー・マネージャーがディスクに書き込む1秒あたりの最大データ量がログで1秒あたりBバイトである場合、ログを次のサイズ以上にする必要があります。

```
logsize >= 2 * (N+1) * B
```

キュー・マネージャーがディスクに書き込む1秒あたりのデータ量が最大になる可能性があるのは、ワークロードのピーク時に実行する場合、あるいはメディア・イメージを記録する場合です。

トランザクションの実行時間が長いために、最初のログ・レコードが入ったログ・エクステントがアクティブ・ログ内に含まれなくなると、キュー・マネージャーはアクティブ・ログを一度に1つずつ、ログ・レコードが最も古いものからロール・バックします。

1次ファイルおよび2次ファイルの最大数が使用される前に、キュー・マネージャーは、古いログ・エクステントを非アクティブにして、他のログ・エクステントを割り振る必要があります。

実行時間が最長のトランザクションを実行する時間を決定します(その時間を超えると、キュー・マネージャーはそのトランザクションをロールバックできます)。実行時間が最長のトランザクションは低速なネットワークのトラフィックを待機している可能性があります。あるいは、トランザクションの設計が不十分でユーザー入力を待機している可能性があります。

実行時間が最長のトランザクションの実行時間を調べるには、次の **runmqsc** コマンドを実行します。

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

**dspmqrtn -a** コマンドを実行すると、すべての状態のすべてのXAおよび非XAコマンドが表示されます。

このコマンドを実行すると、すべての現行トランザクションについて最初のログ・レコードが書き込まれた日時がリストされます。



**重要:** ログ・サイズを計算するという目的で重要なのは、アプリケーション/トランザクションが開始されてから経過した時間ではなく、最初のログ・レコードが書き込まれてから経過した時間です。実行時間が最長のトランザクションの実行時間を最も近い秒に切り上げます。これは、キュー・マネージャーの最適化のためです。

例えば、MQGET呼び出しの発行によって開始されるアプリケーションの場合は、実際にメッセージを取得するまでに一定の時間待機するので、最初のログ・レコードが書き込まれるのはアプリケーションが開始されてからかなり後になります。

以下からの最大監視日時出力を調べます。

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

このコマンドは、最初に発行した、現在の日時からのものです。これにより、実行時間が最長のトランザクションの実行時間を推定できます。

実行時間が最長のトランザクションの実行時間を過小評価しないように、そのトランザクションをワークロードのピーク時に実行し、その間にこの **runmqsc** コマンドを繰り返し実行してください。

IBM MQ 8.0 では、オペレーティング・システムのツール (例えば UNIX プラットフォームの **iostat**) を使用します。

IBM MQ 9.0 以降では、次のコマンドを実行して、キュー・マネージャーがログに書き込んだ 1 秒あたりのバイト数を調べることができます。

```
amqsrua -m qmgr -c DISK -t Log
```

「logical bytes written」が、キュー・マネージャーがログに書き込んだ 1 秒あたりのバイト数を示しています。以下に例を示します。

```
$ amqsrua -m mark -c DISK -t Log
Publication received PutDate:20160920 PutTime:15383157 Interval:4 minutes,39.579 seconds
Log - bytes in use 37748736
Log - bytes max 50331648
Log file system - bytes in use 316243968
Log file system - bytes max 5368709120
Log - physical bytes written 4334030848 15501948/sec
Log - logical bytes written 3567624710 12760669/sec
Log - write latency 411 uSec
```

この例では、1 秒あたりにログに書き込まれた論理バイト数は 12760669/sec (約 12 MiB/秒) です。

次を使用して

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

実行時間が最長のトランザクションが以下であることがわかります。

```
CONN(57E14F6820700069)
EXTCONN(414D51436D61726B2020202020202020)
TYPE(CONN)
APPLTAG(msginteg_r)                UOWLOGDA(2016-09-20)
UOWLOGTI(16.44.14)
```

現在の日時が 2016-09-20 16.44.19 なので、このトランザクションは 5 秒間実行されていたこととなります。しかし、キュー・マネージャーでロールバックするまでにトランザクションを 10 秒間実行することを許容することにします。したがって、必要なログ・サイズは次のようになります。

```
2 * (10 + 1) * 12 = 264 MiB
```

ログ・ファイルの数は、(前のテキストで計算された) 予期される最大サイズのログを格納できるものでなければなりません。これは次のようになります。

ログ・ファイルの最小数 = (必要なログ・サイズ) / (**LogFilePages** \* ログ・ファイルのページ・サイズ (4096))

**LogFilePages** のデフォルト値 4096 と、前のテキストで計算されたログ・サイズの見積もり 264MiB を使用すると、ログ・ファイルの最小数は次のようになります。

```
264MiB / (4096 x 4096) = 16.5
```

つまり、17 個のログ・ファイルです。

予想されるワークロードの実行が 1 次ファイル内に収まるようにログのサイズを設定する場合

- 2次ファイルは、追加のログ・スペースが必要になった場合のための、緊急用になります。
- 事前割り振りされた1次ファイルを常に使用する循環ロギングは、2次ファイルの割り振りと解放を行うよりわずかに速くなります。
- キュー・マネージャーは1次ファイルに残されているスペースのみを使用して、次のチェックポイントを取る時点を計算します。

したがって、上記の例では、ワークロードが1次ログ・ファイルの中で実行するように、次の値を設定します。

- **LogFilePages = 4096**
- **LogPrimaryFiles = 17**
- **LogSecondaryFiles = 5**

次の事項に注意してください。

- この例では、5つの2次ファイルがアクティブ・ログ・スペースの20パーセントを超えています。

**V 9.0.2** IBM MQ 9.0.2 から、ロガーはワークロードを1次ファイルの中だけに保管するように試行します。そのため、ロガーは、1次ファイルの一部だけがフルになるとチェックポイントをスケジュールします。

**V 9.0.2** 2次ファイルは、予期せぬ長期実行トランザクションが生じた場合に備えて、緊急用に用意します。

合計ログ・スペースの80パーセントより多くが使用されると、キュー・マネージャーはログ・スペースの使用を減らすための処理を行うことに注意してください。

- リニア・ロギングまたは循環ロギングのどちらを使用する場合も同じ計算を行います。  
アクティブ・ログの概念はリニア・ロギングでも循環ロギングでも同じであるため、リニア・アクティブ・ログと循環アクティブ・ログのどちらを計算しているかは問題ではありません。
- メディア・リカバリーだけに必要なログ・エクステンツはアクティブ・ログ内にないため、1次ファイルおよび2次ファイルの数にはカウントされません。
- **V 9.0.2** IBM MQ 9.0.2 以降、`DISPLAY QMSTATUS LOG` の `LOGUTIL` フィールドを使用して、必要なアクティブ・ログのサイズを概算することができます。

このフィールドは、長期実行トランザクションの期間、またはキュー・マネージャーのスループットのピークを判別するために、常にサンプリングすることなく、必要なログ・サイズの妥当な見積もりを作成できるように設計されています。

## LogFilePages の大きさの決定方法

通常は、LogFilePages を十分な大きさにしてください。そうすれば、1次ファイルの最大数に達しないようにアクティブ・ログのサイズを簡単に大きくできます。小さなログ・ファイルがたくさんあるよりも、大きなログ・ファイルが少ない数だけの方が望ましいと言えます。その方が、必要に応じてログのサイズを大きくする作業を柔軟に行えるからです。

リニア・ロギングの場合は、ログ・ファイルが非常に大きくなると、パフォーマンスが変わりやすくなります。ログ・ファイルが非常に大きくなると、新しいログ・ファイルを作成して形式を設定したり古いログ・ファイルをアーカイブに保存したりする手順が大掛かりになります。この問題が大きくなるのは、手動ログ管理やアーカイブ・ログ管理の場合です。自動ログ管理では、新しいログ・ファイルが作成されることはまれだからです。

## ログを小さくしすぎた場合の結果

ログの最小サイズを見積もる際に考慮すべきポイントについて説明します。

ログを小さくしすぎた場合:

- 実行時間の長いトランザクションがバックアウトされます。
- 前のチェックポイントが終了する前に次のチェックポイントが開始しようとします。

**重要:** ログ・サイズの見積もりが不正確でも、データの保全性は保たれます。

チェックポイントの説明については、532 ページの『[チェックポイント機能を使用してリカバリーの完了を確認する](#)』を参照してください。アクティブ・ログ・エクステントに残されたログ・スペース量が不足してくると、キュー・マネージャーがチェックポイントをスケジュールする頻度が高くなります。

チェックポイントは一瞬ではなく一定の時間がかかります。チェックポイントで記録しなければならないデータが多くなるほど、チェックポイントにかかる時間は長くなります。ログが小さいとチェックポイントがオーバーラップする可能性があります。つまり、前のチェックポイントが終了する前に次のチェックポイントが要求されます。これが発生すると、エラー・メッセージが書き込まれます。

実行時間の長いトランザクションがバックアウトされたり、チェックポイントがオーバーラップしたりしても、キュー・マネージャーはワークロードの処理を続けます。実行時間の短いトランザクションは通常どおり実行できます。

しかし、キュー・マネージャーの実行状況は好ましいものではなく、パフォーマンスが低下する可能性があります。十分なログ・スペースを割り当ててキュー・マネージャーを再始動する必要があります。

## ログを大きくしすぎた場合の結果

ログの最大サイズを見積もる際に考慮すべきポイントについて説明します。

ログを大きくしすぎた場合:

- めったにありませんが、緊急時再始動にかかる時間が長くなる可能性があります。
- 不要なディスク・スペースを使用することになります。
- 実行時間の非常に長いトランザクションが許容されます。

**重要:** ログ・サイズの見積もりが不正確でも、データの保全性は保たれます。

**V9.0.2** ログ使用率統計を使用すると、ログの最大サイズを見積もるために役立ちます。詳しくは、545 ページの『[IMGLOGLN と IMGINTVL の設定値の決定](#)』および [ALTER QMGR](#) を参照してください。

キュー・マネージャーが再始動時にログを読み取る方法について詳しくは、532 ページの『[チェックポイント機能を使用してリカバリーの完了を確認する](#)』を参照してください。キュー・マネージャーは最後のチェックポイントからログを再生し、キュー・マネージャーの終了時にアクティブだったすべてのトランザクションを解決します。

トランザクションを解決するために、キュー・マネージャーはそのトランザクションに関連するすべてのログ・レコードを読み直します。それらのログ・レコードが最後のチェックポイントより前のものである場合もあります。

キュー・マネージャーに非常に大きなログを割り振れば、キュー・マネージャーは再始動時にログ内のすべてのログ・レコードを読み取ることができますが、通常、これが必要になることはありません。これが行われるような状況が万が一発生した場合、そのプロセスには非常に長い時間がかかります。

チェックポイントが予期せず停止した後にキュー・マネージャーが終了した場合、ログが大きいと、キュー・マネージャーの再始動時間が大幅に増加します。ログ・サイズを制限することで、緊急時再始動にかかる時間を制限できます。

このような問題を回避するには、次のようにする必要があります。

- ワークロードが十分に収まるが、大きすぎないログにする。
- 実行時間の長いトランザクションは回避する。

## **V9.0.2** ログ・ファイル・システムの大きさの決定方法

キュー・マネージャーが必要とするログ・ファイル・システムのサイズの見積もり。

ログ・ファイル・システムを大きくして、キュー・マネージャーがログ記録を書き込むためのスペースが十分にあるようにすることは大切です。ログ・ファイル・システムがキュー・マネージャーによって完全にフルにされると、FFDC が書き出され、トランザクションはロールバックして、キュー・マネージャーが突然終了することがあります。

ログのために予約するディスク・スペースの量は、少なくともアクティブ・ログと同じ大きさでなければなりません。その正確な大きさは、以下の要素に依存します。

- ログ・タイプの選択 (リニアまたは循環)
- アクティブ・ログのサイズ (1次ファイル、2次ファイル、ログ・ファイル・ページ)
- ログ管理の選択 (手動、自動、またはアーカイブ)
- オブジェクトが損傷した場合のコンティンジェンシー・プラン

循環ログを選択した場合、ログ・ファイル・システムには次のサイズが必要です。

```
LogFilesystemSize >= (PrimaryFiles + SecondaryFiles + 1) * LogFileSize
```

これにより、キュー・マネージャーはすべての1次ファイルと2次ファイルに書き込むことができます。例外的な状況では、キュー・マネージャーが2次ファイルの数を超過して追加のエクステントを書き込むことがあります。前述のアルゴリズムは、それを考慮に入れたものです。

リニア・ログを選択した場合、ログ・ファイル・システムはアクティブ・ログよりもかなり大きくなります。

手動ログ管理を選択すると、キュー・マネージャーは、必要に応じて新しいログ・エクステントへの書き込みを続行します。ユーザーは、必要なくなったときにそれらを削除 (およびアーカイブ) する責任があります。

ログ・ファイル・システムに必要な大きさは、不必要なエクステントや非アクティブのエクステントを削除する戦略に大きく依存しています。

非アクティブになったエクステント (再始動リカバリーには不要) をすぐにアーカイブして削除するように決めること、または不必要なエクステント (メディア・リカバリーや再始動リカバリーには不要) だけをアーカイブして削除するように決めることができます。

不必要なエクステントだけをアーカイブおよび削除する場合、損傷したオブジェクトがあれば、**MEDIALOG** は進行しません。そのため、さらに多くのエクステントが不必要になることはありません。オブジェクトをリカバリーするなどして、問題を解決するまでは、エクステントのアーカイブと削除を停止します。

ワークロードを停止しない場合、問題を解決するために使用できる時間は、ログ・ファイル・システムのサイズによって決まります。そのため、ベスト・プラクティスとして、リニア・ロギングを使用する場合には十分に大きなログ・ファイル・システムを用意してください。

リニア・ログと、自動またはアーカイブ・ログ管理を選択した場合、キュー・マネージャーはログ・エクステントを再利用します。

再使用できるログ・エクステントには、接頭部として R という文字が付けられます。メディア・イメージが記録される時に、余分なエクステントがアーカイブされるので、キュー・マネージャーはそれらのエクステントを再利用することができます。

そのため、再使用エクステントは、メディア・イメージの間のログに書き込まれるデータの長さ未満になります。

```
ReuseExtents <= LogDataLengthBetweenMediaImages
```

メディア・イメージを自動的に記録して **IMGLOGLN** を設定するとき、**IMGLOGLN** は固定された最大値ではなく目標値なので、**LogDataLengthBetweenMediaImages** は **IMGLOGLN** の2倍の大きさになることもあります。

メディア・イメージを手動で記録するとき、またはインターバルに基づいて自動的に記録するとき、**LogDataLengthBetweenMediaImages** は、ワークロードおよび複数のイメージを取得する間隔によって決まります。

アクティブ・エクステントおよび再利用エクステントに加えて、非アクティブ・エクステント (メディア・リカバリーにのみ必要) および不必要なエクステント (再始動リカバリーやメディア・リカバリーには不要) もあります。

自動またはアーカイブ・ログ管理を使用する場合、キュー・マネージャーは、メディア・リカバリーに必要なエクステントを再利用しません。したがって、非アクティブ・エクステントの数は、メディア・イメージを取得する頻度、およびそれらを手動または自動のどちらで取得するかによって異なります。

**IMGINTVL** および **IMGLOGLN** は目標値であり、メディア・イメージ間の固定された最小値や最大値ではありません。ただし、必要となる可能性のあるログ・ファイル・システムの最大サイズを見積もるとき、自動メディア・イメージが、**IMGINTVL** または **IMGLOGLN** の 2 倍よりも大きく記録されることはまずありません。

自動またはアーカイブ・ログ管理を使用してログ・ファイル・システムのサイズを決める際には、キューまたは他のオブジェクトに損傷が生じた場合にも検討する必要があります。その場合、キュー・マネージャーは損傷したオブジェクトのメディア・イメージを取得できないので、**MEDIALOG** は進行しません。

ワークロードが続く場合、メディア・リカバリーに必要な最も古いエクステントは引き続き必要となり、再利用できないので、非アクティブ・ログが無制限に拡大します。ワークロードが続く場合には、ログ・ファイル・システムが完全にフルになり、キュー・マネージャーがトランザクションのロールバックを開始して、突然停止する可能性さえ生じる前に、問題を解決する必要があります。

そのため、自動ログ管理とアーカイブ・ログ管理:

```
LogFilesystemSize > (PrimaryFiles + SecondaryFiles +  
  (((TimeBetweenMediaImages * 2) + TimeNeededToResolveDamagedObject) * ExtentsUsedPerHour))  
  * LogFilePages
```

注: 上記のアルゴリズムでは、メディア・リカバリーに必要ななくなった時点で、アーカイブ・ログ管理のために、**SET LOG ARCHIVED** がエクステントごとに呼び出されることを前提としています。

## ログの管理

**V 9.0.2** IBM MQ 9.0.2 以降、製品は自動ログ管理と、リニア・ログの自動メディア・リカバリーをサポートしています。循環ログは、ほぼ自己管理されますが、スペースの問題を解決するために介入が必要になることがあります。

循環ロギングでは、キュー・マネージャーはログ・ファイルの空きスペースを再利用します。この活動はユーザーには気付かれないうで行われます。割り振られるスペースは迅速に再利用されるため、使用されるディスク・スペースの量が減少するのは通常は分かりません。

**V 9.0.2** IBM MQ 9.0.2 から、循環ロギングを使用するときに 2 次ファイルを削除できます。詳しくは、**RESET QMGR TYPE (REDUCELOG)** を参照してください。

リニア・ロギングでは、長い間チェックポイントが取られていない場合や、持続期間の長いトランザクションがはるか前にログ・レコードを作成した場合に、ログがフルになる可能性があります。キュー・マネージャーは、最初の問題を回避するために、頻繁にチェックポイントを取ろうとします。

**Multi** ログが満杯になると、メッセージ AMQ7463 が出されます。さらに、持続期間の長いトランザクションがスペースの解放を妨げているためにログが満杯になった場合、メッセージ AMQ7465 が出されます。

ログ・レコードのうち、最後の完全なチェックポイントの開始以降に書き込まれたもの、およびアクティブ・トランザクションによって書き込まれたもののみが、キュー・マネージャーの再始動に必要です。

時間の経過と共に、作成された最も古いログ・レコードはキュー・マネージャーの再始動には不要になります。

持続期間の長いトランザクションが検出されると、そのトランザクションを非同期でロールバックするようにアクティビティがスケジュールされます。何らかの予期しない理由により、その非同期ロールバックが失敗した場合、一部の MQI 呼び出しはその状況で **MQRC\_RESOURCE\_PROBLEM** を戻します。

**MQCMIT** や **MQBACK** が失敗しないように、すべての未完了トランザクションをコミットまたはロールバックするためのスペースが予約されていることに注意してください。

キュー・マネージャーは、長い期間実行するトランザクションをロールバックします。トランザクションがこのようにしてロールバックされたアプリケーションでは、それ以降の **MQPUT** や **MQGET** の操作を、同じトランザクションの下で同期点を指定して実行することができなくなります。

ただし、トランザクションを手動で終了すると、新しいログが開始します。新しいログ・スペースは即時に割り振られますが、解放されたログ・スペースが完全に空になるには一定の時間がかかることに注意してください。

この状態の同期点でメッセージの書き込みまたは読み取りを行おうとすると、MQRC\_BACKED\_OUT が戻されます。この場合、アプリケーションは、MQCMIT を出す (MQRC\_BACKED\_OUT が戻される) か、MQBACK を出して、新しいトランザクションを開始することができます。過度にログ・スペースを消費するトランザクションがロールバックされると、そのログ・スペースが解放され、キュー・マネージャーは操作を正常に続行します。

## ディスクが満杯になったときに何が起きるか

キュー・マネージャーのロギング・コンポーネントは、満杯のディスクおよび満杯のログ・ファイルに対処できます。ログ・ファイルを含んでいるディスクが満杯になると、キュー・マネージャーはメッセージ AMQ6709 を出し、エラー・レコードが取られます。

ログ・ファイルは、ログ記録を書き込むにつれて拡張されるのではなく、固定サイズで作成されます。これは、IBM MQ が新しいファイルを作成しているときにのみ、ディスク・スペースが不足する可能性があることを意味します。レコードをログに書き込んでいるときにスペースが不足することはありません。IBM MQ は、既存のログ・ファイルに使用可能なスペースがどれだけあるかを常に認識しており、ファイル内のスペースをそれに応じて管理します。

**V 9.0.2** IBM MQ 9.0.2 以降では、リニア・ロギングを使用する場合に、以下のオプションが選択できます。

- ログ・エクステントの自動管理。

新しいログ属性の詳細については、[DISPLAY QMSTATUS](#) を参照してください。

また、以下のコマンド、または PCF でこれらに相当するコマンドも参照してください。

- [RESET QMGR](#)
- [SET LOG](#) (分散プラットフォームの場合)

- メディア・イメージの使用を制御するオプション。

以下の詳細については、[ALTER QMGR](#) コマンドと [ALTER QUEUES](#) コマンドを参照してください:

- [IMGINTVL](#)
- [IMGLOGLN](#)
- [IMGRCOVO](#)
- [IMGRCOVQ](#)
- [IMGSCHEM](#)

循環ロギングでは、リソース問題が返されます。

それでもスペースが足りない場合には、キュー・マネージャーの構成ファイル内のログの構成が正しいかどうか確認してください。使用可能なスペースよりもログが大きくなるように、1 次または 2 次ログ・ファイルの数を減らすこともできます。

既存のキュー・マネージャーでは、ログ・ファイルのサイズを変更することはできません。キュー・マネージャーは、すべてのログ・エクステントが同じサイズであることを必要とします。

## ログ・ファイルを管理する

ログ・ファイルに十分なスペースを割り振ります。リニア・ログでは、古いログ・ファイルが必要ではなくなったときに削除することができます。

## 循環ロギングに固有の情報

循環ロギングを使用している場合、システムの構成時にはログ・ファイルを保持できるだけの十分なスペースを確保してください ([108 ページの『IBM MQ のログのデフォルト』](#)および [116 ページの『キュー・マネージャーのログ』](#)を参照)。ログが使用するディスク・スペースの量は、必要な時に作成される 2 次ファイルのスペースも含めて、構成されたサイズを超えることはありません。

## リニア・ロギングに固有の情報

リニア・ログを使用している場合は、データがログに記録されるにつれて、ログ・ファイルが連続的に追加され、使用されるディスク・スペースの量は時間と共に増大します。データのログの速度が速いと、ディスク・スペースは新しいログ・ファイルによって急速に消費されます。

時間が経過すると、リニア・ログの古いログ・ファイルは、キュー・マネージャーを再始動したり、損傷したオブジェクトのメディア・リカバリーを実行するために必要なくなります。以下に、この時点でどのログ・ファイルが必要かを判断するための方法を示します。

### ロガー・イベント・メッセージ

重要なイベント (例えば、メディア・イメージの記録など) が発生すると、ロガー・イベント・メッセージが生成されます。ロガー・イベント・メッセージの内容により、キュー・マネージャーの再開とメディアの回復のために現時点に必要なログ・ファイルが指定されます。ロガー・イベント・メッセージについて詳しくは、[ロガー・イベント](#)を参照してください。

### キュー・マネージャー状況

MQSC コマンドの DISPLAY QMSTATUS、または PCF コマンドの Inquire Queue Manager Status を実行すると、必要なログ・ファイルの詳細を含む、キュー・マネージャー情報が戻されます。MQSC コマンドについて詳しくは、[スクリプト \(MQSC\) コマンド](#)を、PCF コマンドについて詳しくは、[管理タスクの自動化](#)を参照してください。

### キュー・マネージャー・メッセージ

キュー・マネージャーは、次のような一対のメッセージを定期的に発行して、どのログ・ファイルが必要かを示します。

- メッセージ AMQ7467I は、キュー・マネージャーを再始動するために必要な最も古いログ・ファイルの名前を示しています。このログ・ファイルおよびそれより新しいログ・ファイルはすべて、キュー・マネージャーの再始動時に使用できる必要があります。
- メッセージ AMQ7468I は、メディア・リカバリーを実行するために必要な最も古いログ・ファイルの名前を示しています。

ログ・ファイルが "古い" か "新しい" かを判別するには、ファイル・システムによって適用される変更時刻ではなく、ログ・ファイル番号を使用します。

## 両方のタイプのロギングに適用される情報

キュー・マネージャーの再開に必要なログ・ファイル、つまりアクティブ・ログ・ファイルのみを、オンラインにする必要があります。非アクティブ・ログ・ファイルは、災害復旧用にテープなどのアーカイブ・メディアにコピーして、ログ・ディレクトリーから削除することができます。メディア回復に必要とされない非アクティブ・ログ・ファイルは、不必要なログ・ファイルと見なすことができます。不必要なログ・ファイルは、操作の対象でなくなった場合、削除することができます。

必要なログ・ファイルが見つからないと、オペレーター・メッセージ AMQ6767E が出されます。そのログ・ファイル、およびそれ以降のすべてのログ・ファイルをキュー・マネージャーが使用できるようにして、操作を再試行してください。

## ログ・エクステントを自動的にクリーンアップする - リニア・ロギングのみ

V 9.0.2

Multi

IBM MQ 9.0.2 以降には、リカバリーのために必要なくなったリニア・ログ・エクステントの自動管理を使用するオプションが用意されています。

自動管理をセットアップするには、qm.ini ファイルのログ・スタンザにある **LogManagement** 属性を使用するか、IBM MQ エクスプローラーを使用します。詳しくは、[116 ページの『キュー・マネージャーのログ』](#)を参照してください。

ログの操作の詳細については、**DISPLAY QMSTATUS** の **LOG** パラメーターを参照してください。また、ログの使用方法については、以下のコマンドを参照してください。

- [RESET QMGR](#)
- [SET LOG](#)

## メディア・イメージを自動的に作成する - リニア・ロギングのみ

V 9.0.2

IBM MQ 9.0.2以降、キュー・マネージャーが自動的にメディア・イメージを書き込むかどうかを制御するための全体的なスイッチが装備されます。デフォルトでは、このスイッチは設定されていません。

自動メディア・イメージを実行するかどうかと、そのプロセスの頻度は、以下のキュー・マネージャー属性を使用して制御できます。

### IMGSCHED

キュー・マネージャーがメディア・イメージを自動的に書き込むかどうか

### IMGINTVL

メディア・イメージを書き込む頻度(分)

### IMGLOGLN

オブジェクトの前のメディア・イメージより後に書き込まれたログのメガバイト数。

1日の中でワークロードが非常に重くなる重要な時間があり、システム・スループットが自動メディア・イメージ作成による影響を受けないようにする必要がある場合は、**IMGSCHED(MANUAL)**を設定して自動メディア・イメージ作成を一時的にオフにすることができます。

**IMGSCHED**の切り替えは、ワークロード中にいつでも実行できます。



**重要:**メディア・イメージを作成していない場合に、**MEDIALOG**が繰り返されることはありません。そのため、エクステントをアーカイブするか、十分なディスク・スペースを確保する必要があります。

また、次に挙げるその他のユーザー定義オブジェクトについても、メディア・イメージの自動と手動の切り替えを制御できます。

- 認証情報
- チャンネル
- クライアント 接続
- リスナー
- 名前リスト
- プロセス
- 別名キュー
- ローカル・キュー
- サービス
- トピック

オブジェクト・カタログやキュー・マネージャー・オブジェクトなどの内部システム・オブジェクトについては、必要に応じてキュー・マネージャーが自動的にメディア・イメージを書き込みます。

この属性について詳しくは、[ALTER QMGR](#)を参照してください。

ローカル・キューと永続動的キューに対してのみ、自動および手動のメディア・イメージを有効または無効にすることもできます。これを行うには、**IMGRCOVQ** キュー属性を使用します。

**IMGRCOVQ**属性の詳細については、[ALTER QUEUES](#)を参照してください。

### 注:

1. メディア・イメージは、リニア・ロギングを使用している場合にのみサポートされます。自動メディア・イメージを有効にした場合に、循環ロギングが使用されているときは、エラー・メッセージが出され、キュー・マネージャーの自動メディア・イメージ属性が無効になります。
2. 自動メディア・イメージを有効にしているが、ログの頻度(分またはメガバイト)を指定していない場合は、エラー・メッセージが出され、自動メディア・イメージは書き込まれなくなります。
3. **IMGSCHED(AUTO)**を設定した場合は、必要に応じて [rcdmqimg](#) を使用してメディア・イメージを手動で記録できます。

これを利用すると、企業の都合のよい時間 (例えば、システムが静止しているとき) に、メディア・イメージを作成することができます。自動メディア・イメージ作成では、このような手動によるメディア・イメージを考慮に入れます。つまり、手動メディア・イメージを作成した時点で、次の自動メディア・イメージ作成までの間隔とログの長さがリセットされます。

4. IBM MQ 9.0.2 では、キュー・マネージャーは、永続メッセージのみをメディア・イメージに書き込み、非永続メッセージは書き込みません。このため、IBM MQ 9.0.2 以降にマイグレーションすると、メディア・イメージのサイズを縮小できます。

## IMGLOGLN と IMGINTVL の設定値の決定

### V 9.0.2

**IMGINTVL** と **IMGLOGLN** の値は、キュー・マネージャーがメディア・イメージの記録に費やす時間をわずかに抑えるのに十分な大きさに設定する必要があります。ただし、次の目的を達成するのに十分なだけ、小さく設定する必要があります。

- 損傷したオブジェクトを適切な時間内に回復できるようにする。
- ログの大きさがディスクに見合っており、スペース不足にならないようにする。

**IMGLOGLN** を設定する場合は、**IMGLOGLN** をキュー上のデータ量の数倍、かつワークロードのデータ速度の数倍にすることをお勧めします。**IMGLOGLN** を大きくすると、キュー・マネージャーがメディア・イメージの記録に費やす時間が少なくなります。

同様に、**IMGINTVL** を設定する場合は、**IMGINTVL** を、キュー・マネージャーがメディア・イメージを記録するために費やす時間の数倍にすることをお勧めします。メディア・イメージを記録するのにかかる時間は、手動で記録して調べることができます。

**IMGLOGLN** と **IMGINTVL** が大きすぎると、損傷したオブジェクトをリカバリーするときに、最後のメディア・イメージ以後のすべての範囲を再生する必要があるため、非常に長時間かかることがあります。

損傷したオブジェクトの回復にかかる最大時間を許容範囲内に収めるには、**IMGLOGLN** と **IMGINTVL** の値を十分に小さくします。

**IMGINTVL** と **IMGLOGLN** を非常に大きい値にすると、メディア・イメージを記録する頻度が下がるため、ログが非常に大きくなります。



**重要:** ログ・ファイル・システムが完全に満杯になると、ワークロードがバックアウトされることになるため、このサイズのログをログ・ファイル・システムに無理なく格納できることを確認してください。

**IMGINTVL** と **IMGLOGLN** の両方を設定できます。両方を設定すると、ワークロードが重い間に自動メディア・イメージが定期的に作成される (**IMGLOGLN** によって制御) 一方で、ワークロードが非常に軽い時にも時折作成されます (**IMGINTVL** によって制御)。

**IMGINTVL** と **IMGLOGLN** は、自動メディア・イメージが作成される間隔とログ・データ長の目標値です。

これらの属性を、固定された最大値または最小値と見なすべきではありません。実際、キュー・マネージャーは、以下のような場合には、適切な機会を見計らって、早期に自動メディア・イメージをスケジュールすることがあります。

- キューが空の場合。パフォーマンスの観点から、メディア・イメージ作成の効率が最も高いと判断されません。
- しばらくの間メディア・イメージが記録されていない場合。

時折、自動メディア・イメージの作成間隔が **IMGINTVL** と **IMGLOGLN** の一方または両方より少し長くなることがあります。

キュー上のデータの量が **IMGLOGLN** に近づくと、メディア・イメージの間隔が **IMGLOGLN** より大きくなる場合があります。メディア・イメージの記録に **IMGINTVL** とほぼ同じ時間がかかる場合は、メディア・イメージの間隔が **IMGINTVL** より大きくなる場合があります。

これでは、キュー・マネージャーが大半の時間をメディア・イメージの記録に費やすことになるので、望ましくありません。

自動メディア・イメージ記録を使用している場合、キュー・マネージャーはオブジェクトごとおよびキューごとに個別にメディア・イメージを記録します。そのため、キュー・マネージャーは、イメージの間隔とログ長をオブジェクトごとに個別に追跡します。

時間の経過とともに、メディア・イメージの記録は、すべてのオブジェクトについて同時にメディア・イメージを記録する状態から、交互に記録する状態に変化していきます。この状態になると、メディア・イメージの記録がパフォーマンスに与える影響が分散します。これは、メディア・イメージの手動記録よりも自動記録を使用するほうが優れているもう1つの利点です。

## メディア・イメージを手動で作成する - リニア・ロギングのみ

V9.0.1

キューのメディア・イメージを記録するには、そのキューのすべての永続メッセージをログに書き込む必要があります。大量のメッセージ・データを含むキューの場合、ログに大量のデータが書き込まれるので、このプロセスが実行されている間、システムのパフォーマンスが影響を受ける可能性があります。

その他のオブジェクトのメディア・イメージの記録は、その他のオブジェクトのメディア・イメージにはユーザー・データが含まれていないため、比較的速く実行できます。

キューのメディア・イメージを記録するタイミングは、そのプロセスがピーク時のワークロードを邪魔しないように、注意深く検討する必要があります。

すべてのオブジェクトのメディア・イメージを定期的に記録して、メディア・リカバリーに必要な最も古いログ・エクステントを更新する必要があります。

キューのメディア・イメージを記録する良いタイミングは、キューが空のときです。その時点ではログに書き込まれるメッセージ・データがないからです。反対に、悪いタイミングは、キューが非常に深いときや、非常に大きなメッセージがキューにあるときです。

キューのメディア・イメージを記録するのに良いタイミングは、システムが静かなときで、悪いタイミングはピーク・ワークロードの最中です。例えば、常に午前0時のワークロードが静かな場合は、メディア・イメージを毎晩午前0時に記録することを決定できます。

各キューの記録を別々のタイミングにすると、パフォーマンスへの影響を分散できるため、記録の影響を減らすことができます。メディア・イメージを最後に記録してからの経過時間が長くなるほど、メディア・イメージを記録する重要性は高くなります。メディア・リカバリーに必要なログ・エクステント数が増えるからです。

**注:**メディア・リカバリーの実行中に、必要なログ・ファイルはすべてログ・ファイル・ディレクトリの中で同時に使用可能である必要があります。必要なログ・ファイルをすべて保持するためのディスク・スペースが不足しないようにするために、リカバリーするオブジェクトのメディア・イメージを必ず定期的にとるようにしてください。

例えば、キュー・マネージャー内のすべてのオブジェクトのメディア・イメージを作成するには、次の例で示されているように **rcdmqimg** コマンドを実行します。

### Windows Windows の場合

```
rcdmqimg -m QMNAME -t all *
```

### Linux UNIX and Linux の場合

```
rcdmqimg -m QMNAME -t all "*"
```

**rcdmqimg** コマンドの実行により、メディアのログ順序番号 (LSN) が順方向に移動します。ログ順序番号の詳細については、552 ページの『[dmpmqlog コマンドを使用したログの内容のダンプ](#)』を参照してください。**rcdmqimg** は自動的に実行されません。したがって、手動で実行するか、または作成した自動タスクから実行する必要があります。このコマンドの詳細については、[rcdmqimg](#) および [dmpmqlog](#) を参照してください。

**注:** **rcdmqimg** コマンドの実行時に、メッセージ AMQ7467 および AMQ7468 が出されることもあります。

## 部分メディア・イメージ

V9.0.2

IBM MQ メッセージは、近い将来にコンシュームされるデータに対してのみ使用することをお勧めします。各メッセージがキューに保持される時間が比較的短くなるからです。

一方で、データベースのように長期間データを保管するために IBM MQ メッセージを使用するのは望ましくありません。

また、キューを比較的浅い状態に保持することもお勧めします。逆に、メッセージをキューに長期間保持してキューを深い状態にするのは望ましくありません。

このようなガイドラインに従うと、キュー・マネージャーによるメディア・イメージの自動記録のパフォーマンスを最適化できます。

空のキューのメディア・イメージを記録するのは、(パフォーマンスの観点から)非常に効率の良い方法です。逆に、大量のデータが入ったキューのメディア・イメージを作成するのは、それらのデータをすべてメディア・イメージのログに書き込む必要があるため、非常に効率の悪い方法です。

直前にメッセージを書き込んだ浅いキューの場合、キュー・マネージャーはさらに最適化を行うことができます。

現在キューに入っているすべてのメッセージが最近書き込まれたものであるとすると、キュー・マネージャーは、すべてのメッセージが書き込まれた直前(リカバリー・ポイント)のメディア・イメージを記録することができます。つまり、空のキューのイメージを記録できます。このプロセスは、パフォーマンス上、非常に低コストです。

リカバリー・ポイントの時点でキューに存在していたすべてのメッセージが、その後でキューから取得された場合は、それらのメッセージがメディア・イメージに記録されている必要はありません。それらのメッセージは既にキューに存在しないからです。

これは、部分メディア・イメージと呼ばれます。その後、稀なケースですがキューのリカバリーが必要になった場合は、最後のメディア・イメージより後の、このキューに関連したすべてのログ・レコードを再生することにより、最近書き込まれたメッセージをすべて復元します。

仮に、リカバリー・ポイントの時点で少数のメッセージがキューにあり、現在もキューにある(そのため、部分メディア・イメージに記録する必要がある)としても、すべてのメッセージの完全なメディア・イメージを作成するよりも、小さな部分メディア・イメージを記録するほうが効率的です。

メッセージが短時間だけキューに滞在するようにすれば、メディア・イメージの自動記録のパフォーマンスが向上する可能性があります。

### 不必要なログ・ファイルの判別 - リニア・ロギングのみ

循環ロギングでは、ログ・ディレクトリーからデータを決して削除しないでください。リニア・ログ・ファイルを管理する場合、どのファイルを削除またはアーカイブできるかを確認することが重要です。以下の情報は、この決定を行う上で役立ちます。

ファイル・システムの変更日時によって"古い"ログ・ファイルの判別を行うことはしないでください。ログ・ファイル番号のみを使用してください。キュー・マネージャーによるログ・ファイルの使用は、複雑な規則(必要になる前に行われる事前割り振りやログ・ファイルの形式設定など)に従っています。変更日時を含むログ・ファイルを参照する際に、それらの変更日時によって相対的な経過時間を判別しようとすると、誤解が生じる場合があります。

必要な最も古いログ・ファイルを判別するために、以下の3つの場所を使用できます。

- DISPLAY QMSTATUS コマンド
- ロガー・イベント・メッセージ、そして最後に
- エラー・ログ・メッセージ

DISPLAY QMSTATUS コマンドで、それぞれの目的のために必要な最も古いログ・エクステンツを判別する方法は以下のとおりです。

- キュー・マネージャーを再始動するためには、コマンド DISPLAY QMSTATUS RECLLOG を発行します。
- メディア・リカバリーを実行するためには、コマンド DISPLAY QMSTATUS MEDIALOG を発行します。

- **V 9.0.2** アーカイブ通知の名前を判別し、コマンド `DISPLAY QMSTATUS ARCHLOG` を発行します。

**V 9.0.2** コマンド `RESET QMGR TYPE (REDUCELOG)` を発行することによって、循環ロギングを使用する際に 2 次ログのエクステンツの数を削減できます。

一般的に、ログ・ファイル番号が小さい方が、より古いログ・ファイルであることを意味します。ログ・ファイルの回転率が非常に高い場合 (10 年間で毎日 3000 個のログ・ファイルの配列) を除いて、9 999 999 での数の折り返しを考慮に入れる必要はありません。この場合、RECLOG 値よりも小さい番号のログ・ファイルをアーカイブし、RECLOG 値と MEDIALOG 値の両方よりも小さい番号のログ・ファイルを削除することができます。



**重要:** ログ・ファイルは循環するので、9 999 999 の次の番号はゼロになります。

#### ログ・ファイルの位置

ログ・ファイルの位置を選択する際には、ディスク・スペースの不足のために IBM MQ が新しいログをフォーマットできないと、操作に重大な影響が及ぶので注意してください。

循環ログを使用している場合は、少なくとも構成された 1 次ログ・ファイル用の十分なスペースをドライブ上に確保してください。また、ログが大きくなる場合に必要となる少なくとも 1 つの 2 次ログ・ファイル用のスペースも残しておいてください。

リニア・ログを使用している場合は、十分にスペースを考慮しておく必要があります。というのは、ログが使用するスペースは、データがログに記録されるにつれて絶えず増大するからです。

ログ・ファイルは、キュー・マネージャー・データとは別のディスク・ドライブに配置する必要があります。

このデバイスではデータ保全性が非常に重要です - 組み込みの冗長性を許可する必要があります。

また、ミラーリング配置された複数のディスク・ドライブにログ・ファイルを置くこともできます。これは、ログを含んでいるドライブに障害が起きた場合に保護として機能します。ミラーリングがない場合にはおそらく、IBM MQ システムの最新のバックアップを使用する以外に方法はありません。

## ログをリカバリーに使用する

ログの情報を障害からのリカバリーに役立てることができます。

損傷を受けるデータ方法はいくつかあります。IBM MQ は、以下の方法で回復できる:

- データ・オブジェクトの損傷
- システムの停電
- 通信障害

このセクションでは、これらの問題からリカバリーするのにログがどのように使用されるかを調べます。

### 停電や通信障害からのリカバリー

IBM MQ は、通信障害や停電による障害からリカバリーすることができます。他の種類の問題 (不注意によるファイルの削除など) からのリカバリーも可能です。

通信障害の場合、持続メッセージは、受信側のアプリケーションによって取り除かれるまでキューに残っています。メッセージが伝送中である場合は、正常に伝送できるまで、メッセージは伝送キューにとどまっています。通信障害からのリカバリーは、通常の場合、障害を起こしたリンクを使用しているチャンネルを再始動することで可能です。

停電の場合には、キュー・マネージャーが再始動されたときに、IBM MQ が障害が起きたときのコミットした状態にキューを復元します。これで、持続メッセージが失われることはありません。非持続メッセージは廃棄されます。非持続メッセージは、IBM MQ が不意に停止した場合には残存しません。

### 損傷オブジェクトをリカバリーする

IBM MQ オブジェクトが使用不能になる場合があります。例えば、不注意で損傷した場合などです。そのようなときには、システム全体かまたはその一部をリカバリーする必要があります。どのような処置が必

要かは、損傷が検出された時期、選択されているログ方法がメディア・リカバリーをサポートするかどうか、およびどのオブジェクトが損傷したかによって異なります。

## メディア・リカバリー

**V 9.0.2** IBM MQ 9.0.2 から、リニア・ロギングのキュー・マネージャーで、メディア・イメージを記録できるのはリカバリー可能なオブジェクトのみです。例えば、**IMGRCOVO** および **IMGRCOVQ** オプションを考慮する必要があります。

**V 9.0.2** 同様に、リニア・ロギングのキュー・マネージャーのメディア・イメージからは、メディア・リカバリー可能と定義されたオブジェクトのサブセットのみをリカバリーすることができます。メディア・リカバリー可能と定義されていないオブジェクトが損傷した場合、そのオブジェクト用のオプションは、循環ロギングのキュー・マネージャー用のオプションと同じです。

メディア・リカバリーは、リニア・ログに記録されている情報からオブジェクトを再作成します。例えば、オブジェクト・ファイルが不注意で削除された場合、あるいはその他の理由で使用不能になった場合、それをメディア・リカバリーによって再作成できます。オブジェクトのメディア・リカバリーに必要なログ内の情報のことを、メディア・イメージと呼びます。

メディア・イメージは、オブジェクトのイメージを含んでいる一連のログ・レコードであり、そこからオブジェクト自体を再作成できます。

オブジェクトを再作成するのに必要な最初のログ・レコードは、メディア・リカバリー・レコードとして知られています。これは、オブジェクトの最新のメディア・イメージの開始点です。各オブジェクトのメディア・リカバリー・レコードは、チェックポイント時に記録された情報の一部です。

メディア・イメージからオブジェクトを再作成する際には、最新のイメージが取られた時以降にオブジェクトに対して行われた更新を記述しているログ・ファイルを再生する必要もあります。

例えば、持続メッセージがキューに書き込まれる前にとられたキュー・オブジェクトのイメージを持っているローカル・キューを考えてみます。そのオブジェクトの最新のイメージを再作成するには、イメージ自体を再生するだけでなく、キューにメッセージが書き込まれたことを記録しているログ項目を再生することも必要です。

オブジェクトが作成されるときに、作成されるログ・レコードには、そのオブジェクトを完全に再作成するのに十分な情報が入れられます。それらのレコードは、オブジェクトの最初のメディア・イメージを構成することになります。続いて、シャットダウンするたびに、キュー・マネージャーは自動的に以下のようなメディア・イメージを記録します。

- すべてのプロセス・オブジェクトおよび非ローカル・キューのイメージ
- 空のローカル・キューのイメージ

メディア・イメージは、`rcdmqimg` で説明されている `rcdmqimg` コマンドを使用して、手動で記録することもできます。このコマンドは、IBM MQ オブジェクトのメディア・イメージを記録します。

**V 9.0.2** **IMGSCHED(AUTO)** が設定されている場合、キュー・マネージャーはメディア・イメージを自動的に記録します。詳しくは、**ALTER QMGR** を参照して、**IMGINTVL** および **INGLOGLN** に関する情報を確認してください。

メディア・イメージの書き込みが行われた後は、損傷しているオブジェクトを再作成するために必要なのは、そのメディア・イメージを持つログと、この時点以後に作成されたすべてのログだけです。メディア・イメージを作成する利点は、利用できるフリー・ストレージの容量およびログ・ファイルが作成される速度などの要因によって異なります。

## メディア・イメージからのリカバリー

キュー・マネージャーは、キュー・マネージャーの始動時に、そのメディア・イメージから一部のオブジェクトを自動的にリカバリーします。キュー・マネージャーの最後のシャットダウン時に未完了だったランザクシオンにキューが関与していて、再始動処理時にそのキューが破損または損傷していることが分かると、キュー・マネージャーはそのキューを自動的にリカバリーします。

その他のオブジェクトについては、**rcrmqobj** コマンドを使用して手動でリカバリーする必要があります。このコマンドは、IBM MQ オブジェクトの再作成のためにログのレコードを再生するものです。オブジェクトは、ログの中にある最新のイメージから再作成されます。また、その際に、イメージが保管されてから再作成コマンドが出されるまでの間の 適当なログ・イベントもすべて再作成に使用されます。IBM MQ オブジェクトが損傷した場合には、実行できる有効な処置は、オブジェクトを削除するか、あるいはこの方法でオブジェクトを再作成することです。非持続メッセージは、今述べた方法ではリカバリーできません。

**rcrmqobj** コマンドの詳細については、[rcrmqobj](#) を参照してください。

オブジェクトのメディア・リカバリーを試みる際には、メディア・リカバリー・レコードを含んでいるログ・ファイル、およびそれ以降のすべてのログ・ファイルがログ・ファイル・ディレクトリーの中であって、使用可能状態になっていなければなりません。必要なファイルが見つからない場合には、オペレーター・メッセージ AMQ6767 が出され、メディア・リカバリー操作は失敗します。再作成するオブジェクトのメディア・イメージを定期的に取りっていないと、オブジェクトを再作成するために必要なログ・ファイルすべてを保持するためのディスク・スペースが、不足する可能性があります。

## 存在するオブジェクト・ファイル

V9.0.1

キュー・マネージャーは **runmqsc** に定義されたオブジェクトの属性をディスク上のファイルに保管します。これらのオブジェクト・ファイルはキュー・マネージャーのデータ・ディレクトリーの下にあるサブディレクトリーにあります。

例えば、UNIX および Linux プラットフォームでは、チャンネルは `/var/mqm/qmgrs/qmgr/channel` に保管されます。

これらのオブジェクト・ファイル内のデータは、オブジェクトのメディア・イメージです。これらのオブジェクト・ファイルが削除されたり破損したりすると、ファイル内に保管されたオブジェクトが破損します。リニア・ロギングのキュー・マネージャーを使用する場合は、**rcrmqobj** コマンドを使用して、破損したオブジェクトをログからリカバリーできます。

ほとんどのオブジェクト・ファイルには、そのオブジェクトの属性のみが含まれています。そのため、チャンネル・ファイルにはチャンネルの属性が含まれています。ただし、次の例外があります。

- カタログ

オブジェクト・カタログには、すべてのタイプのすべてのオブジェクトがカタログされており、`qmanager/QMQMOBJCAT` に保管されます。

- 同期ファイル

同期ファイルには、すべてのチャンネルに関連する内部状態データが含まれています。

- キュー

キュー・ファイルには、そのキューのメッセージとキューの属性の両方が含まれています。

カタログ・オブジェクトも同期ファイル・オブジェクトも **runmqsc** または IBM MQ エクスプローラーで公開されないことに注意してください。

カタログおよびキュー・マネージャーは、記録はできますがリカバリーはできません。これらのオブジェクトが破損した場合、キュー・マネージャーは先回りして終了し、再始動時にこれらのオブジェクトが自動的にリカバリーされます。

サブスクリプションは、永続サブスクリプションがシステム・キューに保管されているため、記録またはリカバリーするオブジェクトにリストされません。永続サブスクリプションを記録またはリカバリーするには、代わりに `SYSTEM.DURABLE.SUBSCRIBER.QUEUE` を記録またはリカバリーします。

## 始動時に損傷オブジェクトをリカバリーする

始動時に損傷オブジェクトを見つけた場合、キュー・マネージャーが行うアクションは、オブジェクトのタイプや、キュー・マネージャーがメディア・リカバリーをサポートするように構成されているかどうかによって異なります。

キュー・マネージャーのオブジェクトが損傷している場合、キュー・マネージャーはそのオブジェクトをリカバリーできなければ、開始できません。キュー・マネージャーがリニア・ログと共に構成されていて、メディア・リカバリーをサポートしている場合には、IBM MQ は、自動的にメディア・イメージからキュー・マネージャーのオブジェクトを再作成しようとします。選択したログの方法がメディア・リカバリーをサポートしていない場合には、キュー・マネージャーのバックアップを復元するか、あるいはキュー・マネージャーを削除することができます。

キュー・マネージャーが停止したときにアクティブ状態のトランザクションがあると、それらのトランザクションの中で書き込まれたり読み取られたりした持続メッセージ(コミットされていないもの)のあるローカル・キューも、キュー・マネージャーを正常に始動するためには必要です。キュー・マネージャーがメディア・リカバリーをサポートしている場合で、これらのキューのいずれかが損傷していることが分かった場合は、キュー・マネージャーは、メディア・イメージからそれらのキューを自動的に再作成しようとします。それらのキューのいずれかがリカバリーできないときは、IBM MQ は始動できません。

メディア・リカバリーをサポートしないキュー・マネージャーの始動処理時に、コミットされていないメッセージを含んでいる損傷ローカル・キューが検出された場合、それらのキューには損傷オブジェクトのマークが付けられ、そのキューにあるコミットされていないメッセージは無視されます。その理由は、そのようなキュー・マネージャーでは、損傷オブジェクトのメディア・リカバリーを実行できず、そのオブジェクトを削除するという処置しか行えないためです。メッセージ AMQ7472 が出て、損傷が通知されます。

## その他の場合の損傷オブジェクトのリカバリー

オブジェクトのメディア・リカバリーは、始動時にのみ行われます。その他の場合に、オブジェクトの損傷が検出されたときには、オペレーター・メッセージ AMQ7472 が出され、そのオブジェクトを使用するほとんどの操作は失敗します。キュー・マネージャーが始動した後のあらゆる時点でキュー・マネージャーのオブジェクトが損傷した場合には、キュー・マネージャーは優先シャットダウンを実行します。オブジェクトが損傷している場合には、それを削除することができます。あるいは、キュー・マネージャーがリニア・ログを使用している場合には、`rcrmqobj` コマンドを使用して、メディア・イメージからそのオブジェクトのリカバリーを試みてください(詳細については、`rcrmqobj` を参照してください)。

**V 9.0.2** キュー(または他のオブジェクト)が損傷を受けた場合、**MEDIALOG** は進行しません。これは、**MEDIALOG** がメディア・リカバリーに必要な最も古いエクステンツであるためです。ワークロードが続いている場合、**CURRLOG** は引き続き進行しているので、新しいエクステンツが書き込まれます。構成(LogManagement 設定を含む)によっては、ログ・ファイル・システムが満杯になり始める可能性があります。ログ・ファイル・システムが完全にフルになると、トランザクションはロールバックされて、キュー・マネージャーが突然終了する可能性があります。そのため、キューが損傷すると、キュー・マネージャーが終了する前に対処するには限られた時間しかない可能性があります。どれだけの時間があるかは、キュー・マネージャーがワークロードによる新しいエクステンツを書き込んでいる速度、およびログ・ファイル・システムに存在する空きスペースのサイズによって異なります。

**V 9.0.2** 手動のログ管理を使用している場合には、再始動リカバリーには必要のないエクステンツをアーカイブした後に、それらがメディア・リカバリー用にまだ必要であるにもかかわらず、ログ・ファイル・システムから削除している可能性があります。これは、それらが必要なときにアーカイブから復元できる限りは許容されることです。このポリシーでは、キューが損傷して **MEDIALOG** が進行を停止しても、ログ・ファイル・システムがフルになることはありません。ただし、再始動リカバリーとメディア・リカバリーのいずれにも必要のないエクステンツをアーカイブして削除するだけであれば、キューが損傷した場合に、ログ・ファイル・システムはフルになっていきます。

**V 9.0.2** 自動またはアーカイブ・ログ管理を使用している場合、**SET LOG ARCHIVED** を使用してアーカイブしてからキュー・マネージャーに通知した場合でも、キュー・マネージャーはメディア・リカバリーに必要なエクステンツを再利用しません。その結果、キューが損傷した場合には、ログ・ファイル・システムがフルになっていきます。

**V 9.0.2** キューが損傷すると、OBJECT DAMAGED FFDC が書き込まれて、**MEDIALOG** は進行を停止します。損傷したオブジェクトは、FFDC から識別できます。または、`runmqsc` で状態を表示したときに、最も古い **MEDIALOG** を持つオブジェクトとして識別することもできます。

ログ・ファイル・システムがフルになっていく場合、ログ・ファイル・システムがフルになってワークロードがバックアウトする心配が生じます。オブジェクトをリカバリーすることにより、またはワークロードを静止させることにより、そのような事態を回避できることがあります。

## IBM MQ ログ・ファイルの保護

キュー・マネージャーの実行中はログ・ファイルに触らないでください。リカバリーできなくなる場合があります。誤って変更しないように、スーパーユーザー権限または mqm 権限を使用してログ・ファイルを保護してください。

IBM MQ キュー・マネージャーの実行中は、アクティブ・ログ・ファイルを手動で削除しないでください。キュー・マネージャーの再始動に必要なログ・ファイルをユーザーが誤って削除した場合でも、IBM MQ はエラー・メッセージを発行しないで、持続メッセージが入っているデータの処理を続けます。この場合、キュー・マネージャーは正常にシャットダウンしますが、再始動できない場合があります。そして、メッセージのリカバリーも不可能になります。

アクティブなキュー・マネージャーで使用中のログを削除する権限を持つユーザーは、キュー・マネージャーのその他の重要なリソース(キュー・ファイル、オブジェクト・カタログ、IBM MQ 実行可能ファイルなど)を削除する権限も持っています。したがって、そのようなユーザーは、おそらくは経験不足から、IBM MQ が防ぎきれない方法で実行中または休止中のキュー・マネージャーに損傷を与える場合があります。

スーパーユーザー権限または mqm 権限を付与する場合には、十分に注意してください。

## dmpmqlog コマンドを使用したログの内容のダンプ

dmpmqlog コマンドを使用して、キュー・マネージャーのログの内容をダンプする方法。

キュー・マネージャーのログの内容をダンプするには、dmpmqlog コマンドを使用します。デフォルトでは、アクティブなログ・レコードがすべてダンプされます。つまり、このコマンドは、ログのヘッドからダンプを開始します(通常、最後に完了したチェックポイントの開始)。

通常、キュー・マネージャーが実行されていない場合にのみログをダンプできます。終了中にキュー・マネージャーはチェックポイントを取るため、通常ログのアクティブ部分には、少数のログ・レコードが入ります。ただし、次のオプションのいずれかを使用すると、dmpmqlog コマンドを使用してログ・レコードをより多くダンプし、ダンプの開始位置を変更することができます。

- ログのベースからダンプを開始します。ログのベースとは、ログのヘッドを含むログ・ファイルの中にある最初のログ・レコードです。この場合、ダンプされるデータ追加量は、ログのヘッドがログ・ファイルの中で位置付けされる場所によって異なります。ログ・ファイルの最初に近い場合には、少量の追加データしかダンプされません。ログ・ファイルの終わりに近い場合には、極めて多量のデータがダンプされます。
- ダンプの開始位置を個々のログ・レコードとして指定します。各ログ・レコードは、固有のログ順序番号 (LSN) により識別されます。循環ログの場合、この開始ログ・レコードをログのベースの前に置くことはできません。この制限は、リニア・ログには適用されません。コマンドの実行前に、非アクティブ・ログ・ファイルを復元する必要があるかもしれません。開始位置として、前の dmpmqlog 出力から取った、有効な LSN を指定することが必要です。

例えば、リニア・ロギングでは、最後の dmpmqlog 出力から nextlsn を指定します。すると、次の nextlsn が Log File Header に表示され、次のログ・レコードの LSN が書き込まれることを示しています。ログが最後にダンプされた後に、書き込まれるログ・レコードをすべてフォーマットするため、開始位置としてこれを使用することができます。

- **リニア・ログのみの場合**、指定のログ・ファイル・エクステンツからログ・レコードのフォーマット化を開始するよう dmpmqlog に指示できます。この場合、dmpmqlog により、このログ・ファイル、およびアクティブ・ログ・ファイルと同じディレクトリーにある次のファイルが検索されます。このオプションは循環ログには適用されません。dmpmqlog がログのベース前のログ・レコードにアクセスできないためです。

dmpmqlog コマンドからの出力は、ログ・ファイルのヘッダー および一連のフォーマット済みログ・レコードです。キュー・マネージャーは、いくつかのログ・レコードを使用して、データに対する変更を記録します。

フォーマットされる一部の情報は、内部的に使用されるものです。次のリストには、最も役立つログ・レコードを示しています。

#### ログ・ファイルのヘッダー

各ログには単一のログ・ファイルのヘッダーがあり、dmpmqlog コマンドにより必ず最初にフォーマットされます。これには、次のようなフィールドがあります。

<i>logactive</i>	1 次ログ・エクステントの数
<i>loginactive</i>	2 次ログ・エクステントの数
<i>logsize</i>	エクステントあたり 4 KB ページ数
<i>baselsn</i>	ログのヘッドを含むログ・エクステントの中の最初の LSN
<i>nextlsn</i>	書き込まれる次のログ・レコードの LSN
<i>headlsn</i>	ログのヘッドでのログ・レコードの LSN
<i>tailsn</i>	ログのテール位置を識別する LSN
<i>hflag1</i>	ログが CIRCULAR であるか LOG RETAIN (リニア) であるかどうか
拡張ヘッダー ID	ログのヘッドを含むログ・エクステント

#### ログ・レコードのヘッダー

ログ内の各ログ・レコードには以下の情報を含む固定のヘッダーがあります。

<i>LSN</i>	ログの順序番号
<i>LogRecdType</i>	ログ・レコードのタイプ
<i>XTranid</i>	このログ・レコード (ある場合) と関連したトランザクション ID MQI の <i>TranType</i> は IBM MQ トランザクションのみを示しています。 XA の <i>TranType</i> は、他のリソース・マネージャーに関連するトランザクションを示しています。同じ作業単位内に含まれる更新には同じ <i>XTranid</i> があります。
<i>QueueName</i>	このログ・レコード (ある場合) と関連したキュー
<i>Qid</i>	キュー用の固有の内部 ID
<i>PrevLSN</i>	同じトランザクション内の前のログ・レコード (ある場合) の LSN

#### キュー・マネージャーの始動

これは、キュー・マネージャーを始動した記録を取ります。

開始日	キュー・マネージャーを始動した日付
<i>startTime</i>	キュー・マネージャーを始動した時刻

#### キュー・マネージャーの停止

これは、キュー・マネージャーを停止した記録を取ります。

停止日	キュー・マネージャーを停止した日付
停止時刻	キュー・マネージャーを停止した時刻
<i>ForceFlag</i>	使用した終了のタイプ

#### チェックポイントの開始

これはキュー・マネージャーのチェックポイントの開始を示しています。

## チェックポイントの終了

これはキュー・マネージャーのチェックポイントの終了を示しています。

**ChkPtLSN**                      このチェックポイントを開始したログ・レコードの LSN

## メッセージの書き込み

これはキューに書き込んだ持続メッセージの記録を取ります。メッセージが同期点で書き込まれた場合、ログ・レコードのヘッダーには非ヌル文字 *XTranid* が入ります。レコードの残りの部分には以下のものが入ります。

**MapIndex**                      キュー上のメッセージの ID。これは、キューからこのメッセージを得るのに使用した該当する MQGET を突き合わせる場合に使用できます。この場合、同じ *QueueName* および *MapIndex* が含まれている次の *Get Message* ログ・レコードを検出できます。この時点で、キューにメッセージを続けて書き込むために *MapIndex* ID を再使用できます。

データ                              このログ・レコードには 16 進ダンプで、さまざまな内部データと、それに続くメッセージ記述子 (目印 MD) の表現と、それに続くメッセージ・データ本体が入っています。

## 部分の書き込み

単一のログ・レコードとしては長すぎる持続メッセージは、複数の *Put Part* ログ・レコードと、その後続く単一の *Put Message* レコードとして記録されます。複数の *Put Part* レコードがある場合は、*PrevLSN* フィールドによって、*Put Part* レコードと最後の *Put Message* レコードが連結されます。

データ                              前のログ・レコードが終了した場所のメッセージ・データが続きます。

## メッセージの読み取り

持続メッセージの読み取りのみが記録されます。メッセージが同期点で読み取られた場合、ログ・レコードのヘッダーには非ヌル文字 *XTranid* が入ります。レコードの残りの部分には以下のものが入ります。

**MapIndex**                      キューから得られたメッセージを識別します。同じ *QueueName* および *MapIndex* を含む最新の *Put Message* ログ・レコードは得られたメッセージを識別します。

**QPriority**                      キューから得られたメッセージの優先順位。

## トランザクションの開始

新規トランザクションの開始を示しています。MQI の *TranType* は IBM MQ トランザクションのみを示しています。XA の *TranType* は、他のリソース・マネージャーに関連するトランザクションを示しています。このトランザクションが行ったすべての更新には同じ *XTranid* があります。

## トランザクションの準備

キュー・マネージャーは、指定した *XTranid* と関連した更新をコミットする準備が整っていることを示しています。このログ・レコードは、他のリソース・マネージャーに関連する 2 フェーズ・コミットの一部として書き込まれます。

## トランザクションのコミット

トランザクションが行ったすべての更新をキュー・マネージャーがコミットしたことを示しています。

## トランザクションのロールバック

これは、キュー・マネージャーがトランザクションをロールバックしようとしていることを表しています。

## トランザクションの終了

これはロールバック・トランザクションの終了を表しています。

## トランザクション表

このレコードは同期点の最中に書き込まれます。これは持続更新を行ったトランザクションそれぞれの状態を記録します。トランザクションごとに、次の情報が記録されます。

<i>XTranid</i>	トランザクション ID。
<i>FirstLSN</i>	トランザクションに関連した最初のログ・レコードの LSN
<i>LastLSN</i>	トランザクションに関連した最後のログ・レコードの LSN

#### トランザクションに関連するプログラム

このログ・レコードは、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントによって書き込まれます。これは、トランザクションに関連する外部リソース・マネージャーを記録します。関連するプログラムごとに次の情報が記録されます。

<i>RMName</i>	リソース・マネージャーの名前
<i>RMID</i>	リソース・マネージャー ID。これは後続の <i>Transaction Prepared</i> ログ・レコードにもログインします。このレコードはリソース・マネージャーが関連しているグローバル・トランザクションを記録するものではありません。
<i>SwitchFile</i>	このリソース・マネージャー用のスイッチ・ロード・ファイル
<i>XAOpenString</i>	このリソース・マネージャー用の XA オープン・ストリング
<i>XACloseString</i>	このリソース・マネージャー用の XA クローズ・ストリング

#### 準備済みトランザクション

このログ・レコードは、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントによって書き込まれます。指定したグローバル・トランザクションが正常に作成されたことを示しています。関連するリソース・マネージャーのそれぞれにコミットが指示されます。準備済みの各リソース・マネージャーの *RMID* は、ログ・レコードに記録されます。キュー・マネージャー自体がトランザクションに関連している場合、*RMID* がゼロである *Participant Entry* が示されます。

#### 忘れられたトランザクション

このログ・レコードは、キュー・マネージャーの XA トランザクション・マネージャーのコンポーネントによって書き込まれます。コミットの決定が関連する各プログラムに出された場合に *Transaction Prepared* ログ・レコードの後に続くものです。

#### キューの除去

これは、キュー上のすべてのメッセージが、例えば MQSC コマンド CLEAR QUEUE を使用して除去されたという事実の記録を取ります。

#### キューの属性

これは、キューの属性の初期化または変更の記録を取ります。

#### オブジェクトの作成

これは、IBM MQ オブジェクトの作成を記録します。

<i>ObjName</i>	作成したオブジェクトの名前
<i>UserId</i>	作成を実行するユーザー ID

#### オブジェクトの削除

これは、IBM MQ オブジェクトの削除を記録します。

<i>ObjName</i>	削除したオブジェクトの名前
----------------	---------------

## IBM MQ キュー・マネージャー・データのバックアップと復元

ハードウェア障害に起因する破壊からキュー・マネージャーを保護できます。そのためには、キュー・マネージャーとキュー・マネージャー・データをバックアップするか、キュー・マネージャーの構成のみをバックアップするか、バックアップ・キュー・マネージャーを使用します。

## このタスクについて

ハードウェア障害に起因する破壊からキュー・マネージャーを保護する対策を定期的に講じることができ  
ます。キュー・マネージャーには、次に示す3種類の保護方法があります。

### キュー・マネージャー・データのバックアップ

ハードウェア障害が発生した場合、キュー・マネージャーは強制的に停止されることがあります。ハ  
ードウェア障害によってキュー・マネージャーのログ・データが失われた場合、キュー・マネージャー  
が再始動できないことがあります。キュー・マネージャー・データをバックアップしておくこと、失わ  
れたキュー・マネージャー・データの一部またはすべてを回復できる可能性があります。

一般に、ハードウェア障害が発生したために、リカバリー・ログの整合性が失われたとしても、キュー  
・マネージャー・データのバックアップの頻度を高くするほど、失われるデータ量は少なくなります。

キュー・マネージャー・データをバックアップするには、キュー・マネージャーが実行中であってはな  
りません。

### キュー・マネージャー構成のみのバックアップ

ハードウェア障害が発生した場合、キュー・マネージャーは強制的に停止されることがあります。ハ  
ードウェア障害によってキュー・マネージャー構成とログ・データの両方が失われた場合、キュー・マ  
ネージャーは再始動したり、ログからリカバリーしたりできなくなります。キュー・マネージャー構  
成をバックアップした場合は、保存されている定義から、キュー・マネージャーとそのすべてのオブジ  
ェクトを再作成できます。

キュー・マネージャー構成をバックアップするには、キュー・マネージャーが実行中でなければなりま  
せん。

### バックアップ・キュー・マネージャーの使用

重大なハードウェア障害が起きた場合は、キュー・マネージャーをリカバリーできない可能性がありま  
す。この状態では、リカバリー不能なキュー・マネージャーに専用のバックアップ・キュー・マネー  
ジャーが存在すれば、バックアップ・キュー・マネージャーをリカバリー不能なキュー・マネージャー  
の代わりに活動化することができます。定期的に更新していた場合は、バックアップ・キュー・マネ  
ージャーのログに、リカバリー不能なキュー・マネージャーからの最新の完全なログが含まれたログ・  
データがある可能性があります。

バックアップ・キュー・マネージャーは、既存のキュー・マネージャーの実行中に更新できます。

## 手順

- キュー・マネージャー・データをバックアップおよびリストアするには、以下を参照してください。
  - [556 ページの『キュー・マネージャー・データのバックアップ』](#)。
  - [557 ページの『キュー・マネージャー・データの復元』](#)。
- キュー・マネージャー構成をバックアップおよびリストアするには、以下を参照してください。
  - [558 ページの『キュー・マネージャー構成のバックアップ』](#)
  - [559 ページの『キュー・マネージャー構成の復元』](#)
- バックアップ・キュー・マネージャーを作成、更新、および始動するには、[560 ページの『バックア  
ップ・キュー・マネージャーの使用』](#)を参照してください。

## キュー・マネージャー・データのバックアップ

キュー・マネージャーのバックアップを行うと、ハードウェア・エラーによって生じる可能性のあるデー  
タの消失から保護することができます。

## 始める前に

キュー・マネージャーのバックアップを開始する前に、キュー・マネージャーが実行されていないことを  
確認してください。実行中のキュー・マネージャーのバックアップを取ろうとすると、ファイルがコピー  
されるときに更新が進行中であるため、バックアップが一貫していない可能性があります。可能な場合は、

**endmqm -w** コマンド (待機シャットダウン) を実行してキュー・マネージャーを停止します。これが失敗した場合にのみ、**endmqm -i** コマンドを使用します (即時シャットダウン)。

## このタスクについて

キュー・マネージャーのデータのバックアップ・コピーを取るには、次のタスクを実行します。

### 手順

1. 構成ファイル中の情報を使用して、キュー・マネージャーのデータとログ・ファイルが置かれているディレクトリーを検索します。

詳しくは、[85 ページの『IBM MQ およびキュー・マネージャー構成情報の変更』](#)を参照してください。

**注:** IBM MQ を使用しているプラットフォームとの互換性を持たせるために、ディレクトリーに表示される名前が変換されています。名前の変換について詳しくは、[IBM MQ ファイル名についての理解](#)を参照してください。

2. キュー・マネージャーのデータおよびログ・ファイルのディレクトリーすべてを、すべてのサブディレクトリーと一緒に、コピーしてください。

必ずすべてのファイルをコピーしてください。特に、[529 ページの『ログの概要』](#)で説明されているログ制御ファイルと、[187 ページの『初期設定および構成ファイル』](#)で説明されている構成ファイルのコピーを忘れないでください。一部のディレクトリーは空である場合がありますが、後日バックアップを復元するにはすべて必要になります。

循環ログでは、キュー・マネージャー・データとログ・ファイル・ディレクトリーと一緒にバックアップしてください。これにより、整合したキュー・マネージャー・データとログの集合を復元できます。

リニア・ロギングの場合は、キュー・マネージャー・データとログ・ファイル・ディレクトリーと一緒にバックアップしてください。キュー・マネージャーのデータ・ファイルに対応する一連の全ログ・ファイルが使用可能である場合に、キュー・マネージャーのデータ・ファイルのみを復元することが可能です。

3. ファイルの所有権も存続させます。

 IBM MQ for UNIX および Linux システムでは、**tar** コマンドを使用してこれを行うことができます。(2 GB より大きいキューがある場合、**tar** コマンドは使用できません。詳しくは、[大規模キューの使用可能化](#)を参照してください。

**注:** IBM WebSphere MQ 7.5 以降にアップグレードする場合、**qm.ini** ファイルおよびレジストリー項目のバックアップを必ず取っておいてください。キュー・マネージャーの情報は **qm.ini** ファイルに保管され、前のバージョンの IBM MQ に戻すために使用できます。

### 関連タスク

[12 ページの『キュー・マネージャーの停止』](#)

**endmqm** コマンドを使用して、キュー・マネージャーを停止できます。このコマンドでは、制御または静止状態でのシャットダウン、即時シャットダウン、およびプリエンティブ・シャットダウンの3つの方法でキュー・マネージャーを停止できます。または、Windows および Linux では、IBM MQ Explorer を使用してキュー・マネージャーを停止できます。

[11 ページの『キュー・マネージャーの作成後の構成ファイルのバックアップ』](#)

IBM MQ 構成情報は、UNIX, Linux, and Windows 上の構成ファイルに保管されています。キュー・マネージャーの作成後、構成ファイルをバックアップしてください。その後、問題の原因となる別のキュー・マネージャーを作成した場合には、問題の原因を取り除いたときにバックアップを復元させることができます。

## キュー・マネージャー・データの復元

キュー・マネージャーのデータのバックアップを復元するには、以下のステップに従います。

## 始める前に

バックアップを開始する前に、キュー・マネージャーが実行されていないことを確認してください。

クラスター内のキュー・マネージャーのバックアップをリストアする場合の詳細については、[317 ページの『クラスター・キュー・マネージャーのリカバリー』](#) および「[クラスタリング: 可用性、マルチインスタンス、および災害復旧](#)」を参照してください。

注: IBM WebSphere MQ 7.5 以降にアップグレードする場合、**.ini** ファイルおよびレジストリー項目のバックアップを必ず取っておいてください。キュー・マネージャーの情報は **.ini** ファイルに保管され、前のバージョンの IBM MQ に戻すために使用できます。

## 手順

1. 構成ファイル中の情報を使用して、キュー・マネージャーのデータとログ・ファイルが置かれているディレクトリーを探します。
2. バックアップ・データの格納先となるディレクトリーを空にしてください。
3. バックアップされたキュー・マネージャーのデータおよびログ・ファイルを、正しい場所にコピーします。

ログ・ファイルだけでなく、ログ制御ファイルもあることを確認してください。

循環ログでは、キュー・マネージャー・データとログ・ファイル・ディレクトリーを一緒にバックアップしてください。これにより、整合したキュー・マネージャー・データとログの集合を復元できます。

リニア・ロギングの場合は、キュー・マネージャー・データとログ・ファイル・ディレクトリーを一緒にバックアップしてください。キュー・マネージャーのデータ・ファイルに対応する一連の全ログ・ファイルが使用可能である場合に、キュー・マネージャーのデータ・ファイルのみを復元することが可能です。

4. 構成情報ファイルを更新します。

IBM MQ が復元データの検索にあたって正しい場所を調べられるように、IBM MQ およびキュー・マネージャーの構成ファイルが一貫しているかどうかを確認してください。

5. できあがったディレクトリー構造をチェックして、必要なディレクトリーがすべて入っていることを確認してください。

IBM MQ のディレクトリーとサブディレクトリーについては、[Windows システムでのディレクトリー構造](#) および [UNIX and Linux システムでのディレクトリー](#) の内容を参照してください。

## タスクの結果

データが正しくバックアップされ復元されていれば、キュー・マネージャーは始動します。

**Multi**

### キュー・マネージャー構成のバックアップ

キュー・マネージャー構成をバックアップすると、ハードウェア障害が原因でキュー・マネージャー構成とログ・データの両方が失われ、キュー・マネージャーを再始動したりログからリカバリーしたりできない場合に、キュー・マネージャーをその定義から再構築することができます。

## このタスクについて

**ULW**

UNIX, Linux, and Windows では、**dmpmqcfg** コマンドを使用して IBM MQ キュー・マネージャーの構成をダンプできます。

**IBM i**

IBM i では、MQ 構成のダンプ (**DMPMQCFG**) コマンドを使用して、キュー・マネージャーの構成オブジェクトと権限をダンプできます。

## 手順

1. キュー・マネージャーが実行中であることを確認してください。

2. 使用しているプラットフォームに応じて、以下のコマンドのいずれかを使用してキュー・マネージャー構成をバックアップします。

- **ULW** UNIX, Linux, and Windows の場合: MQ 構成のダンプ・コマンド **dmpmqcfcg** を、デフォルトのフォーマット・オプションの (-f mqsc) MQSC、およびすべての属性 (-a) を指定して実行し、標準出力のリダイレクトを使用して、定義をファイルに保存します。以下に例を示します。

```
dmpmqcfcg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

- **IBM i** IBM i の場合: MQ 構成のダンプ・コマンド (**DMPMQMCFG**) を、デフォルトのフォーマット・オプション OUTPUT(\*MQSC) および EXPATTR(\*ALL) を指定して実行し、TOFILE および TOMBR を使用して、定義を物理ファイル・メンバーに保存します。以下に例を示します。

```
DMPMQMCFG MQMNAME(MYQMGR) OUTPUT(*MQSC) EXPATTR(*ALL) TOFILE(QMQMSAMP/QMQSC)
TOMBR (MYQMGRDEF)
```

## 関連タスク

559 ページの『[キュー・マネージャー構成の復元](#)』

まずキュー・マネージャーが稼働していることを確認してから、プラットフォームに適したコマンドを実行することによって、キュー・マネージャーの構成をバックアップから復元することができます。

## 関連情報

[dmpmqcfcg \(キュー・マネージャー構成のダンプ\)](#)

[MQ 構成のダンプ \(DMPMQMCFG\)](#)

## Multi キュー・マネージャー構成の復元

まずキュー・マネージャーが稼働していることを確認してから、プラットフォームに適したコマンドを実行することによって、キュー・マネージャーの構成をバックアップから復元することができます。

## このタスクについて

**ULW** UNIX, Linux, and Windows では、**runmqsc** コマンドを使用して IBM MQ キュー・マネージャーの構成を復元できます。

**IBM i** IBM i では、**STRMQMMQSC** コマンドを使用してキュー・マネージャーの構成オブジェクトおよび権限を復元できます。

## 手順

1. キュー・マネージャーが実行中であることを確認してください。  
データおよびログの損傷が、その他の方法ではリカバリー不能だった場合、キュー・マネージャーが再作成された可能性があることに注意してください。
2. 使用しているプラットフォームに応じて、以下のコマンドのいずれかを使用してキュー・マネージャーの構成を復元します。
  - **ULW** UNIX, Linux, and Windows の場合: キュー・マネージャーに対して **runmqsc** を実行し、標準入力のリダイレクトを使用して、MQ 構成のダンプ (**dmpmqcfcg**) コマンドによって生成されたスクリプト・ファイルから定義を復元します (558 ページの『[キュー・マネージャー構成のバックアップ](#)』を参照)。以下に例を示します。

```
runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
```

- **IBM i** IBM i の場合: キュー・マネージャーに対して **STRMQMMQSC** を実行し、**SRCMBR** および **SRCFILE** パラメーターを使用して、MQ 構成のダンプ (**DMPMQMCFG**) コマンドによって生成された物

理ファイル・メンバーから定義を復元します (558 ページの『キュー・マネージャー構成のバックアップ』を参照)。以下に例を示します。

```
STRMQMMQSC MQMNAME(MYQMGR) SRCFILE(QMQMSAMP/QMQSC) SRCMBR(MYQMGR)
```

## 関連タスク

[558 ページの『キュー・マネージャー構成のバックアップ』](#)

キュー・マネージャー構成をバックアップすると、ハードウェア障害が原因でキュー・マネージャー構成とログ・データの両方が失われ、キュー・マネージャーを再始動したりログからリカバリーしたりできない場合に、キュー・マネージャーをその定義から再構築することができます。

## 関連情報

[dmpmqcfg \(キュー・マネージャー構成のダンプ\)](#)

[runmqsc \(MQSC コマンドの実行\)](#)

[MQ 構成のダンプ \(DMPMQMCFG\)](#)

[IBM MQ コマンドの開始 \(STRMQMMQSC\)](#)

## バックアップ・キュー・マネージャーの使用

既存のキュー・マネージャーは、専用のバックアップ・キュー・マネージャーが災害復旧のための有効な方法です

## このタスクについて

バックアップ・キュー・マネージャーは、既存のキュー・マネージャーの非アクティブなコピーです。既存のキュー・マネージャーが重大なハードウェア障害によってリカバリー不能になった場合、バックアップ・キュー・マネージャーをオンラインにして、リカバリー不能のキュー・マネージャーの代わりに使用することができます。

バックアップ・キュー・マネージャーが災害復旧のための有効な方法であり続けるには、既存のキュー・マネージャーのログ・ファイルを定期的にバックアップ・キュー・マネージャーにコピーしておく必要があります。ログ・ファイルをコピーするために、既存のキュー・マネージャーを停止する必要はありません。必要なことは、キュー・マネージャーがログ・ファイルへの書き込みを完了したときに、ログ・ファイルをコピーすることだけです。既存のキュー・マネージャー・ログは継続的に更新されるため、既存のキュー・マネージャー・ログとバックアップ・キュー・マネージャー・ログにコピーされたログ・データの間には常にわずかな矛盾が存在します。バックアップ・キュー・マネージャーを定期的に更新することにより、この2つのログの間の矛盾を最小に抑えることができます。

バックアップ・キュー・マネージャーをオンラインにする必要がある場合は、必ず始動する前にアクティブ化してください。バックアップ・キュー・マネージャーをアクティブ化してから始動するという必要条件是、バックアップ・キュー・マネージャーが偶発的に始動されないようにするための予防措置です。バックアップ・キュー・マネージャーを活動化した後は、更新できなくなります。

**重要:** 古いバックアップ・キュー・マネージャーが新しいアクティブ・キュー・マネージャーになると、理由が何であろうと、バックアップ・キュー・マネージャーはなくなります。これは、実際には非同期複製の1つの形式であるため、新しいアクティブ・キュー・マネージャーは、論理的にいつかは古いアクティブ・キュー・マネージャーの後方に存在することになります。したがって、古いアクティブ・キュー・マネージャーは、新しいアクティブ・キュー・マネージャーのバックアップとして機能しなくなります。

## 手順

- バックアップ・キュー・マネージャーの作成、更新、および始動方法については、以下のトピックを参照してください。
  - [561 ページの『バックアップ・キュー・マネージャーの作成』](#)
  - [561 ページの『バックアップ・キュー・マネージャーの更新』](#)
  - [562 ページの『バックアップ・キュー・マネージャーの開始』](#)

## 関連概念

528 ページの『[ロギング: メッセージが失われないようにするための機能](#)』

IBM MQ は、キュー・マネージャーが制御する持続データに対する重要な変更をすべてリカバリー・ログに記録します。

## バックアップ・キュー・マネージャーの作成

既存のキュー・マネージャーの非アクティブなコピーとしてバックアップ・キュー・マネージャーを作成します。

## このタスクについて

**重要:** バックアップ・キュー・マネージャーは、リニア・ロギングの使用時にのみ使用できます。

バックアップ・キュー・マネージャーは、以下の条件を満たすことが必要です。

- 既存のキュー・マネージャーと同じ属性 (キュー・マネージャー名、ロギング・タイプ、ログ・ファイル・サイズなど) がある。
- 既存のキュー・マネージャーと同じプラットフォーム上に存在する。
- コード・レベルが既存のキュー・マネージャーと同等以上である。

## 手順

1. 制御コマンド **crtmqm** を使用して、既存のキュー・マネージャーのバックアップ・キュー・マネージャーを作成します。
2. [556 ページの『キュー・マネージャー・データのバックアップ』](#) の説明に従って、すべての既存のキュー・マネージャーのデータおよびログ・ファイル・ディレクトリー (すべてのサブディレクトリーを含む) をコピーします。
3. バックアップ・キュー・マネージャーのデータおよびログ・ファイル・ディレクトリー (すべてのサブディレクトリーを含む) を、既存のキュー・マネージャーからのコピーで上書きします。
4. 以下の例に示すように、バックアップ・キュー・マネージャー上で **strmqm** 制御コマンドを実行します。

```
strmqm -i BackupQMName
```

このコマンドにより、キュー・マネージャーが IBM MQ 内のバックアップ・キュー・マネージャーであることをフラグで示し、コピーされたすべてのログ・エクステンツを再生して、バックアップ・キュー・マネージャーを既存のキュー・マネージャーと一致させます。

## 関連情報

[crtmqm \(キュー・マネージャーの作成\)](#)

[strmqm \(キュー・マネージャーの始動\)](#)

## バックアップ・キュー・マネージャーの更新

バックアップ・キュー・マネージャーが災害復旧のための有効な方法であり続けるには、バックアップ・キュー・マネージャーを定期的に更新しておく必要があります。

## このタスクについて

定期的な更新によって、バックアップ・キュー・マネージャー・ログと現在のキュー・マネージャー・ログの間の矛盾を少なくすることができます。バックアップを行う前に、キュー・マネージャーを停止する必要はありません。



**警告:** バックアップ・キュー・マネージャーのログ・ディレクトリーに、不連続なログのセットをコピーした場合、欠落した最初のログが見つかるポイントまでのログが再生されます。

## 手順

1. バックアップを行うキュー・マネージャーで、以下のスクリプト (MQSC) コマンドを発行します。

```
RESET QMGR TYPE(ADVANCELOG)
```

これにより、現在のログへの書き込みが停止され、キュー・マネージャー・ロギングが次のログ・エクステントに拡張されます。これにより、現在時刻までに記録されたすべての情報が確実にバックアップされます。

2. バックアップを行うキュー・マネージャーで以下のスクリプト (MQSC) コマンドを発行することにより、(新しい) 現在のアクティブ・ログ・エクステント番号を入手します。

```
DIS QMSTATUS CURRLOG
```

3. 更新されたログ・エクステント・ファイルを、現在のキュー・マネージャー・ログ・ディレクトリーからバックアップ・キュー・マネージャー・ログ・ディレクトリーにコピーします。

最終更新から [562 ページの『2』](#) で示されている現在のエクステントまで (ただし、現在のエクステント自体は含まない) のすべてのログ・エクステントをコピーします。ログ・エクステント・ファイルのみをコピーします(「S..」で始まる)。

4. 以下の例に示すように、バックアップ・キュー・マネージャー上で **strmqm** 制御コマンドを実行します。

```
strmqm -r BackupQMName
```

これにより、コピーされたすべてのログ・エクステントが再生され、バックアップ・キュー・マネージャーをキュー・マネージャーと一致させます。再生が完了すると、再始動リカバリーに必要なすべてのログ・エクステント、およびメディア・リカバリーに必要なすべてのログ・エクステントを示すメッセージを受け取ります。

## 関連情報

[RESET QMGR](#)

[DISPLAY QMSTATUS](#)

[strmqm \(キュー・マネージャーの始動\)](#)

## バックアップ・キュー・マネージャーの開始

リカバリー不能なキュー・マネージャーのバックアップ・キュー・マネージャーを置き換えることができます。

## このタスクについて

リカバリー不能なキュー・マネージャーに専用のバックアップ・キュー・マネージャーが存在すれば、バックアップ・キュー・マネージャーをリカバリー不能なキュー・マネージャーの代わりにアクティブ化することができます。

リカバリー不能のキュー・マネージャーがバックアップ・キュー・マネージャーに置き換えられた場合、リカバリー不能のキュー・マネージャーの一部のキュー・マネージャー・データが失われることがあります。失われるデータの量は、バックアップ・キュー・マネージャーの最終更新時によって異なります。最終更新時が現在により近い時点であるほど、失われるキュー・マネージャー・データは少なくなります。

**注:** キュー・マネージャー・データおよびログ・ファイルが異なるディレクトリーに保持されている場合でも、それらのディレクトリーのバックアップおよび復元を同時に行ってください。キュー・マネージャーのデータとログ・ファイルの経過日数が異なっていると、キュー・マネージャーは有効な状態ではなく、おそらく始動しません。始動する場合でも、データは破損している可能性があります。

## 手順

1. 以下の例に示すように、**strmqm** 制御コマンドを実行してバックアップ・キュー・マネージャーをアクティブ化します。

```
strmqm -a BackupQMName
```

バックアップ・キュー・マネージャーがアクティブ化されます。これでアクティブになり、バックアップ・キュー・マネージャーは更新できなくなります。

- 以下の例に示すように、**strmqm** 制御コマンドを実行してバックアップ・キュー・マネージャーを始動します。

```
strmqm BackupQMName
```

IBM MQ は、これを再始動リカバリーと見なし、バックアップ・キュー・マネージャーのログを使用します。バックアップ・キュー・マネージャーへの最後の更新時に、再生が行われているため、最後に記録されたチェックポイントからのアクティブ・トランザクションのみがロールバックされます。

- すべてのチャンネルを再開します。
- できあがったディレクトリー構造をチェックして、必要なディレクトリーがすべて入っていることを確認してください。

IBM MQ のディレクトリーとサブディレクトリーの詳細については、[ファイル・システム・サポートの計画](#)を参照してください。

- ログ・ファイルだけでなく、ログ制御ファイルもあることを確認してください。また、IBM MQ が復元データの検索にあたって正しい場所を調べられるように、IBM MQ およびキュー・マネージャーの構成ファイルが一貫しているかどうかを確認してください。

## タスクの結果

データが正しくバックアップされ復元されていれば、キュー・マネージャーは始動します。

### 関連タスク

[180 ページの『停止したチャンネルの再始動』](#)

チャンネルが STOPPED 状態になるときは、手動でチャンネルを再始動しなければなりません。

### 関連情報

[strmqm \(キュー・マネージャーの始動\)](#)

## クラスター・エラー・リカバリーの変更 (z/OS 以外のサーバー)

IBM WebSphere MQ 7.1 以降では、キュー・マネージャーは、問題が解決するまで、問題を引き起こした操作を再実行します。5 日後になっても問題が解決していない場合、キュー・マネージャーはシャットダウンして、キャッシュがそれ以上古くならないようにします。

IBM WebSphere MQ 7.1 より前のバージョンでは、クラスターを管理するローカル・リポジトリー・マネージャーに関する問題をキュー・マネージャーが検出した場合、エラー・ログが更新されました。場合によっては、その後、管理しているクラスターが停止されました。キュー・マネージャーは、徐々に古くなるクラスター定義のキャッシュに依存したまま、クラスターとのアプリケーション・メッセージの交換を続行しました。IBM WebSphere MQ 7.1 以降では、キュー・マネージャーは、問題が解決するまで、問題を引き起こした操作を再実行します。5 日後になっても問題が解決していない場合、キュー・マネージャーはシャットダウンして、キャッシュがそれ以上古くならないようにします。キャッシュは古くなるにつれて、より多くの問題を引き起こします。7.1 以降のクラスター・エラーに関して変更された振る舞いは、z/OS には適用されません。

クラスター管理のすべての側面は、ローカル・リポジトリー・マネージャー・プロセス `amqrrmfa` によって、キュー・マネージャーのために処理されます。このプロセスは、クラスター定義がない場合を含め、すべてのキュー・マネージャーで実行されます。

IBM WebSphere MQ 7.1 より前のバージョンでは、ローカル・リポジトリー・マネージャーの問題をキュー・マネージャーが検出すると、少し間隔をおいてリポジトリー・マネージャーが停止されました。キュー・マネージャーは実行を継続して、アプリケーション・メッセージおよびキュー開始要求の処理や、トピックのパブリッシュまたはサブスクライブを行いました。

リポジトリー・マネージャーが停止すると、キュー・マネージャーが使用できるクラスター定義のキャッシュは、徐々に古くなりました。時間が経過するにつれて、メッセージは誤った宛先に送られ、アプリケーションは失敗しました。アプリケーションが、ローカル・キュー・マネージャーに伝搬されていないクラスター・キューまたはパブリケーション・トピックを開こうとして、失敗しました。

管理者は、エラー・ログ内のリポジトリー・メッセージを確認しない限り、クラスター構成に問題があることに気付かない可能性があります。長時間にわたって障害が認識されず、キュー・マネージャーがそ

のクラスター・メンバーシップを更新しないままになると、より多くの問題が発生しました。この不安定さはそのクラスター内のすべてのキュー・マネージャーに影響し、クラスターが不安定になりました。

IBM WebSphere MQ 7.1 以降、IBM MQ は別の方法でクラスターのエラー処理を行うようになりました。リポジトリ・マネージャーを停止してそのまま続行するのではなく、失敗した操作がリポジトリ・マネージャーによって再実行されます。キュー・マネージャーがリポジトリ・マネージャーの問題を検出すると、2 とおりある一連のアクションのいずれか 1 つを実行します。

1. キュー・マネージャーの操作を阻害しないエラーの場合、キュー・マネージャーはエラー・ログにメッセージを書き込みます。操作が成功するまで、失敗した操作を 10 分ごとに再実行します。デフォルトでは、エラーに対処するための時間は 5 日間です。この間に対処できなかった場合、キュー・マネージャーはエラー・ログにメッセージを書き込み、シャットダウンします。5 日後のシャットダウンは延期できます。
2. キュー・マネージャーの操作を阻害するエラーの場合、キュー・マネージャーはエラー・ログにメッセージを書き込み、直ちにシャットダウンします。

キュー・マネージャーの操作を阻害するエラーとは、キュー・マネージャーがそれまでに診断できなかったエラー、または予測不能な結果をもたらす可能性のあるエラーです。このタイプのエラーの場合、キュー・マネージャーは通常、結果的に FFST ファイルを作成します。キュー・マネージャーの操作を阻害するエラーは、IBM MQ のバグによって引き起こされるか、あるいは管理者またはプログラムが IBM MQ プロセスを終了するなどの予期しない操作を行うことによって引き起こされます。

エラー・リカバリーの振る舞いを変更する際のポイントは、矛盾するクラスター定義の数が増えて行く中で、キュー・マネージャーが実行を継続する時間を制限することです。クラスター定義の矛盾が増えるに連れて、アプリケーションが異常な振る舞いをする機会も増えることになります。

5 日後にキュー・マネージャーをシャットダウンするというデフォルトの選択は、矛盾の数を抑えることと、問題が検出され、解決されるまで、キュー・マネージャーを使用可能な状態に維持することのバランスを取った結果です。

このキュー・マネージャーがシャットダウンされるまでの時間は、問題を修正するか、キュー・マネージャーが予定どおりにシャットダウンされるまでの間、無期限に延長することができます。5 日間実行を継続することで、長い週末が終わるまでキュー・マネージャーの実行状態を維持し、あらゆる問題に対処する時間を稼ぐことや、キュー・マネージャーが再始動するまでの時間を延長することができます。

## 修正処置

クラスター・エラー・リカバリーの問題に対する処置は選択できます。1 つ目の選択肢では、問題をモニターし、修正します。2 つ目の選択肢では、問題をモニターし、修正を延期します。そして最後の選択肢では、IBM WebSphere MQ 7.1 より前のリリースで行われていたように、クラスター・エラー・リカバリーの管理を続行します。

1. キュー・マネージャーのエラー・ログで、エラー・メッセージ [AMQ9448](#) および [AMQ5008](#) をモニターし、問題を修正します。

[AMQ9448](#) は、コマンドの実行後にリポジトリ・マネージャーがエラーを返したことを示します。このエラーは、10 分ごとのコマンド再試行の開始時にマークを付け、シャットダウンが延期されなければ、最終的には 5 日後にキュー・マネージャーを停止します。

[AMQ5008](#) は、IBM MQ プロセスの欠落が原因で、キュー・マネージャーが停止したことを示します。5 日後にリポジトリ・マネージャーが停止した結果、[AMQ5008](#) になります。リポジトリ・マネージャーが停止すると、キュー・マネージャーも停止します。

2. キュー・マネージャーのエラー・ログで、エラー・メッセージ [AMQ9448](#) をモニターし、問題の修正を延期します。

SYSTEM.CLUSTER.COMMAND.QUEUE からのメッセージの取得を使用不可にすると、リポジトリ・マネージャーはコマンド実行の試行を止めて、処理を行うことなく無期限に継続します。ただし、リポジトリ・マネージャーがキューに対して保持しているハンドルはすべて解放されます。リポジトリ・マネージャーが停止されないため、5 日後になってもキュー・マネージャーは停止されません。

MQSC コマンドを実行して、SYSTEM.CLUSTER.COMMAND.QUEUE からのメッセージの取得を使用不可にします。

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(DISABLED)
```

SYSTEM.CLUSTER.COMMAND.QUEUE からのメッセージの受信を再開するには、以下の MQSC コマンドを実行します。

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
```

3. キュー・マネージャーのクラスター・エラー・リカバリー時の振る舞いを、IBM WebSphere MQ 7.1 より前の振る舞いと同じものに戻します。

キュー・マネージャーの調整パラメーターを設定して、リポジトリ・マネージャーが停止してもキュー・マネージャーは実行を継続するようにすることができます。

チューニング・パラメーターはファイルの TuningParameters スタンザにある

TolerateRepositoryFailure。qm.ini リポジトリ・マネージャーの停止時にキュー・マネージャーが停止しないようにするには、TolerateRepositoryFailure を TRUE に設定します。

565 ページの図 88 を参照してください。

キュー・マネージャーを再始動して、TolerateRepositoryFailure オプションを有効にします。

クラスター・エラーが発生し、そのためにリポジトリ・マネージャーの正常な始動が阻害されて、キュー・マネージャーの始動も阻害される場合、リポジトリ・マネージャーを使用せずにキュー・マネージャーを始動するには、TolerateRepositoryFailure を TRUE に設定します。

## 特別な考慮事項

IBM WebSphere MQ 7.1 の前に、クラスターの一部ではないキュー・マネージャーを管理する一部の管理者が、amqrrmfa プロセスを停止しました。amqrrmfa を停止しても、キュー・マネージャーに影響がありません。

以降で停止する amqrrmfa IBM WebSphere MQ 7.1 と、キュー・マネージャーの障害とみなされるため、キュー・マネージャーが停止する。キュー・マネージャーのチューニング・パラメーター TolerateRepositoryFailure を設定しない限り、amqrrmfa プロセスを 7.1 以降で停止することはできません。

## 例

```
TuningParameters:  
TolerateRepositoryFailure=TRUE
```

図 88. qm.ini で TolerateRepositoryFailure を TRUE に設定する

## 関連情報

キュー・マネージャー構成ファイル qm.ini

## JMS リソースの構成

JMS アプリケーションが、IBM MQ に接続してメッセージの送受信を行う宛先にアクセスするために必要なリソースを作成および構成する方法の 1 つに、Java Naming and Directory Interface (JNDI) を使用して、JNDI 名前空間という命名/ディレクトリー・サービス内のロケーションから管理対象オブジェクトを取り出す方法があります。JMS アプリケーションで管理対象オブジェクトを JNDI 名前空間から取り出すには、その前にまず管理対象オブジェクトを作成して構成する必要があります。

## このタスクについて

以下のいずれかのツールを使用して、IBM MQ で管理対象オブジェクトを作成して構成できます。

### IBM MQ Explorer

IBM MQ Explorer を使用して、LDAP、ローカル・ファイル・システム、またはその他のロケーションに格納される JMS オブジェクト定義を作成および管理できます。

## IBM MQ JMS 管理ツール

IBM MQ JMS 管理ツールは、LDAP、ローカル・ファイル・システム、またはその他のロケーションに格納される IBM MQ JMS オブジェクトの作成および構成に使用できる、コマンド行ツールです。JMS 管理ツールは、`runmqsc` に似た構文を使用します。また、スクリプト記述もサポートします。

この管理ツールは、構成ファイルを使用して特定のプロパティの値を設定します。サンプル構成ファイルが提供されています。開始する前に、このファイルを管理ツールを使用してご使用のシステムに合わせて編集し、JMS リソースを構成できます。構成ファイルについて詳しくは、[572 ページの『JMS 管理ツールの構成』](#)を参照してください。

WebSphere Application Server にデプロイする IBM MQ JMS アプリケーションは、アプリケーション・サーバーの JNDI リポジトリから JMS オブジェクトにアクセスする必要があります。したがって、JMS メッセージングを WebSphere Application Server と IBM MQ の間で使用する場合は、IBM MQ で作成するオブジェクトに対応するオブジェクトを WebSphere Application Server で作成する必要があります。

IBM MQ Explorer および IBM MQ JMS 管理ツールを、WebSphere Application Server に格納される IBM MQ JMS オブジェクトの管理に使用することはできません。その代わりに、以下のいずれかのツールを使用して、WebSphere Application Server の管理対象オブジェクトを作成して構成できます。

### WebSphere Application Server 管理コンソール

WebSphere Application Server 管理コンソールは、WebSphere Application Server 内の IBM MQ JMS オブジェクトの管理に使用できる、Web ベースのツールです。

### WebSphere Application Server wsadmin スクリプト・クライアント

WebSphere Application Server wsadmin スクリプト・クライアントには、WebSphere Application Server 内の IBM MQ JMS オブジェクトの管理に特化したコマンドが用意されています。

JMS アプリケーションを使用して、WebSphere Application Server 内から IBM MQ キュー・マネージャーのリソースにアクセスする場合は、IBM MQ classes for JMS のバージョンを持つ WebSphere Application Server で IBM MQ メッセージング・プロバイダーを使用する必要があります。IBM MQ メッセージング・プロバイダーを使用して JMS メッセージングを実行するすべてのアプリケーションは、WebSphere Application Server に付属の IBM MQ リソース・アダプターを使用します。通常、IBM MQ リソース・アダプターは、WebSphere Application Server フィックスパックが適用されると自動的に更新されますが、以前にリソース・アダプターを手動で更新した場合は、保守が正しく適用されるように構成を手動で更新する必要があります。

### 関連情報

[IBM MQ classes for JMS アプリケーションの作成](#)

[runmqsc](#)

## JNDI ネーム・スペースでの接続ファクトリーおよび宛先の構成

JMS アプリケーションは、Java Naming and Directory Interface (JNDI) を使用して、ネーミングおよびディレクトリー・サービスの管理対象オブジェクトにアクセスします。JMS 管理対象オブジェクトは、JNDI 名前空間と呼ばれるネーミングおよびディレクトリー・サービス内の特定の場所に保管されます。JMS アプリケーションは、管理対象オブジェクトを検索して IBM MQ に接続し、メッセージを送受信するために宛先にアクセスすることができます。

### このタスクについて

JMS アプリケーションは、以下のようにコンテキストを使用することによって、ネーミングおよびディレクトリー・サービスの JMS オブジェクトの名前を検索します。

#### 初期コンテキスト

JNDI 名前空間のルートを定義する初期コンテキスト。ネーミングおよびディレクトリー・サービスのロケーションごとに、開始点を定める初期コンテキストを指定する必要があります。JMS アプリケーションは、ネーミングおよびディレクトリー・サービスのそのロケーションにある管理対象オブジェクトの名前をその開始点から解決できます。

#### サブコンテキスト

1つのコンテキストには1つ以上のサブコンテキストを含めることができます。サブコンテキストは JNDI 名前空間のサブディビジョンであり、接続ファクトリーや宛先などの管理対象オブジェクトやそ

他のサブコンテキストを含めることができます。サブコンテキストはそれ自体がオブジェクトではなく、サブコンテキストにあるオブジェクトの命名規則を拡張したものにすぎません。

コンテキストは、IBM MQ Explorer または IBM MQ JMS 管理ツールを使用して作成できます。

IBM MQ classes for JMS アプリケーションで管理対象オブジェクトを JNDI 名前空間から取り出すには、その前に、IBM MQ Explorer または IBM MQ JMS 管理ツールを使用してまず管理対象オブジェクトを作成する必要があります。以下のタイプの JMS オブジェクトを作成して構成することができます。

#### 接続ファクトリー

JMS 接続ファクトリー・オブジェクトは、接続の標準的な構成プロパティのセットを定義します。JMS アプリケーションは、接続ファクトリーを使用して IBM MQ への接続を作成します。Point-to-Point メッセージ・ドメインとパブリッシュ/サブスクライブ・メッセージ・ドメインという、2つのメッセージ・ドメインのいずれかに固有の接続ファクトリーを作成できます。あるいは、JMS 1.1 以降、Point-to-Point およびパブリッシュ/サブスクライブ両方のメッセージングで使用できる、ドメイン非依存型接続ファクトリーを作成することができます。

#### Destination

JMS 宛先は、クライアントが生成するメッセージのターゲットを表すオブジェクトと、JMS アプリケーションがコンシュームするメッセージのソースを表すオブジェクトです。JMS アプリケーションは、メッセージの書き込みや取得に単一の宛先オブジェクトを使用することもできますし、個別の宛先オブジェクトを使用することもできます。以下の2つのタイプの宛先オブジェクトがあります。

- Point-to-Point メッセージングで使用される JMS キュー宛先
- パブリッシュ/サブスクライブ・メッセージングで使用される JMS トピック宛先

以下の図は、IBM MQ JNDI 名前空間で作成される JMS オブジェクトの例を示しています。

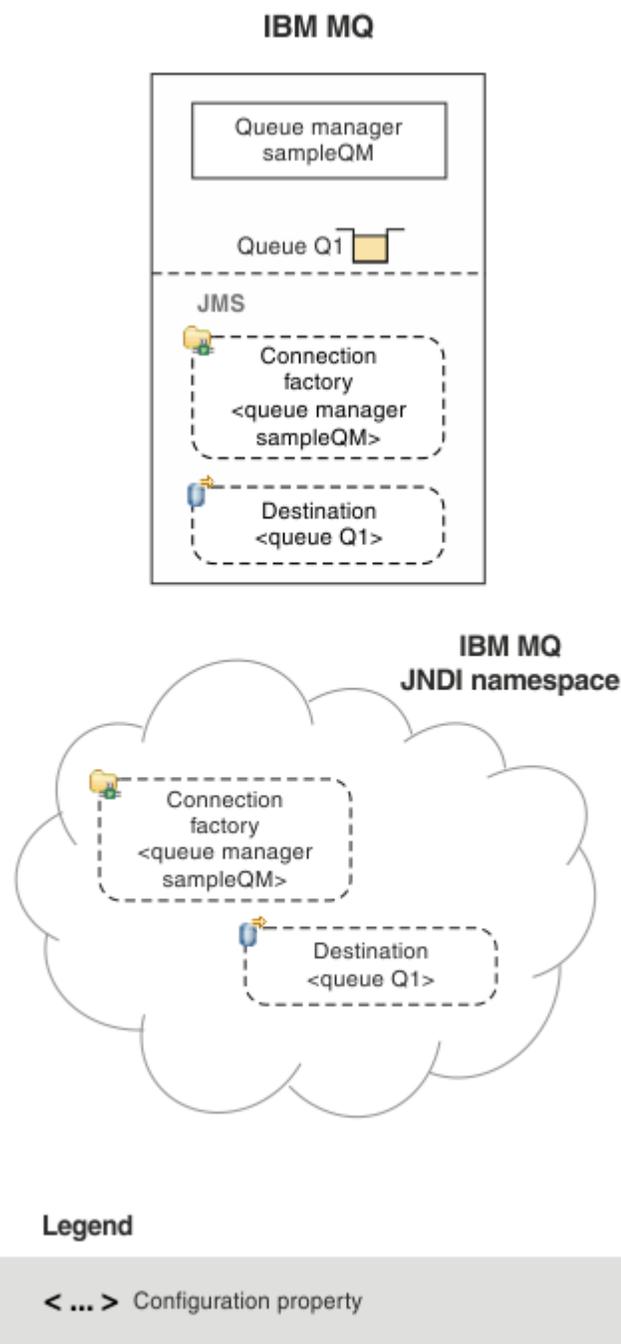


図 89. IBM MQ で作成される JMS オブジェクト

WebSphere Application Server と IBM MQ との間で JMS メッセージングを使用する場合、IBM MQ との通信に使用する対応オブジェクトを WebSphere Application Server で作成する必要があります。これらのオブジェクトのいずれかを WebSphere Application Server で作成すると、以下の図に示すように、それは WebSphere Application Server JNDI 名前空間に保管されます。

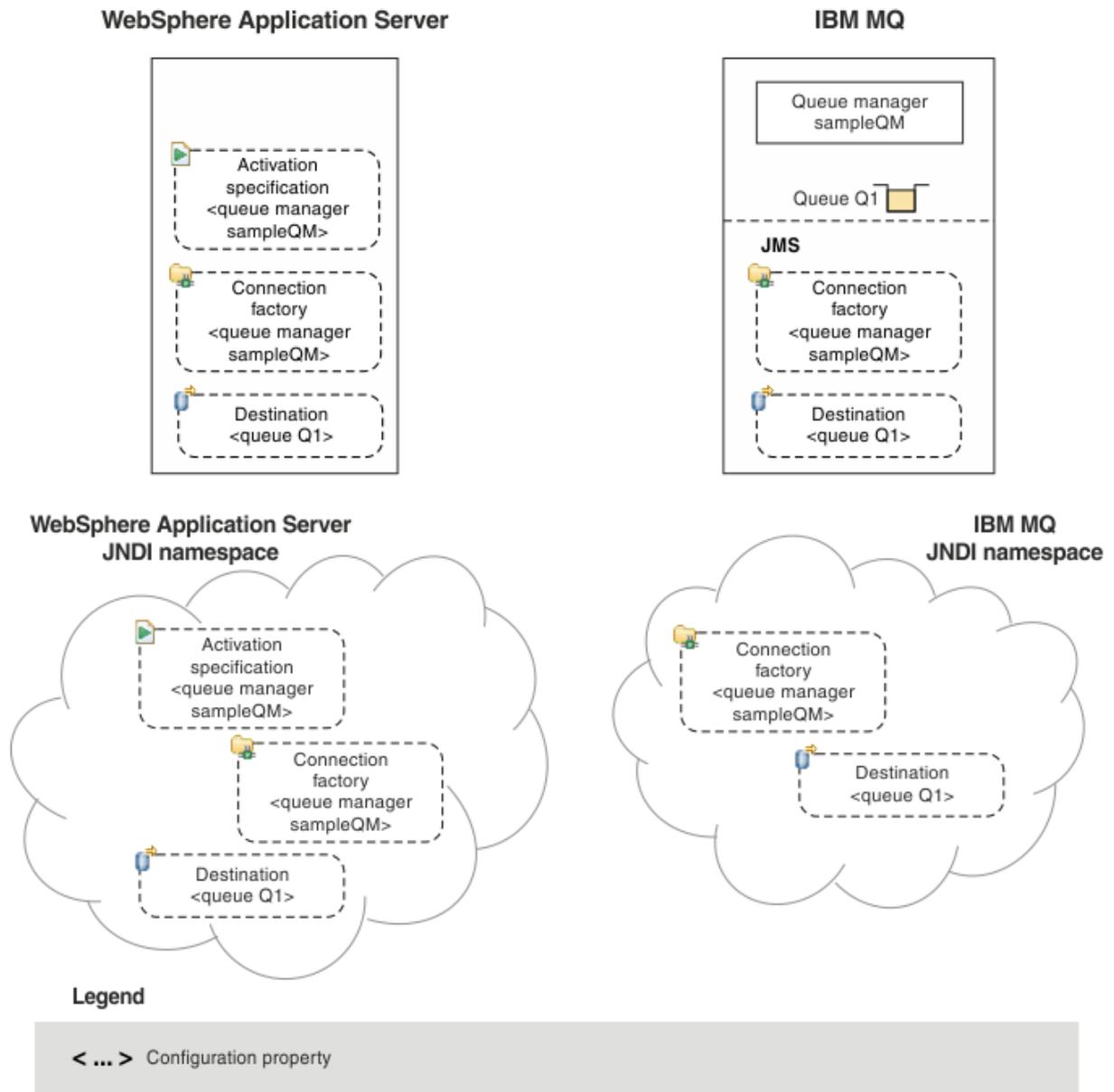


図 90. WebSphere Application Server で作成されるオブジェクトと、対応する IBM MQ のオブジェクト

アプリケーションがメッセージ駆動型 Bean (MDB) を使用する場合、接続ファクトリーはアウトバウンド・メッセージにのみ使用され、インバウンド・メッセージはアクティベーション・スペックによって受信されます。活動化仕様は、Java EE Connector Architecture 1.5 (JCA 1.5) 規格の一部です。JCA 1.5 は、IBM MQ などの JMS プロバイダーを、WebSphere Application Server などの Java EE アプリケーション・サーバーと統合する標準的な方法を提供します。JMS アクティベーション・スペックは、1つ以上のメッセージ駆動型 Bean (MDB) と関連付けることが可能で、これらの MDB が宛先に到達するメッセージを listen するのに必要な構成を提供します。

WebSphere Application Server 管理コンソールまたは wsadmin スクリプト・コマンドを使用して、必要な JMS リソースを作成および構成できます。

## 手順

- IBM MQ Explorer を使用した IBM MQ 用の JMS オブジェクトを構成するには、570 ページの『IBM MQ Explorer を使用した JMS オブジェクトの構成』を参照してください。

- IBM MQ JMS 管理ツールを使用した IBM MQ 用の JMS オブジェクトを構成するには、[571 ページの『管理ツールを使用した JMS オブジェクトの構成』](#)を参照してください。
- WebSphere Application Server 用の JMS オブジェクトを構成するには、[580 ページの『WebSphere Application Server での JMS リソースの構成』](#)を参照してください。

## タスクの結果

IBM MQ classes for JMS アプリケーションは、JNDI 名前空間から管理対象オブジェクトを取得することが可能で、必要な場合にはその 1 つ以上のプロパティを設定または変更できます。その操作には、IBM JMS 拡張機能または IBM MQ JMS 拡張機能のいずれかを使用します。

### 関連情報

[JNDI を使用して JMS アプリケーションで管理対象オブジェクトを取り出す](#)

[IBM MQ classes for JMS アプリケーションでの接続ファクトリーおよび宛先の作成と構成](#)

## IBM MQ Explorer を使用した JMS オブジェクトの構成

IBM MQ Explorer のグラフィカル・ユーザー・インターフェースを使用すると、JMS オブジェクトを IBM MQ オブジェクトからしたり、IBM MQ オブジェクトを JMS オブジェクトから作成したり、他の IBM MQ オブジェクトを管理およびモニターしたりできます。

### このタスクについて

IBM MQ Explorer は、IBM MQ オブジェクトがローカル・コンピューターによってホストされるか、リモート・システム上にあるかに関係なく、それらのオブジェクトを管理およびモニターできるグラフィカル・ユーザー・インターフェースです。IBM MQ Explorer は、Windows および Linux x86-64 で実行します。サポートされるプラットフォーム (z/OS を含む) で稼働中のキュー・マネージャーにリモートで接続することができるので、コンソールから、メッセージング・バックボーン全体を表示、探索、および変更することができます。

IBM MQ Explorer では、すべての接続ファクトリーが適切なコンテキストおよびサブコンテキストの「接続ファクトリー」フォルダーに格納されます。

以下のタイプのタスクは、IBM MQ Explorer を使用して、IBM MQ Explorer 内の既存のオブジェクトからコンテキストにより、または新規オブジェクト作成ウィザード内から実行できます。

- 以下のいずれかの IBM MQ オブジェクトから JMS 接続ファクトリーを作成します。
  - IBM MQ キュー・マネージャー (ローカル・コンピューターまたはリモート・システムのどちらでも)。
  - IBM MQ チャンネル。
  - IBM MQ リスナー。
- JMS 接続ファクトリーを使用して、IBM MQ キュー・マネージャーを IBM MQ Explorer に追加します。
- JMS キューを IBM MQ キューから作成します。
- IBM MQ キューを JMS キューから作成します。
- JMS トピックを IBM MQ トピックから作成します。これは、IBM MQ オブジェクトまたは動的トピックにすることができます。
- IBM MQ トピックを JMS トピックから作成します。

### 手順

- IBM MQ Explorer を開始します (まだ実行されていない場合)。  
IBM MQ Explorer が実行されていて、ウェルカム・ページが表示されている場合は、ウェルカム・ページを閉じて IBM MQ オブジェクトの管理を開始します。
- まだ行っていない場合は、JMS オブジェクトが保管される JNDI 名前空間のルートを定義する初期コンテキストを、名前指定およびディレクトリー・サービスに作成します。  
初期コンテキストを IBM MQ Explorer に追加したら、接続ファクトリー・オブジェクト、宛先オブジェクト、およびサブコンテキストを JNDI 名前空間に作成できます。

ナビゲーター・ビューの JMS 管理対象オブジェクト・フォルダーに、初期コンテキストが表示されます。JNDI 名前空間の内容がすべて表示されますが、IBM MQ Explorer で編集できるオブジェクトは、そこに格納されている IBM MQ classes for JMS オブジェクトに限られます。詳しくは、[初期コンテキストの追加](#)を参照してください。

- 必要なサブコンテキストおよび JMS 管理対象オブジェクトを作成して構成します。  
詳しくは、[JMS 管理対象オブジェクトの作成および構成](#)を参照してください。
- IBM MQ を構成します。  
詳しくは、[IBM MQ Explorer の使用による IBM MQ の構成](#)を参照してください。

## 関連情報

概要 [IBM MQ Explorer](#)

[IBM MQ classes for JMS アプリケーションでの接続ファクトリーおよび宛先の作成と構成](#)

## 管理ツールを使用した JMS オブジェクトの構成

IBM MQ JMS 管理ツールを使用して、8 つのタイプの IBM MQ classes for JMS オブジェクトのプロパティを定義し、JNDI 名前空間内に格納できます。アプリケーションは、JNDI を使用して、これらの管理対象オブジェクトをネーム・スペースから取り出すことができます。

### このタスクについて

次の表に、動詞を使用して作成、構成、操作できる管理対象オブジェクトの 8 つのタイプを示します。「キーワード」という列は、[571 ページの表 33](#)にあるコマンドの *TYPE* の部分と置き換えられるストリングを示します。

オブジェクト・タイプ	キーワード	説明
MQConnectionFactory	CF	JMS ConnectionFactory インターフェースの IBM MQ 実装。これは、Point-to-Point ドメインおよびパブリッシュ/サブスクライブ・ドメインの両方に接続を作成するためのファクトリー・オブジェクトを表します。
MQQueueConnectionFactory	QCF	JMS QueueConnectionFactory インターフェースの IBM MQ 実装。これは、Point-to-Point ドメインに接続を作成するためのファクトリー・オブジェクトを表します。
MQTopicConnectionFactory	TCF	JMS TopicConnectionFactory インターフェースの IBM MQ 実装。これは、パブリッシュ/サブスクライブ・ドメインに接続を作成するためのファクトリー・オブジェクトを表します。
MQQueue	Q	JMS Queue インターフェースの IBM MQ 実装。これは、Point-to-Point ドメインにおけるメッセージの宛先を表します。
MQTopic	T	JMS Topic インターフェースの IBM MQ 実装。これは、パブリッシュ/サブスクライブ・ドメインにおけるメッセージの宛先を表します。

表 33. 管理ツールで処理される JMS オブジェクト・タイプ (続き)

オブジェクト・タイプ	キーワード	説明
MQXAConnectionFactory <a href="#">572 ページの『1』</a>	XACF	JMS XAConnectionFactory インターフェースの IBM MQ 実装。これは、XA バージョンの JMS クラスを使用する Point-to-Point ドメインおよびパブリッシュ/サブスクライブ・ドメインの両方で接続を作成するためのファクトリー・オブジェクトを表します。
MQXAQueueConnectionFactory <a href="#">572 ページの『1』</a>	XAQCF	JMS XAQueueConnectionFactory インターフェースの IBM MQ 実装。これは、XA バージョンの JMS クラスを使用する Point-to-Point ドメインに接続を作成するためのファクトリー・オブジェクトを表します。
MQXATopicConnectionFactory <a href="#">572 ページの『1』</a>	XATCF	JMS XATopicConnectionFactory インターフェースの IBM MQ 実装。これは、XA バージョンの JMS クラスを使用するパブリッシュ/サブスクライブ・ドメインに接続を作成するためのファクトリー・オブジェクトを表します。

注:

1. これらのクラスは、アプリケーション・サーバーのベンダーが使用するために提供されたものです。アプリケーション・プログラマーには直接的には役立たないと考えられます。

これらのオブジェクトの構成方法について詳しくは、[579 ページの『JMS オブジェクトの構成』](#)を参照してください。

このツールを使用する必要があるプロパティ・タイプと値は、[IBM MQ classes for JMS オブジェクトのプロパティ](#)にリストされています。

また、[576 ページの『サブコンテキストの構成』](#)の説明のように、このツールを使用して JNDI 内のディレクトリー名前空間のサブコンテキストを操作することもできます。

IBM MQ Explorer を使用して、JMS 管理対象オブジェクトを作成して構成することもできます。

#### 関連情報

[IBM MQ classes for JMS アプリケーションでの接続ファクトリーおよび宛先の作成と構成](#)

[JNDI を使用して JMS アプリケーションで管理対象オブジェクトを取り出す](#)

## JMS 管理ツールの構成

IBM MQ JMS 管理ツールでは、構成ファイルを使用して、特定のプロパティの値を設定します。ご使用のシステムに合わせて編集できるサンプル構成ファイルが提供されています。

### このタスクについて

構成ファイルは、キーと値のペアのセットで構成されるプレーンテキストファイルで、等号(=)で区切られています。構成ファイルで定義されている 3 つのプロパティの値を設定して、管理ツールを構成します。これらの 3 つのプロパティの例は、以下のとおりです。

```
#Set the service provider
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
#Set the initial context
PROVIDER_URL=ldap://polaris/o=ibm_us,c=us
#Set the authentication type
SECURITY_AUTHENTICATION=none
```

(この例では、行の最初の列にあるハッシュ記号(#)は、コメントもしくは使用されない行を表します。)

IBM MQ では、デフォルトの構成ファイルとして使用されるサンプル構成ファイルが提供されています。このサンプル・ファイルは、JMSAdmin.config という名前で、MQ\_JAVA\_INSTALL\_PATH/bin ディレクトリにあります。このサンプル・ファイルを編集してシステムに必要な設定を定義することも、独自の構成ファイルを作成することもできます。

管理ツールの始動時に、574 ページの『管理ツールの開始』で説明されているように、コマンド行パラメーター `-cfg` を使用して、使用する構成ファイルを指定できます。ツールの起動時に構成ファイル名を指定しない場合は、ツールはデフォルトの構成ファイル (JMSAdmin.config) のロードを試行します。まず現行ディレクトリでこのファイルの検索が行われた後、MQ\_JAVA\_INSTALL\_PATH/bin ディレクトリでファイルが検索されます。ここで、MQ\_JAVA\_INSTALL\_PATH は IBM MQ classes for JMS インストール・システムへのパスを指します。

LDAP 環境に格納されている JMS オブジェクトの名前は、LDAP 命名規則に準拠する必要があります。命名規則の 1 つとして、オブジェクトやコンテキストの名前には、`cn=` (共通名) や `ou=` (組織単位) といった接頭部を付けなければなりません。この点、管理ツールでは、接頭部を付けなくてもオブジェクトやコンテキストの名前を参照できるようにすることで、LDAP サービス・プロバイダーの使用を簡単にしています。ツールは、接頭部が入力されない場合に、自動的に、入力された名前にデフォルトの接頭部を付けます。LDAP の場合、これは `cn=` です。必要に応じて、構成ファイル内の**名前**の接頭部プロパティを設定することにより、デフォルトの接頭部を変更できます。

注: 場合によっては、LDAP サーバーを Java オブジェクトの保管用に構成する必要があります。詳しくは、ご使用の LDAP サーバーの資料を参照してください。

## 手順

1. ツールが使用するサービス・プロバイダーを定義するには、**INITIAL\_CONTEXT\_FACTORY** プロパティを構成します。

このプロパティでサポートされる値は、以下のとおりです。

- `com.sun.jndi.ldap.LdapCtxFactory` (LDAP 用)
- `com.sun.jndi.fscontext.RefFSContextFactory` (ファイル・システム・コンテキスト用)
-  `com.ibm.jndi.LDAPCtxFactory` は、z/OS でのみサポートされており、LDAP サーバーへのアクセスが提供されます。ただし、このクラスは `com.sun.jndi.ldap.LdapCtxFactory` との互換性はありません。つまり、ある `InitialContextFactory` を使用して作成されたオブジェクトは、他の `InitialContextFactory` を使用して読み取ったり変更したりすることはできません。

管理ツールを使用して、他の JNDI コンテキストに接続することもできます。その際、JMSAdmin 構成ファイルで定義されている 3 つのパラメーターを使用します。別の `InitialContextFactory` を使用するには、次のようにします。

- a) **INITIAL\_CONTEXT\_FACTORY** プロパティを、必要とするクラス名に設定します。
- b) **USE\_INITIAL\_DIR\_CONTEXT**、**NAME\_PREFIX**、および **NAME\_READABILITY\_MARKER** プロパティを使用して、`InitialContextFactory` の振る舞いを定義します。

これらのプロパティの設定については、サンプル構成ファイルのコメントで説明されています。

サポートされている **INITIAL\_CONTEXT\_FACTORY** 値のいずれかを使用する場合は、**USE\_INITIAL\_DIR\_CONTEXT**、**NAME\_PREFIX**、および **NAME\_READABILITY\_MARKER** プロパティを定義する必要はありません。ただし、それらのプロパティに値を指定してシステム・デフォルトをオーバーライドすることは可能です。例えば、オブジェクトが LDAP 環境に格納されている場合、**NAME\_PREFIX** プロパティを必要な接頭部に設定することで、ツールによってオブジェクト名とコンテキスト名に追加されるデフォルトの接頭部を変更することができます。

3 つの `InitialContextFactory` プロパティのうちの 1 つ以上を省略した場合、管理ツールはその他のプロパティの値に基づいて適切なデフォルトを提供します。

2. **PROVIDER\_URL** プロパティを構成して、セッションの初期コンテキストの URL を定義します。

この URL は、ツールが実行するすべての JNDI 操作のルートです。このプロパティは、次の 2 つの形式でサポートされています。

- `ldap://hostname/contextname`

- file:[drive:]/pathname

LDAP URL の形式は、LDAP プロバイダーに応じて変わる場合があります。詳細は、LDAP の資料を参照してください。

3. **SECURITY\_AUTHENTICATION** プロパティを構成して、JNDI がセキュリティー信任状をサービス・プロバイダーに渡すかどうかを定義します。

このプロパティは、LDAP サービス・プロバイダーが使用されている場合にのみ使用され、以下の 3 つの値のうちいずれかを指定できます。

#### **none (匿名認証)**

このパラメーターが none に設定されている場合、JNDI は、セキュリティー信任状を一切サービス・プロバイダーに渡さず、匿名認証が実行されます。

#### **simple (単純認証)**

パラメーターが simple に設定された場合は、セキュリティー信任状が JNDI を介して基礎となるサービス・プロバイダーに渡されます。これらのセキュリティー信任状は、ユーザー識別名 (User DN) とパスワードの形式で与えられます。

#### **CRAM-MD5 (CRAM-MD5 認証メカニズム)**

パラメーターが CRAM-MD5 に設定された場合は、セキュリティー信任状が JNDI を介して基礎となるサービス・プロバイダーに渡されます。これらのセキュリティー信任状は、ユーザー識別名 (User DN) とパスワードの形式で与えられます。

**SECURITY\_AUTHENTICATION** プロパティに有効な値が指定されない場合、プロパティのデフォルトは none になります。

セキュリティー資格認定が必要な場合は、ツールの初期設定時にこれらの入力を要求するプロンプトが出されます。これを避けるには、JMSAdmin 構成ファイルで **PROVIDER\_USERDN** プロパティと **PROVIDER\_PASSWORD** プロパティを設定します。

**注:** これらのプロパティを使用しない場合、入力されたテキスト (パスワードを含む) が画面にもう一度表示されます。これには、セキュリティー上の事柄が含まれている場合があります。

ツール自体は認証を行わず、認証タスクは LDAP サーバーが代行します。ディレクトリーの様々な部分に対するアクセス権限のセットアップと保守は、LDAP サーバーの管理者が行う必要があります。詳細は、LDAP の資料を参照してください。認証が失敗した場合、ツールは、適切なエラー・メッセージを表示して終了します。

セキュリティーおよび JNDI について詳しくは、Oracle の Java Web サイト ([Oracle Technology Network for Java Developers](#)) にある文書を参照してください。

## 管理ツールの開始

管理ツールには、コマンド行インターフェースがあります。これは、対話式で使用することも、バッチ処理の開始に使用することもできます。

### このタスクについて

対話モードで使用する場合は、管理コマンドを入力するためのコマンド・プロンプトが表示されます。バッチ・モードを使用する場合は、ツールを開始するコマンドの中に、管理コマンド・スクリプトが入っているファイルの名前が組み込まれています。

### 手順

対話モード

- このツールを対話モードで開始する場合は、次のコマンドを入力します。

```
JMSAdmin [-t] [-v] [-cfg config_filename]
```

ここで、

-t

トレースをオンにする (デフォルトはオフ)

トレース・ファイルは"%MQ\_JAVA\_DATA\_PATH%\errors(Windows)または/var/mqm/trace(UNIX)で生成されます。トレース・ファイルの名前は、次の形式になります:

mqjms\_PID.trc

ここで、PID は JVM のプロセス ID です。

-v

詳細な出力を生成する (デフォルトは簡潔な出力)

-cfg config\_filename

代替の構成ファイルの名前を指定する。このパラメーターを省略した場合は、デフォルトの構成ファイル JMSAdmin.config が使用されます。構成ファイルについては、[572 ページの『JMS 管理ツールの構成』](#)を参照してください。

コマンド・プロンプトが表示されます。これは、管理ツールが管理コマンドを受け入れられる状態になったことを示します。このプロンプトは、最初に、次のように表示されます。

```
InitCtx>
```

これは、現行コンテキスト (つまり、すべての名前指定およびディレクトリー操作が現在参照している JNDI コンテキスト) が、**PROVIDER\_URL** 構成パラメーターに定義された初期コンテキストであることを示します。このパラメーターについては、[572 ページの『JMS 管理ツールの構成』](#)を参照してください。

ディレクトリーのネーム・スペースをトラバースすると、プロンプトにもそれが反映され、プロンプトには常に現行コンテキストが表示されます。

バッチ・モード

- このツールをバッチ・モードで開始する場合は、次のコマンドを入力します。

```
JMSAdmin test.scp
```

ここで、*test.scp* は、管理コマンドが入っているスクリプト・ファイルを表します。詳しくは、[575 ページの『管理コマンドの使用』](#)を参照してください。ファイルの最後は、END コマンドでなければなりません。

## 管理コマンドの使用

管理ツールは、管理動詞とその適切なパラメーターで構成されたコマンドを受け入れます。

### このタスクについて

以下の表に、管理ツールを使用してコマンドを入力する際に使用できる管理動詞をリストします。

動詞	短縮形	説明
ALTER	ALT	管理対象オブジェクトのプロパティを少なくとも 1 つ変更する
DEFINE	DEF	管理対象オブジェクトの作成、保管、またはサブコンテキストの作成を行う
DISPLAY	DIS	保管されている 1 つ以上の管理対象オブジェクトのプロパティ、または現行コンテキストの内容を表示する
DELETE	DEL	1 つ以上の管理対象オブジェクトをネーム・スペースから除去する、あるいは空のサブコンテキストを除去する

表 34. 管理動詞 (続き)

動詞	短縮形	説明
CHANGE	CHG	初期コンテキストの下に属する任意のディレクトリー・ネーム・スペースをトラバースし、現行コンテキストを変える (セキュリティの許可は保留)
COPY	CP	保管されている管理対象オブジェクトのコピーを作成し、それを別名で保管する
MOVE	MV	管理対象オブジェクトの保管場所の名前を変える
END		管理ツールを閉じる

## 手順

- まだ管理ツールが開始していない場合は、574 ページの『[管理ツールの開始](#)』で説明されているようにして開始します。

コマンド・プロンプトが表示されます。これは、管理ツールが管理コマンドを受け入れられる状態になったことを示します。このプロンプトは、最初に、次のように表示されます。

```
InitCtx>
```

現在のコンテキストを変更するには、576 ページの『[サブコンテキストの構成](#)』で説明されているように、CHANGE 動詞を使用します。

- 以下の形式でコマンドを入力します。

```
verb [param]*
```

ここで、**verb** の部分には、575 ページの表 34 にリストされているいずれかの管理動詞が入ります。すべての有効なコマンドには、コマンドの先頭に、標準の形式か短縮された形式の動詞が 1 つ含まれています。動詞の名前は大/小文字を区別しません。

- コマンドを終了するには、Enter を押します。複数のコマンドを一緒に入力する場合は、Enter を押す直前に正符号 (+) を入力します。

通常、コマンドの終わりには Enter を押します。ただし、Enter を押す直前に正符号 (+) を入力すると、これをオーバーライドできます。これによって、下の例に示されているように、複数行に渡ってコマンドを入力することが可能になります。

```
DEFINE Q(BookingsInputQueue) +
QMGR(QM.POLARIS.TEST) +
QUEUE(BOOKINGS.INPUT.QUEUE) +
PORT(1415) +
CCSID(437)
```

- 管理ツールを閉じるには、**END** 動詞を使用します。  
この動詞にはパラメーターを指定できません。

## サブコンテキストの構成

動詞 **CHANGE**、**DEFINE**、**DISPLAY**、および **DELETE** を使用して、ディレクトリー名前空間のサブコンテキストを構成できます。

## このタスクについて

これらの動詞の使用法を、以下の表に示します。

表 35. サブコンテキストの操作に使用されるコマンドの構文と説明

コマンド構文	説明
DEFINE CTX(ctxName)	現行コンテキストに ctxName という名前の子サブコンテキストを作成する処理を試行します。セキュリティー違反があった場合、そのサブコンテキストがすでに存在している場合、または無効な名前が指定された場合は、コマンドが失敗します。
DISPLAY CTX	現在のコンテキストの内容を表示します。管理対象オブジェクトには a、サブコンテキストには [D] の注釈が付けられます。各オブジェクトの Java タイプも表示されます。
DELETE CTX(ctxName)	現行コンテキストから ctxName という名前の子コンテキストを削除する処理を試行します。そのコンテキストが見つからない場合、コンテキストが空でない場合、またはセキュリティー違反があった場合は、コマンドが失敗します。
CHANGE CTX(ctxName)	<p>現行コンテキストを変更して、ctxName という子のコンテキストを参照するようにします。ctxName には次の 2 つの特殊値があり、このいずれかを指定できます。</p> <p><b>=UP</b> 現在のコンテキストの親に移動する</p> <p><b>=INIT</b> 初期コンテキストに直接移動する</p> <p>指定されたコンテキストが存在しない場合やセキュリティー違反があった場合は、コマンドが失敗します。</p>

LDAP 環境に格納されている JMS オブジェクトの名前は、LDAP 命名規則に準拠する必要があります。命名規則の 1 つとして、オブジェクトやコンテキストの名前には、cn= (共通名) や ou= (組織単位) といった接頭部を付けなければなりません。この点、管理ツールでは、接頭部を付けなくてもオブジェクトやコンテキストの名前を参照できるようにすることで、LDAP サービス・プロバイダーの使用を簡単にしています。ツールは、接頭部が入力されない場合に、自動的に、入力された名前にデフォルトの接頭部を付けます。LDAP の場合、これは cn= です。必要に応じて、構成ファイル内の **NAME\_PREFIX** プロパティーを設定することにより、デフォルトの接頭部を変更できます。詳しくは、572 ページの『JMS 管理ツールの構成』を参照してください。

注: 場合によっては、LDAP サーバーを Java オブジェクトの保管用に構成する必要があります。詳しくは、ご使用の LDAP サーバーの資料を参照してください。

## JMS オブジェクトの作成

JMS 接続ファクトリーおよび宛先オブジェクトを作成し、それらを JNDI 名前空間に格納するには、DEFINE 動詞を使用します。作成したオブジェクトを LDAP 環境に保管する場合、その名前は、一定の規則に準拠していなければなりません。管理ツールを使用すると、デフォルトの接頭部をオブジェクト名に追加することによって、LDAP 命名規則に準拠できます。

### このタスクについて

DEFINE 動詞は、指定したタイプ、名前、およびプロパティーを持つ管理対象オブジェクトを作成します。新しいオブジェクトが現在のコンテキストに格納されます。

LDAP 環境に格納されている JMS オブジェクトの名前は、LDAP 命名規則に準拠する必要があります。命名規則の 1 つとして、オブジェクトやコンテキストの名前には、cn= (共通名) や ou= (組織単位) といった接頭部を付けなければなりません。この点、管理ツールでは、接頭部を付けなくてもオブジェクトやコンテキストの名前を参照できるようにすることで、LDAP サービス・プロバイダーの使用を簡単にしています。ツールは、接頭部が入力されない場合に、自動的に、入力された名前にデフォルトの接頭部を付けます。LDAP の場合、これは cn= です。必要に応じて、構成ファイル内の **NAME\_PREFIX** プロパティーを設定する

ことにより、デフォルトの接頭部を変更できます。詳しくは、[572 ページの『JMS 管理ツールの構成』](#)を参照してください。

**注:** 場合によっては、LDAP サーバーを Java オブジェクトの保管用に構成する必要があります。詳しくは、ご使用の LDAP サーバーの資料を参照してください。

## 手順

1. まだ管理ツールが開始していない場合は、[574 ページの『管理ツールの開始』](#)で説明されているようにして開始します。

コマンド・プロンプトが表示されます。これは、管理ツールが管理コマンドを受け入れられる状態になったことを示します。

2. コマンド・プロンプトに、新規オブジェクトの作成場所となるコンテキストが表示されていることを確認します。

管理ツールを開始すると、最初はプロンプトが以下のように表示されます。

```
InitCtx>
```

現在のコンテキストを変更するには、[576 ページの『サブコンテキストの構成』](#)で説明されているように、CHANGE 動詞を使用します。

3. 接続ファクトリー、キュー宛先、またはトピック宛先を作成するには、以下のコマンド構文を使用します。

```
DEFINE TYPE (name) [property]*
```

つまり、DEFINE 動詞、管理対象オブジェクト参照 TYPE (name)、0 個以上のプロパティ ([IBM MQ classes for JMS](#) オブジェクトのプロパティを参照) の順に入力します。

4. 接続ファクトリー、キュー宛先、またはトピック宛先を作成するには、以下のコマンド構文を使用します。

```
DEFINE TYPE (name) [property]*
```

5. 新しく作成されたオブジェクトを表示するには、以下のコマンド構文で DISPLAY 動詞を使用します。

```
DISPLAY TYPE (name)
```

## 例

以下の例では、DEFINE 動詞を使用して、testQueue というキューが初期コンテキストに作成されたことを示しています。このオブジェクトは LDAP 環境に格納されるため、オブジェクト名 testQueue は接頭部がない状態で入力されますが、LDAP 命名規則に準拠するようにツールが自動的に接頭部を追加します。コマンド DISPLAY Q(testQueue) をサブミットした場合も同様に、この接頭部が追加されます。

```
InitCtx> DEFINE Q(testQueue)
InitCtx> DISPLAY CTX
Contents of InitCtx
a cn=testQueue          com.ibm.mq.jms.MQQueue
1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

## JMS オブジェクトを作成する時のエラー状態の例

オブジェクトの作成時に、いくつかの一般的なエラー状態が発生する場合があります。

このようなエラー状態の例を以下に示します。

## CipherSuite にマップされる CipherSpec

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SSLCIPHERSUITE(RC4_MD5_US)
WARNING: Converting CipherSpec RC4_MD5_US to
CipherSuite SSL_RSA_WITH_RC4_128_MD5
```

## オブジェクトのプロパティが無効

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PRIORITY(4)
Unable to create a valid object, please check the parameters supplied
Invalid property for a QCF: PRI
```

## プロパティ値のタイプが無効

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) CCSID(english)
Unable to create a valid object, please check the parameters supplied
Invalid value for CCS property: English
```

## プロパティの不調和 - クライアント/バインディング

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) HOSTNAME(polaris.hursley.ibm.com)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: Client-bindings attribute clash
```

## プロパティの不調和 - 出口の初期化

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SECEXITINIT(initStr)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: ExitInit string supplied
without Exit string
```

## プロパティ値が有効範囲外

```
InitCtx/cn=Trash> DEFINE Q(testQ) PRIORITY(12)
Unable to create a valid object, please check the parameters supplied
Invalid value for PRI property: 12
```

## プロパティが不明

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PIZZA(ham and mushroom)
Unable to create a valid object, please check the parameters supplied
Unknown property: PIZZA
```

以下は、JMS アプリケーションから JNDI 管理オブジェクトを探すときに Windows で起きる可能性のあるエラー状態の例です。

1. WebSphere JNDI プロバイダーの `com.ibm.websphere.naming.WsnInitialContextFactory` を使用する場合、サブコンテキストで定義されている管理オブジェクトにアクセスするには、スラッシュ (/) を使う必要があります。例えば、`jms/MyQueueName` です。円記号 (&#xa5;) を使うと、`InvalidNameException` がスローされます。
2. Oracle JNDI プロバイダーの `com.sun.jndi.fscontext.RefFSContextFactory` を使用している場合、サブコンテキストで定義されている管理オブジェクトにアクセスするには、円記号 (&#xa5;) を使う必要があります。例えば、`ctx1&#xa5;&#xa5;fred` です。スラッシュ (/) を使うと、`NameNotFoundException` がスローされます。

## JMS オブジェクトの構成

ディレクトリー・ネーム・スペース内の管理対象オブジェクトの操作には、動詞 ALTER、DEFINE、DISPLAY、DELETE、COPY、および MOVE を使用できます。

## このタスクについて

580 ページの表 36 に、これらの動詞の使い方を要約します。TYPE の部分は、571 ページの『管理ツールを使用した JMS オブジェクトの構成』で説明されている、必要な管理対象オブジェクトを表すキーワードに置き換えてください。

コマンド構文	説明
ALTER TYPE(name) [property]*	管理対象オブジェクトのプロパティを、指定された値に更新する処理を試行します。セキュリティ違反があった場合、指定されたオブジェクトが見つからない場合、または指定された新しいプロパティが無効である場合は、コマンドが失敗します。
DEFINE TYPE (name) [property]*	指定されたプロパティを持つタイプ TYPE のオブジェクトを作成し、それを name という名前で現行コンテキストに保管する処理を試行します。セキュリティ違反があった場合、指定された名前が無効であるかその名前のオブジェクトが存在している場合、または指定されたプロパティが無効である場合は、コマンドが失敗します。
DISPLAY TYPE (name)	name という名前で現行コンテキストにバインドされている、タイプ TYPE の管理対象オブジェクトのプロパティを表示します。オブジェクトが存在しない場合やセキュリティ違反があった場合は、コマンドが失敗します。
DELETE TYPE(name)	name という名前を持つタイプ TYPE の管理対象オブジェクトを現行コンテキストから除去する処理を試行します。オブジェクトが存在しない場合やセキュリティ違反があった場合は、コマンドが失敗します。
COPY TYPE (nameA) TYPE (nameB)	nameA という名前を持つタイプ TYPE の管理対象オブジェクトから、nameB という名前のコピーを作成します。これはすべて現行コンテキストの有効範囲内で行われます。コピー元のオブジェクトが存在しない場合、nameB という名前のオブジェクトが存在する場合、またはセキュリティ違反があった場合は、コマンドが失敗します。
MOVE TYPE(nameA) TYPE(nameB)	nameA という名前を持つタイプ TYPE の管理対象オブジェクトを、nameB という名前の管理対象オブジェクトに移動(名前変更)します。これはすべて現行コンテキストの有効範囲内で行われます。移動するオブジェクトが存在しない場合、nameB という名前のオブジェクトが存在する場合、またはセキュリティ違反があった場合は、コマンドが失敗します。

## WebSphere Application Server での JMS リソースの構成

WebSphere Application Server で JMS リソースを構成するために、管理コンソールまたは wsadmin コマンドを使用できます。

### このタスクについて

Java Message Service (JMS) アプリケーションは通常、アプリケーションの JMS プロバイダーに接続する方法とアクセスする宛先を記述した、外部で構成されたオブジェクトに依存します。JMS アプリケーションは Java Naming Directory Interface (JNDI) を使用して実行時に以下のタイプのオブジェクトにアクセスします。

- アクティベーション・スペック (Java EE アプリケーション・サーバーで使用)

- 統合接続ファクトリー (JMS 1.1 の場合、ドメイン非依存型 (統合) 接続ファクトリーはドメイン固有のキュー接続ファクトリーおよびトピック接続ファクトリーよりも優先されます)
- トピック接続ファクトリー (JMS 1.0 アプリケーションで使用)
- キュー接続ファクトリー (JMS 1.0 アプリケーションで使用)
- キュー
- トピック

WebSphere Application Server の IBM MQ メッセージング・プロバイダーを使用して、Java Message Service (JMS) メッセージング・アプリケーションは、IBM MQ システムを JMS メッセージング・リソースの外部プロバイダーとして使用できます。この方法を可能にするには、IBM MQ メッセージング・プロバイダーを WebSphere Application Server で構成し、IBM MQ ネットワーク上の任意のキュー・マネージャーに接続するために JMS リソースを定義します。

WebSphere Application Server を使用して、アプリケーション用の IBM MQ リソース(キュー接続ファクトリーなど)を構成したり、JMS 宛先に関連付けられたメッセージやサブスクリプションを管理したりすることができます。セキュリティは、IBM MQ を介して管理します。

### **WebSphere Application Server バージョン 8.5.5 の関連情報**

[IBM MQ メッセージング・プロバイダーを使用した相互運用](#)

[IBM MQ メッセージング・プロバイダーによるメッセージングの管理](#)

[管理コンソールのパネル名の、コマンド名および IBM MQ 名へのマッピング](#)

### **WebSphere Application Server 8.0 の関連情報**

[IBM MQ メッセージング・プロバイダーを使用した相互運用](#)

[IBM MQ メッセージング・プロバイダーによるメッセージングの管理](#)

[管理コンソールのパネル名の、コマンド名および IBM MQ 名へのマッピング](#)

### **WebSphere Application Server 7.0 の関連情報**

[IBM MQ メッセージング・プロバイダーを使用した相互運用](#)

[IBM MQ メッセージング・プロバイダーによるメッセージングの管理](#)

[管理コンソールのパネル名の、コマンド名および IBM MQ 名へのマッピング](#)

## **管理コンソールを使用した JMS リソースの構成**

WebSphere Application Server 管理コンソールを使用すると、IBM MQ JMS プロバイダーのアクティベーション・スペック、接続ファクトリー、および宛先を構成することができます。

### **このタスクについて**

WebSphere Application Server 管理コンソールを使用すると、以下のリソースを作成、表示、または変更することができます。

- アクティベーション・スペック
- ドメイン非依存型接続ファクトリー (JMS 1.1 以降)
- キュー接続ファクトリー
- トピック接続ファクトリー
- キュー
- トピック

以下の手順は、IBM MQ メッセージング・プロバイダーで使用する JMS リソースを、管理コンソールを使用して構成する方法の概要を説明しています。各手順には、詳細情報を参照できる WebSphere Application Server 製品資料のトピックの名前が含まれています。WebSphere Application Server 8.5.5、8.0、および 7.0 の製品資料内のこれらのトピックへのリンクについては、[関連リンク](#) を参照してください。

混合バージョンの WebSphere Application Server セルでは、すべてのバージョンのノード上の IBM MQ リソースを管理できます。ただし、一部のプロパティはすべてのバージョンでは使用できません。この場合、その特定のノードのプロパティのみが管理コンソールに表示されます。

## 手順

IBM MQ メッセージング・プロバイダーで使用するアクティベーション・スペックを作成または構成するには、以下のようにします。

- アクティベーション・スペックを作成するには、「IBM MQ JMS リソースの作成」ウィザードを使用します。  
ウィザードを使用してアクティベーション・スペックのすべての詳細を指定することもできますし、クライアント・チャンネル定義テーブル (CCDT) を使用することによって IBM MQ の接続詳細を指定することもできます。ウィザードを使用して接続詳細を指定する場合、ホストとポートの情報を別々に入力することもできますし、複数インスタンス・キュー・マネージャーを使用している場合は、接続名のリストの形式でホストとポートの情報を入力することもできます。詳しくは、「IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの作成」を参照してください。
- アクティベーション・スペックの構成プロパティを表示または変更するには、管理コンソールの IBM MQ メッセージング・プロバイダー接続ファクトリーの設定パネルを使用します。  
これらの構成プロパティは、関連したキューおよびトピックに対して接続を作成する方法を制御します。詳しくは、「IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの構成」を参照してください。

IBM MQ メッセージング・プロバイダーで使用する統合接続ファクトリー、キュー接続ファクトリー、またはトピック接続ファクトリーを作成または構成するには、以下のようにします。

- 接続ファクトリーを作成するには、作成する接続ファクトリーのタイプを最初に選択し、「IBM MQ JMS リソースの作成」ウィザードを使用して詳細を指定します。
  - JMS アプリケーションが Point-to-Point メッセージングのみを使用する場合、Point-to-Point メッセージング専用の接続を作成する際に使用できる、Point-to-Point メッセージング・ドメイン用のドメイン固有の接続ファクトリーを作成します。
  - JMS アプリケーションがパブリッシュ/サブスクライブ・メッセージングのみを使用する場合、パブリッシュ/サブスクライブ・メッセージング専用の接続を作成する際に使用できる、パブリッシュ/サブスクライブ・メッセージング・ドメイン用のドメイン固有の接続ファクトリーを作成します。
  - JMS 1.1 以降の場合、Point-to-Point メッセージングとパブリッシュ/サブスクライブ・メッセージングの両方で使用できるドメイン非依存型接続ファクトリーを作成します。これにより、アプリケーションは同じトランザクションで Point-to-Point とパブリッシュ/サブスクライブの両方の処理を実行できます。

ウィザードを使用して接続ファクトリーのすべての詳細を指定するか、クライアント・チャンネル定義テーブル (CCDT) を使用することによって IBM MQ の接続詳細を指定するかを選択できます。ウィザードを使用して接続詳細を指定する場合、ホストとポートの情報を別々に入力することもできますし、複数インスタンス・キュー・マネージャーを使用している場合は、接続名のリストの形式でホストとポートの情報を入力することもできます。詳しくは、「IBM MQ メッセージング・プロバイダーの接続ファクトリーの作成」を参照してください。

接続ファクトリーの構成プロパティを表示または変更するには、以下のようにします。

- 構成する接続ファクトリーのタイプについては、管理コンソールの接続ファクトリー設定パネルを使用します。  
構成プロパティは、関連したキューおよびトピックに対して接続を作成する方法を制御します。詳しくは、「IBM MQ メッセージング・プロバイダーのコレクション・ファクトリーの構成」、「IBM MQ メッセージング・プロバイダーのキュー・コレクション・ファクトリーの構成」、または「IBM MQ メッセージング・プロバイダーのトピック・コレクション・ファクトリーの構成」を参照してください。

IBM MQ メッセージング・プロバイダーを使用する Point-to-Point メッセージングの JMS キュー宛先を構成するには、以下のようにします。

- 管理コンソールの IBM MQ メッセージング・プロバイダー・キュー設定パネルを使用して、以下のタイプのプロパティを定義します。
  - 一般プロパティ (管理プロパティおよび IBM MQ キュー・プロパティを含む)。
  - キューをホストするキュー・マネージャーに接続する方法を指定する接続プロパティ。
  - IBM MQ メッセージング・プロバイダーの宛先に対する接続の動作を制御する拡張プロパティ。
  - キュー宛先用のすべてのカスタム・プロパティ。

詳しくは、「[IBM MQ メッセージング・プロバイダーのキューの構成](#)」を参照してください。  
IBM MQ メッセージング・プロバイダーを使用するパブリッシュ/サブスクライブ・メッセージングの JMS トピック宛先を作成または構成するには、以下のようにします。

- IBM MQ メッセージング・プロバイダー・トピック設定パネルを使用して、以下のタイプのプロパティを定義します。
  - 一般プロパティ (管理プロパティおよび IBM MQ トピック・プロパティを含む)。
  - IBM MQ メッセージング・プロバイダーの宛先に対する接続の動作を制御する拡張プロパティ。
  - キュー宛先のすべてのカスタム・プロパティ。

詳しくは、「[IBM MQ メッセージング・プロバイダーのトピックの構成](#)」を参照してください。

## 関連概念

### [40 ページの『クライアント・チャンネル定義テーブル』](#)

クライアント・チャンネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャンネル定義と認証情報を判別します。Multiplatforms の場合、CCDT は自動的に作成されます。それを、クライアント・アプリケーションに対して使用可能にする必要があります。

### [437 ページの『複数インスタンス・キュー・マネージャー』](#)

複数インスタンス・キュー・マネージャーは、異なる複数のサーバー上で構成されている同じキュー・マネージャーのインスタンスです。キュー・マネージャーの 1 つのインスタンスはアクティブ・インスタンスとして定義され、もう 1 つのインスタンスはスタンバイ・インスタンスとして定義されます。アクティブ・インスタンスで障害が発生すると、複数インスタンス・キュー・マネージャーは、スタンバイ・サーバーで自動的に再始動します。

## 関連タスク

### [357 ページの『パブリッシュ/サブスクライブ・メッセージングの構成』](#)

キュー型パブリッシュ/サブスクライブの開始、停止、および状況の表示を行えます。ストリームの追加および除去、ブローカー階層に対するキュー・マネージャーの追加および削除も行えます。

## WebSphere Application Server traditional 9.0 の関連情報

[IBM MQ メッセージング・プロバイダーのアクティベーション・スペック](#)

[IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの作成](#)

[IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの構成](#)

[IBM MQ メッセージング・プロバイダーの接続ファクトリーの作成](#)

[IBM MQ メッセージング・プロバイダーの統合接続ファクトリーの構成](#)

[IBM MQ メッセージング・プロバイダーのキュー接続ファクトリーの構成](#)

[IBM MQ メッセージング・プロバイダーのトピック接続ファクトリーの構成](#)

[IBM MQ メッセージング・プロバイダーのキューの構成](#)

[IBM MQ メッセージング・プロバイダーのトピックの構成](#)

## WebSphere Application Server 8.5.5 の関連情報

[IBM WebSphere MQ メッセージング・プロバイダーのアクティベーション・スペック](#)

[IBM WebSphere MQ メッセージング・プロバイダーのアクティベーション・スペックの作成](#)

[IBM WebSphere MQ メッセージング・プロバイダーのアクティベーション・スペックの構成](#)

[IBM WebSphere MQ メッセージング・プロバイダーの接続ファクトリーの作成](#)

[IBM WebSphere MQ メッセージング・プロバイダーの統合接続ファクトリーの構成](#)

[IBM WebSphere MQ メッセージング・プロバイダーのキュー接続ファクトリーの構成](#)

[IBM WebSphere MQ メッセージング・プロバイダーのトピック接続ファクトリーの構成](#)

[IBM WebSphere MQ メッセージング・プロバイダーのキューの構成](#)

[IBM WebSphere MQ メッセージング・プロバイダーのトピックの構成](#)

## WebSphere Application Server 8.0 の関連情報

[IBM WebSphere MQ メッセージング・プロバイダーのアクティベーション・スペック](#)

[IBM WebSphere MQ メッセージング・プロバイダーのアクティベーション・スペックの作成](#)

[IBM WebSphere MQ メッセージング・プロバイダーのアクティベーション・スペックの構成](#)

[IBM WebSphere MQ メッセージング・プロバイダーの接続ファクトリーの作成](#)

[IBM WebSphere MQ メッセージング・プロバイダーの統合接続ファクトリーの構成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのキュー接続ファクトリーの構成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのトピック接続ファクトリーの構成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのキューの構成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのトピックの構成](#)

### WebSphere Application Server 7.0 の関連情報

[IBM WebSphere MQ メッセージング・プロバイダーのアクティベーション・スペック](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのアクティベーション・スペックの作成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのアクティベーション・スペックの構成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーの接続ファクトリーの作成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーの統合接続ファクトリーの構成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのキュー接続ファクトリーの構成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのトピック接続ファクトリーの構成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのキューの構成](#)  
[IBM WebSphere MQ メッセージング・プロバイダーのトピックの構成](#)

## wsadmin スクリプト・コマンドを使用した JMS リソースの構成

WebSphere Application Server wsadmin スクリプト・コマンドを使用して、JMS アクティベーション・スペック、接続ファクトリー、キュー、およびトピックに関する情報を作成、変更、削除、または表示できます。IBM MQ リソース・アダプターの設定を表示および管理することもできます。

### このタスクについて

以下の手順は、IBM MQ メッセージング・プロバイダーで使用する JMS リソースを、WebSphere Application Server wsadmin コマンドを使用して構成する方法の概要を説明しています。これらのコマンドの使用方法について詳しくは、WebSphere Application Server 8.5.5、8.0、および 7.0 製品資料へのリンクについて「関連リンク」を参照してください。

コマンドを実行するには、wsadmin スクリプト・クライアントの AdminTask オブジェクトを使用します。

コマンドを使用して新規オブジェクトを作成したり変更したりした後で、その変更をマスター構成に保存します。例えば、次のコマンドを使用します。

```
AdminConfig.save()
```

使用可能な IBM MQ メッセージング・プロバイダー管理コマンドのリストと各コマンドの要旨を表示するには、wsadmin プロンプトで以下のコマンドを入力します。

```
print AdminTask.help('WMQAdminCommands')
```

指定されたコマンドの概要ヘルプを表示するには、wsadmin プロンプトで以下のコマンドを入力します。

```
print AdminTask.help('command_name')
```

### 手順

コマンドが発行されるスコープで定義された IBM MQ メッセージング・プロバイダー・リソースをすべてリストするには、以下のコマンドを使用します。

- アクティベーション・スペックをリストするには、**listWMQActivationSpecs** コマンドを使用します。
- 接続ファクトリーをリストするには、**listWMQConnectionFactoryies** コマンドを使用します。
- キュー・タイプ宛先をリストするには、**listWMQQueues** コマンドを使用します。

- トピック・タイプ宛先をリストするには、**listWMQTopics** コマンドを使用します。

特定のスコープにある、IBM MQ メッセージング・プロバイダーの JMS リソースを作成するには、以下のコマンドを使用します。

- アクティベーション・スペックを作成するには、**createWMQActivationSpec** コマンドを使用します。  
接続の確立に使用するすべてのパラメーターを指定してアクティベーション・スペックを作成することもできますし、クライアント・チャンネル定義テーブル (CCDT) を使用して接続先のキュー・マネージャーを見つけるようにアクティベーション・スペックを作成することもできます。
- 接続ファクトリーを作成するには、**createWMQConnectionFactory** コマンドで **-type** パラメーターを使用して、作成する接続ファクトリーのタイプを指定します。
  - JMS アプリケーションが Point-to-Point メッセージングのみを使用する場合、Point-to-Point メッセージング専用の接続を作成する際に使用できる、Point-to-Point メッセージング・ドメイン用のドメイン固有の接続ファクトリーを作成します。
  - JMS アプリケーションがパブリッシュ/サブスクライブ・メッセージングのみを使用する場合、パブリッシュ/サブスクライブ・メッセージング専用の接続を作成する際に使用できる、パブリッシュ/サブスクライブ・メッセージング・ドメイン用のドメイン固有の接続ファクトリーを作成します。
  - JMS 1.1 以降の場合、Point-to-Point メッセージングとパブリッシュ/サブスクライブ・メッセージングの両方で使用できるドメイン非依存型接続ファクトリーを作成します。これにより、アプリケーションは同じトランザクションで Point-to-Point とパブリッシュ/サブスクライブの両方の処理を実行できます。

デフォルトのタイプはドメイン非依存型接続ファクトリーです。

- キュー・タイプ宛先を作成するには、**createWMQQueue** コマンドを使用します。
- トピック・タイプ宛先を作成するには、**createWMQTopic** コマンドを使用します。

特定のスコープにある、IBM MQ メッセージング・プロバイダーの JMS リソースを変更するには、以下のコマンドを使用します。

- アクティベーション・スペックを変更するには、**modifyWMQActivationSpec** コマンドを使用します。  
アクティベーション・スペックのタイプは変更できません。例えば、アクティベーション・スペックを作成してすべての構成情報を手動で入力した後で、CCDT を使用するためにそれを変更することはできません。
- 接続ファクトリーを変更するには、**modifyWMQConnectionFactory** コマンドを使用します。
- キュー・タイプ宛先を変更するには、**modifyWMQQueue** コマンドを使用します。
- トピック・タイプ宛先を変更するには、**modifyWMQTopic** コマンドを使用します。

特定のスコープにある、IBM MQ メッセージング・プロバイダーの JMS リソースを削除するには、以下のコマンドを使用します。

- アクティベーション・スペックを削除するには、**deleteWMQActivationSpec** コマンドを使用します。
- 接続ファクトリーを削除するには、**deleteWMQConnectionFactory** コマンドを使用します。
- キュー・タイプ宛先を削除するには、**deleteWMQQueue** コマンドを使用します。
- トピック・タイプ宛先を削除するには、**deleteWMQTopic** コマンドを使用します。

特定の IBM MQ メッセージング・プロバイダー・リソースに関する情報を表示するには、以下のコマンドを使用します。

- 特定のアクティベーション・スペックに関連付けられたすべてのパラメーターとその値を表示するには、**showWMQActivationSpec** コマンドを使用します。
- 特定の接続ファクトリーに関連付けられたすべてのパラメーターとその値を表示するには、**showWMQConnectionFactory** コマンドを使用します。
- 特定のキュー・タイプ宛先に関連付けられたすべてのパラメーターとその値を表示するには、**showWMQQueue** コマンドを使用します。

- 特定のトピック・タイプ宛先に関連付けられたすべてのパラメーターとその値を表示するには、**deleteWMQTopic** コマンドを使用します。

IBM MQ リソース・アダプターまたは IBM MQ メッセージング・プロバイダーの設定を管理するには、以下のコマンドを使用します。

- 特定のスコープでインストールされた IBM MQ リソース・アダプターの設定を管理するには、**manageWMQ** コマンドを使用します。
- **manageWMQ** コマンドで設定可能なすべてのパラメーターとその値を表示するには、**showWMQ** コマンドを使用します。これらの設定は、IBM MQ リソース・アダプターまたは IBM MQ メッセージング・プロバイダーと関連しています。**showWMQ** コマンドは、IBM MQ リソース・アダプターで設定されたすべてのカスタム・プロパティーも表示します。

## 関連概念

### [40 ページの『クライアント・チャンネル定義テーブル』](#)

クライアント・チャンネル定義テーブル (CCDT) は、クライアント・アプリケーションがキュー・マネージャーに接続するために使用するチャンネル定義と認証情報を判別します。Multiplatforms の場合、CCDT は自動的に作成されます。それを、クライアント・アプリケーションに対して使用可能にする必要があります。

### [437 ページの『複数インスタンス・キュー・マネージャー』](#)

複数インスタンス・キュー・マネージャーは、異なる複数のサーバー上で構成されている同じキュー・マネージャーのインスタンスです。キュー・マネージャーの 1 つのインスタンスはアクティブ・インスタンスとして定義され、もう 1 つのインスタンスはスタンバイ・インスタンスとして定義されます。アクティブ・インスタンスで障害が発生すると、複数インスタンス・キュー・マネージャーは、スタンバイ・サーバーで自動的に再始動します。

## 関連タスク

### [357 ページの『パブリッシュ/サブスクライブ・メッセージングの構成』](#)

キュー型パブリッシュ/サブスクライブの開始、停止、および状況の表示を行えます。ストリームの追加および除去、ブローカー階層に対するキュー・マネージャーの追加および削除も行えます。

## WebSphere Application Server バージョン 8.5.5 の関連情報

[createWMQActivationSpec](#) コマンド

[createWMQConnectionFactory](#) コマンド

[createWMQQueue](#) コマンド

[createWMQTopic](#) コマンド

[deleteWMQActivationSpec](#) コマンド

[deleteWMQConnectionFactory](#) コマンド

[deleteWMQQueue](#) コマンド

[deleteWMQTopic](#) コマンド

[listWMQActivationSpecs](#) コマンド

[listWMQConnectionFactories](#) コマンド

[listWMQQueues](#) コマンド

[listWMQTopics](#) コマンド

[modifyWMQActivationSpec](#) コマンド

[modifyWMQConnectionFactory](#) コマンド

[modifyWMQQueue](#) コマンド

[modifyWMQTopic](#) コマンド

[showWMQActivationSpec](#) コマンド

[showWMQConnectionFactory](#) コマンド

[showWMQQueue](#) コマンド

[showWMQTopic](#) コマンド

[showWMQ](#) コマンド

[manageWMQ](#) コマンド

## WebSphere Application Server 8.5.5 の関連情報

[createWMQActivationSpec コマンド](#)  
[createWMQConnectionFactory コマンド](#)  
[createWMQQueue コマンド](#)  
[createWMQTopic コマンド](#)  
[deleteWMQActivationSpec コマンド](#)  
[deleteWMQConnectionFactory コマンド](#)  
[deleteWMQQueue コマンド](#)  
[deleteWMQTopic コマンド](#)  
[listWMQActivationSpecs コマンド](#)  
[listWMQConnectionFactories コマンド](#)  
[listWMQQueues コマンド](#)  
[listWMQTopics コマンド](#)  
[modifyWMQActivationSpec コマンド](#)  
[modifyWMQConnectionFactory コマンド](#)  
[modifyWMQQueue コマンド](#)  
[modifyWMQTopic コマンド](#)  
[showWMQActivationSpec コマンド](#)  
[showWMQConnectionFactory コマンド](#)  
[showWMQQueue コマンド](#)  
[showWMQTopic コマンド](#)  
[showWMQ コマンド](#)  
[manageWMQ コマンド](#)

## WebSphere Application Server 8.0 の関連情報

[createWMQActivationSpec コマンド](#)  
[createWMQConnectionFactory コマンド](#)  
[createWMQQueue コマンド](#)  
[createWMQTopic コマンド](#)  
[deleteWMQActivationSpec コマンド](#)  
[deleteWMQConnectionFactory コマンド](#)  
[deleteWMQQueue コマンド](#)  
[deleteWMQTopic コマンド](#)  
[listWMQActivationSpecs コマンド](#)  
[listWMQConnectionFactories コマンド](#)  
[listWMQQueues コマンド](#)  
[listWMQTopics コマンド](#)  
[modifyWMQActivationSpec コマンド](#)  
[modifyWMQConnectionFactory コマンド](#)  
[modifyWMQQueue コマンド](#)  
[modifyWMQTopic コマンド](#)  
[showWMQActivationSpec コマンド](#)  
[showWMQConnectionFactory コマンド](#)  
[showWMQQueue コマンド](#)  
[showWMQTopic コマンド](#)  
[showWMQ コマンド](#)  
[manageWMQ コマンド](#)

## WebSphere Application Server 7.0 の関連情報

[createWMQActivationSpec コマンド](#)  
[createWMQConnectionFactory コマンド](#)  
[createWMQQueue コマンド](#)  
[createWMQTopic コマンド](#)  
[deleteWMQActivationSpec コマンド](#)  
[deleteWMQConnectionFactory コマンド](#)  
[deleteWMQQueue コマンド](#)  
[deleteWMQTopic コマンド](#)  
[listWMQActivationSpecs コマンド](#)  
[listWMQConnectionFactoryies コマンド](#)  
[listWMQQueues コマンド](#)  
[listWMQTopics コマンド](#)  
[modifyWMQActivationSpec コマンド](#)  
[modifyWMQConnectionFactory コマンド](#)  
[modifyWMQQueue コマンド](#)  
[modifyWMQTopic コマンド](#)  
[showWMQActivationSpec コマンド](#)  
[showWMQConnectionFactory コマンド](#)  
[showWMQQueue コマンド](#)  
[showWMQTopic コマンド](#)  
[showWMQ コマンド](#)  
[manageWMQ コマンド](#)

## JMS 2.0 共有サブスクリプションの使用

WebSphere Application Server traditional 9.0 では、IBM MQ 9.0 に JMS 2.0 共有サブスクリプションを構成して使用することができます。

### このタスクについて

JMS 2.0 の仕様で、1 つのサブスクリプションを 1 つ以上のコンシューマーがオープンできるという共有サブスクリプションの概念が導入されました。メッセージはそれらのすべてのコンシューマー間で共有されます。同じキュー・マネージャーに接続する限り、コンシューマーが存在する場所に制限はありません。

共有サブスクリプションは、非共有サブスクリプションと呼ばれるようになったサブスクリプションと同じ意味で、永続サブスクリプションにすることも非永続サブスクリプションにすることもできます。

使用するサブスクリプションをコンシューマーが識別できるように、サブスクリプション名を指定する必要があります。これは非共有の永続サブスクリプションと似ていますが、共有サブスクリプションが必要な場合は、あらゆるケースでサブスクリプション名が必要になります。しかし、クライアント ID は、共有の永続サブスクリプションには必要ありません (指定することもできますが必須ではありません)。

共有サブスクリプションはロード・バランシング・メカニズムと見なすことができますが、IBM MQ でも JMS 2.0 仕様でも、メッセージをコンシューマー間に分散させる方法についての規定はありません。

WebSphere Application Server traditional 9.0 では、IBM MQ 9.0 リソース・アダプターが事前にインストールされます。

以下の手順は、WebSphere Application Server traditional 管理コンソールで共有永続サブスクリプションまたは共有非永続サブスクリプションを使用するためのアクティベーション・スペックを構成する方法を示しています。

## 手順

最初に JNDI でオブジェクトを作成します。

1. トピック宛先を JNDI で通常どおり作成します (581 ページの『[管理コンソールを使用した JMS リソースの構成](#)』を参照)。
2. アクティベーション・スペックを作成します (581 ページの『[管理コンソールを使用した JMS リソースの構成](#)』を参照)。

必要なプロパティを正確に指定してアクティベーション・スペックを作成できます。永続サブスクリプションを使用する場合は、作成時にそのことを選択して名前を指定できます。非永続サブスクリプションを使用する場合は、現時点では名前を指定できません。その代わりに、サブスクリプション名としてカスタム・プロパティを作成する必要があります。

必要なカスタム・プロパティを指定して、作成したアクティベーション・スペックを更新します。次の2つのカスタム・プロパティを指定する必要があります。

- いずれの場合でも、このアクティベーション・スペックが共用サブスクリプションを使用することを示すカスタム・プロパティを作成する必要があります。
- サブスクリプションを非永続として作成した場合は、サブスクリプション名プロパティをカスタム・プロパティとして設定する必要があります。

以下の表に、各カスタム・プロパティに指定できる有効な値を示します。

プロパティ名	タイプ	有効値
sharedSubscription	ストリング	true、false
subscriptionName	ストリング	ゼロでない長さの Java ストリング

3. 「**アクティベーション・スペック・コレクション (Activation specification collection)**」フォームに表示されたリストから、該当するアクティベーション・スペックを選択します。  
アクティベーション・スペックの詳細が「**IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定**」フォームに表示されます。
4. 「**IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定**」フォームで、「**カスタム・プロパティ**」をクリックします。  
「**カスタム・プロパティ**」フォームが表示されます。
5. 非永続サブスクリプションを使用している場合は、subscriptionName カスタム・プロパティを作成します。  
アクティベーション・スペックの「**カスタム・プロパティ**」パネルで、「**新規**」をクリックし、以下の詳細情報を入力します。

### 名前

カスタム・プロパティの名前。この場合は subscriptionName です。

### 値

カスタム・プロパティの値。「**値**」フィールドで JNDI 名を使用できます (例: WASSharedSub0ne)。

### タイプ

カスタム・プロパティのタイプ。リストからカスタム・プロパティ・タイプを選択します。この場合 java.lang.String です。

6. 共用永続サブスクリプションと共用非永続サブスクリプションのどちらの場合も、sharedSubscription カスタム・プロパティを作成します。  
アクティベーション・スペックの「**カスタム・プロパティ**」パネルで、「**新規**」をクリックし、以下の詳細情報を入力します。

### 名前

カスタム・プロパティの名前。この場合は sharedSubscription です。

## 値

カスタム・プロパティの値。アクティベーション・スペックが共用サブスクリプションを使用することを示すために、この値を `true` に設定します。後でこのアクティベーション・スペックで共用サブスクリプションの使用を止める場合は、このカスタム・プロパティの値を `false` に設定します。

## タイプ

カスタム・プロパティのタイプ。リストからカスタム・プロパティ・タイプを選択します。この場合 `java.lang.String` です。

7. プロパティを設定したら、アプリケーション・サーバーを再始動します。

メッセージを受信するとアクティベーション・スペックのメッセージ駆動型 Bean (MDB) が起動され、それらの MDB の間でのみ、送信されたメッセージが共用されます。

## 関連情報

[複製サブスクリプションおよび共用サブスクリプション](#)

[サブスクリプション永続性](#)

[インバウンド通信のリソース・アダプターの構成](#)

**WebSphere Application Server traditional 9.0 の関連情報**

[IBM MQ メッセージング・プロバイダーのトピックの構成](#)

[IBM MQ メッセージング・プロバイダーのアクティベーション・スペック](#)

[IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの作成](#)

[IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの構成](#)

[IBM MQ メッセージング・プロバイダー JMS リソースのカスタム・プロパティの構成](#)

## JMS 2.0 の ConnectionFactory および Destination の Lookup プロパティの使用

WebSphere Application Server traditional 9.0 では、アクティベーション・スペックで `ConnectionFactoryLookup` プロパティと `DestinationLookup` プロパティに管理対象オブジェクトの JNDI 名を指定して、他のアクティベーション・スペック・プロパティより優先的に使用することができます。

## このタスクについて

JMS 2.0 仕様では、メッセージ駆動型 Bean (MDB) の起動に使用するアクティベーション・スペックに、2 つの追加のプロパティが規定されました。以前は、メッセージング・システムへの接続に必要な詳細情報を入力し、どの宛先からメッセージを取得するのかを定義するためのカスタム・プロパティを、各ベンダーがアクティベーション・スペックに指定する必要がありました。

標準になった `connectionFactoryLookup` および `destinationLookup` プロパティを使用して、検索して使用する該当オブジェクトの JNDI 名を指定できます。WebSphere Application Server traditional 9.0 では、IBM MQ 9.0 リソース・アダプターが事前にインストールされます。

以下の手順は、WebSphere Application Server traditional 管理コンソールを使用してこれらの 2 つのプロパティをカスタマイズして使用方法を示しています。

## 手順

最初に JNDI でオブジェクトを作成します。

1. `ConnectionFactory` を JNDI で通常どおり作成します ([581 ページの『管理コンソールを使用した JMS リソースの構成』](#)を参照)。
2. 宛先を JNDI で通常どおり作成します ([581 ページの『管理コンソールを使用した JMS リソースの構成』](#)を参照)。

宛先オブジェクトには正確な値を指定してください。

3. 必要な値を使用してアクティベーション・スペックを作成します (581 ページの『[管理コンソールを使用した JMS リソースの構成](#)』を参照)。

必要なプロパティを正確に指定してアクティベーション・スペックを作成できます。ただし、以下の考慮事項に留意してください。

- IBM MQ リソース・アダプターで Java EE の接続ファクトリーと宛先のルックアップ・プロパティを使用する場合、アクティベーション・スペックの作成時にどのプロパティが使用されているかはあまり重要ではありません ([ActivationSpec の ConnectionFactoryLookup および DestinationLookup プロパティ](#)を参照)。
- 接続ファクトリーまたは宛先でまだ定義されていないプロパティであっても、アクティベーション・スペックには指定する必要があります。したがって、接続コンシューマー・プロパティと追加のプロパティ、および実際に接続が作成されたときに使用する認証情報を定義する必要があります。
- 接続ファクトリーで定義されるプロパティのうち、ClientID プロパティには特別な処理があります。その理由は、一般的なシナリオでは、単一の接続ファクトリーを複数のアクティベーション・スペックで使用するからです。これによって管理が簡略化されますが、JMS 仕様では固有のクライアント ID が必要なため、ConnectionFactory に設定された値をアクティベーション・スペックでオーバーライドできなければなりません。ClientID をアクティベーション・スペックで設定しないと、接続ファクトリーの値が使用されます。

WebSphere Application Server 管理コンソールで 2 つの新しいカスタム・プロパティを指定して、作成したアクティベーション・スペックを更新する (手順 [591 ページの『4』](#)を参照) か、代わりにアノテーションを使用します (手順 [591 ページの『5』](#)を参照)。

4. WebSphere Application Server 管理コンソールでアクティベーション・スペックを更新します。

これらの 2 つのプロパティは、アクティベーション・スペックのカスタム・プロパティ・パネルで設定する必要があります。これらのプロパティは、アクティベーション・スペックのメイン・パネルにもアクティベーション・スペックの作成ウィザードにも表示されません。

- a) 「**アクティベーション・スペック・コレクション (Activation specification collection)**」フォームに表示されたリストから、該当するアクティベーション・スペックを選択します。

アクティベーション・スペックの詳細が「**IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定**」フォームに表示されます。

- b) 「**IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定**」フォームで、「**カスタム・プロパティ**」をクリックします。

「**カスタム・プロパティ**」フォームが表示されます。

- c) 「**カスタム・プロパティ**」フォームで、2 つの新しいカスタム・プロパティ (いずれも `java.lang.String` 型) を作成します。

いずれの場合も、「**新規**」をクリックしてから、以下のカスタム・プロパティの詳細情報を入力します。

#### 名前

カスタム・プロパティの名前。connectionFactoryLookup または destinationLookup のいずれかです。

#### 値

カスタム・プロパティの値。「**値**」フィールドには、QuoteCF や QuoteQ などの JNDI 名を使用できます。

#### タイプ

カスタム・プロパティのタイプ。リストからカスタム・プロパティ・タイプを選択します。この場合 `java.lang.String` です。

これで、デプロイされた MDB がこれらの値を使用して接続ファクトリーおよび宛先を作成するようになります。MDB をデプロイするときに、JNDI 値の構成を設定する必要はありません。

5. アクティベーション・スペックの代わりにアノテーションを使用します。

MDB コード内でアノテーションを使用して値を指定することもできます。例えば、QuoteCF と QuoteQ の JNDI 名を使用すると、コードは以下のようになります。

```
@MessageDriven(activationConfig = {
    @ActivationConfigProperty(propertyName = "destinationType" , propertyValue =
"javax.jms.Topic" ),
    @ActivationConfigProperty(propertyName = "destinationLookup" , propertyValue =
"QuoteQ" ),
    @ActivationConfigProperty(propertyName = "connectionFactoryLookup" , propertyValue
= "QuoteCF" )}, mappedName = "LookupMDB" )
@TransactionAttribute(TransactionAttributeType.REQUIRED)
@TransactionManagement(TransactionManagementType.CONTAINER)
publicclass LookupMDB implements MessageListener {
```

## 関連情報

[インバウンド通信のリソース・アダプターの構成](#)

### WebSphere Application Server traditional 9.0 の関連情報

[IBM MQ メッセージング・プロバイダーの統合接続ファクトリーの構成](#)

[IBM MQ メッセージング・プロバイダーのトピックの構成](#)

[IBM MQ メッセージング・プロバイダーのアクティベーション・スペック](#)

[IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの作成](#)

[IBM MQ メッセージング・プロバイダーのアクティベーション・スペックの構成](#)

[IBM MQ メッセージング・プロバイダー JMS リソースのカスタム・プロパティの構成](#)

## 最新のリソース・アダプター保守レベルを使用するためのアプリケーション・サーバーの構成

WebSphere Application Server フィックスパックを適用したときに IBM MQ リソース・アダプターが入手可能な最新の保守レベルに自動更新されるようにするために、各ノードのインストール済み環境に適用した WebSphere Application Server フィックスパックに含まれている最新のバージョンのリソース・アダプターを使用するように、ご使用の環境内のすべてのサーバーを構成することができます。

### 始める前に

**重要:** いずれかのプラットフォームで WebSphere Application Server 7.0、8、または 8.5 を使用している場合は、IBM MQ 8.0 リソース・アダプターをアプリケーション・サーバーにインストールしないでください。IBM MQ 8.0 リソース・アダプターは、JMS 2.0 をサポートするアプリケーション・サーバーにのみデプロイできます。ただし、WebSphere Application Server 7.0、8、および 8.5 は、JMS 1.1 のみをサポートします。これらのバージョンの WebSphere Application Server には IBM WebSphere MQ 7.0 リソース・アダプターが付属しており、これを使用して BINDINGS または CLIENT トランスポートで IBM MQ 8.0 キュー・マネージャーに接続できます。

### このタスクについて

現在の構成に以下のいずれかの状況が当てはまり、最新のバージョンの IBM MQ リソース・アダプターを使用するよう環境内のすべてのサーバーを構成する場合は、このタスクを使用します。

- WebSphere Application Server 7.0.0 Fix Pack 1 以降が適用された後、環境内のどのアプリケーション・サーバーの JVM ログにも次の IBM MQ リソース・アダプター・バージョン情報が示されます。

```
WMSG1703I:RAR 実装バージョン 7.0.0.0-k700-L080820
```

- 環境内のどのアプリケーション・サーバーの JVM ログにも次の項目が含まれます。

```
WMSG1625E: 検出できませんでした
```

```
指定されたパス < null > にある IBM MQ メッセージング・プロバイダー・コード
```

- これまでは、特定の保守レベルの IBM MQ リソース・アダプターを使用するように 1 つ以上のノードが手動で更新されていましたが、現在は現行の WebSphere Application Server 保守レベルに含まれる最新バージョンのリソース・アダプターを使用するように変更されました。

例で参照されている `profile_root` ディレクトリーは、WebSphere Application Server プロファイルのホーム・ディレクトリーです (例: C:\Program Files\IBM\WebSphere\AppServer1)。

環境内のすべてのセルと単一サーバー・インストール済み環境に対して以下の手順を実行した場合、新しい WebSphere Application Server フィックスパックが適用されたときに、サーバーは IBM MQ リソース・アダプターに対する保守を自動的に受け取ります。

## 手順

1. アプリケーション・サーバーを始動します。プロファイルがネットワーク・デプロイメント構成の一部である場合、デプロイメント・マネージャーとすべてのノード・エージェントを始動します。プロファイルに管理エージェントが含まれる場合は、管理エージェントを始動します。
2. IBM MQ リソース・アダプターの保守レベルを確認します。
  - a) コマンド・プロンプト・ウィンドウを開き、`profile_root\bin` ディレクトリーに移動します。例えば、`cd C:\Program Files\IBM\WebSphere\AppServer1\bin` と入力します。
  - b) `wsadmin.bat -lang jython` と入力して `wsadmin` ツールを開始し、指示された場合にはユーザー名とパスワードを入力します。
  - c) 次のコマンドを入力してから、Return キーを 2 回押します。

```
wmqInfoMBeansUnsplit = AdminControl.queryNames("WebSphere:type=WMQInfo,*")
wmqInfoMBeansSplit = AdminUtilities.convertToList(wmqInfoMBeansUnsplit)
for wmqInfoMBean in wmqInfoMBeansSplit: print wmqInfoMBean; print AdminControl.invoke(wmqInfoMBean, 'getInfo', '')
```

このコマンドは Jacl でも実行できます。これを行う方法について詳しくは、WebSphere Application Server 製品資料の「使用可能な最新の IBM MQ リソース・アダプター保守レベルを使用するためのサーバーの構成」を参照してください。

- d) コマンドの出力表示から `WMSG1703I` メッセージを探し、リソース・アダプター・レベルを確認します。例えば、WebSphere Application Server 7.0.1 Fix Pack 5 の場合、メッセージは次のようになるはずでです。  
`WMSG1703I: RAR 実装バージョン 7.0.1.3-k701-103-100812`  
このメッセージは、バージョンがこのフィックスパックにおける正しいリソース・アダプター・レベルである `7.0.1.3-k701-103-100812` であることを示しています。ただし、代わりに以下のメッセージが表示された場合は、リソース・アダプターをフィックスパック 15 用の正しい保守レベルに調整する必要があることを意味します。  
`WMSG1703I: RAR 実装バージョン 7.0.0.0-k700-L080820`
3. 以下の Jython スクリプトを `convertWMQRA.py` というファイルにコピーし、それをプロファイル・ルート・ディレクトリー (例えば、`C:\Program Files\IBM\WebSphere\AppServer1\bin`) に保存します。

```
ras = AdminUtilities.convertToList(AdminConfig.list('J2CResourceAdapter'))

for ra in ras :
    desc = AdminConfig.showAttribute(ra, "description")
    if (desc == "WAS 7.0 Built In IBM MQ Resource Adapter") or (desc == "WAS 7.0.0.1 Built In IBM MQ Resource Adapter"):
        print "Updating archivePath and classpath of " + ra
        AdminConfig.modify(ra, [['archivePath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
        AdminConfig.unsetAttributes(ra, ['classpath'])
        AdminConfig.modify(ra, [['classpath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
        AdminConfig.save()
    #end if
#end for
```

**ヒント:** ファイルを保存するとき、テキスト・ファイルではなく python ファイルとして保存されていることを確認します。

4. ここで作成した Jython スクリプトを WebSphere Application Server `wsadmin` ツールを使用して実行します。  
コマンド・プロンプトを開き、WebSphere Application Server ディレクトリーのホーム・ディレクトリーにある `\bin` ディレクトリーに移動します。例えば、`C:\Program`

Files\IBM\WebSphere\AppServer1\bin ディレクトリーの場合は、以下のコマンドを入力して、Return キーを押します:

```
wsadmin -lang jython -f convertWMQRA.py
```

指示された場合にはユーザー名とパスワードを入力します。

**注:** ネットワーク・デプロイメント構成の一部であるプロファイルに対してスクリプトを実行すると、スクリプトはその構成で更新が必要なプロファイルをすべて更新します。既存の構成ファイルに不整合がある場合、完全な再同期が必要になる場合があります。

5. ネットワーク・デプロイメント構成で実行する場合、ノード・エージェントが完全に再同期されていることを確認します。詳しくは、「wsadmin スクリプト・ツールによるノードの同期化」または「ノードの追加、管理、および除去」を参照してください。
6. プロファイル内のすべてのサーバーを停止します。プロファイルがネットワーク・デプロイメント構成の一部である場合、構成内のすべてのクラスター・メンバーを停止し、構成内のすべてのノード・エージェントを停止し、さらにデプロイメント・マネージャーも停止します。プロファイルに管理エージェントが含まれる場合は、管理エージェントを停止します。
7. **osgiCfgInit** コマンドを *profile\_root/bin* ディレクトリーから実行します。  
osgiCfgInit コマンドは、OSGi ランタイム環境で使用されるクラス・キャッシュをリセットします。プロファイルがネットワーク・デプロイメント構成の一部である場合、構成の一部であるすべてのプロファイルの *profile\_root/bin* ディレクトリーから **osgiCfgInit** コマンドを実行します。
8. プロファイル内のすべてのサーバーを再始動します。プロファイルがネットワーク・デプロイメント構成の一部である場合、構成内のすべてのクラスター・メンバーを再始動し、構成内のすべてのノード・エージェントを再始動し、さらにデプロイメント・マネージャーも再始動します。プロファイルに管理エージェントが含まれる場合は、管理エージェントを再始動します。
9. 手順 2 を繰り返して、リソース・アダプターが正しいレベルになったことを確認します。

## 次のタスク

このトピックで説明されているステップを実行した後も引き続き問題が発生し、WebSphere Application Server 管理コンソールの「JMS プロバイダー設定」パネルの「リソース・アダプターの更新」ボタンを使用して、ご使用の環境内のすべてのノードで IBM MQ リソース・アダプターを更新した場合は、[APAR PM10308](#) で説明されている問題が発生している可能性があります。

### 関連情報

[IBM MQ リソース・アダプターの使用](#)

#### WebSphere Application Server 8.5.5 の関連情報

[使用可能な最新の IBM MQ リソース・アダプター保守レベルを使用するためのサーバーの構成](#)

[wsadmin スクリプト・ツールによるノードの同期化](#)

[ノードの追加、管理、および除去](#)

[JMS プロバイダー設定](#)

#### WebSphere Application Server 8.0 の関連情報

[使用可能な最新の IBM MQ リソース・アダプター保守レベルを使用するためのサーバーの構成](#)

[wsadmin スクリプト・ツールによるノードの同期化](#)

[ノードの追加、管理、および除去](#)

[JMS プロバイダー設定](#)

#### WebSphere Application Server 7.0 の関連情報

[使用可能な最新の IBM MQ リソース・アダプター保守レベルを使用するためのサーバーの構成](#)

[wsadmin スクリプト・ツールによるノードの同期化](#)

[ノードの追加、管理、および除去](#)

[JMS プロバイダー設定](#)

## JMS PROVIDERVERSION プロパティの構成

IBM MQ メッセージング・プロバイダーには、通常モード、制限付き通常モード、マイグレーション・モードという3つの操作モードがあります。JMS **PROVIDERVERSION** プロパティを設定して、これらのモードのうちJMSアプリケーションがパブリッシュおよびサブスクライブに使用するモードを選択できます。

### このタスクについて

IBM MQ メッセージング・プロバイダーの操作モードを選択するときには、主に **PROVIDERVERSION** 接続ファクトリー・プロパティを設定して制御できます。操作モードが指定されていない場合は、操作モードを自動的に選択することもできます。

**PROVIDERVERSION** プロパティによって、以下に示す3つのIBM MQ メッセージング・プロバイダー操作モードを区別します。

#### IBM MQ メッセージング・プロバイダーの通常モード

通常モードは、IBM MQ キュー・マネージャーのすべての機能を使用してJMSを実装します。このモードは、JMS 2.0のAPIと機能を使用するように最適化されています。

#### IBM MQ メッセージング・プロバイダーの制限付き通常モード

制限付き通常モードは、JMS 2.0のAPIを使用しますが、新機能(共有サブスクリプション、遅延送達、および非同期送信)は使用しません。

#### IBM MQ メッセージング・プロバイダーのマイグレーション・モード

移行モードを使用すると、IBM MQ 8.0以降のキュー・マネージャーに接続できますが、IBM WebSphere MQ 7.0以降のキュー・マネージャーの機能(先読みやストリーミングなど)はどれも使用されません。

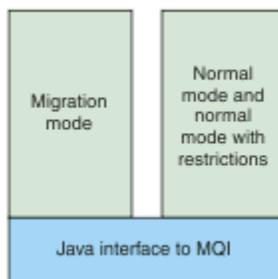


図 91. メッセージング・プロバイダー・モード

### 手順

特定の接続ファクトリーの **PROVIDERVERSION** プロパティを構成するには、次のようにします。

- IBM MQ Explorer を使用して **PROVIDERVERSION** プロパティを構成するには、[キュー・マネージャーおよびオブジェクトの構成](#)を参照してください。
- JMS 管理ツールを使用して **PROVIDERVERSION** プロパティを構成するには、[キュー・マネージャーおよびオブジェクトの構成](#)を参照してください。
- IBM JMS 拡張または IBM MQ JMS 拡張を使用して JMS アプリケーションで **PROVIDERVERSION** プロパティを構成するには、[IBM MQ classes for JMS アプリケーションでの接続ファクトリーおよび宛先の作成と構成](#)を参照してください。

JVM 内のすべての接続ファクトリーの接続ファクトリー・プロバイダー・モード設定を指定変更するには、次のようにします。

- 接続ファクトリー・プロバイダー・モード設定を指定変更するには、`com.ibm.msg.client.wmq.overrideProviderVersion` プロパティを使用します。  
使用している接続ファクトリーを変更できない場合、`com.ibm.msg.client.wmq.overrideProviderVersion` プロパティを使用して、接続ファクトリーでの設定を指定変更できます。この指定変更は JVM 中のすべての接続ファクトリーに適用されますが、実際の接続ファクトリー・オブジェクトは変更されません。

## 関連情報

### PROVIDERVERSION

[JMS プロバイダー・バージョンのトラブルシューティング](#)

[接続ファクトリーのプロパティ](#)

[IBM MQ classes for JMS オブジェクトのプロパティ間の依存関係](#)

## IBM MQ メッセージング・プロバイダーの操作のモード

接続ファクトリーの PROVIDERVERSION プロパティに適切な値を設定することで、JMS アプリケーションのパブリッシュおよびサブスクライブに使用する IBM MQ メッセージング・プロバイダーの操作モードを選択できます。場合によっては、PROVIDERVERSION プロパティを指定なしに設定します。このようにすると、JMS クライアントではアルゴリズムを使って、使用する操作モードを判別します。

### PROVIDERVERSION プロパティ値

接続ファクトリーの **PROVIDERVERSION** プロパティは、以下のいずれかの値に設定できます。

#### 8 - 通常モード

JMS アプリケーションは、通常モードを使用します。このモードは、IBM MQ キュー・マネージャーのすべての機能を使用して JMS を実装します。

#### 7 - 制限付き通常モード

JMS アプリケーションは、制限付き通常モードを使用します。このモードは JMS 2.0 API を使用しますが、共用サブスクリプション、遅延送達、非同期送信などの新機能は使用しません。

#### 6 - 移行モード

JMS アプリケーションは、移行モードを使用します。移行モードでは、IBM MQ classes for JMS は、IBM WebSphere MQ 6.0 に付属の機能やアルゴリズムに似た機能やアルゴリズムを使用します。

#### unspecified (デフォルト値)

JMS クライアントでは、アルゴリズムを使って、使用する操作モードを判別します。

**PROVIDERVERSION** プロパティに指定する値は、ストリングでなければなりません。オプション 8、7、または 6 を指定する場合、次のいずれかのフォーマットでこれを行えます。

- V.R.M.F
- V.R.M
- V.R
- V

ここで、V、R、M、および F は、ゼロ以上の整数値です。追加の R、M、および F の値は任意指定で、細かい制御が必要な場合に使用できます。例えば、**PROVIDERVERSION** レベル 7 を使用する場合は、**PROVIDERVERSION** = 7、7.0、7.0.0 または 7.0.0.0 を設定できます。

### 接続ファクトリー・オブジェクトのタイプ

**PROVIDERVERSION** プロパティは、以下の接続ファクトリー・オブジェクト・タイプに設定できます。

- MQConnectionFactory
- MQQueueConnectionFactory
- MQTopicConnectionFactory
- MQXAConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXATopicConnectionFactory

これらのさまざまなタイプの接続ファクトリーについて詳しくは、571 ページの『管理ツールを使用した JMS オブジェクトの構成』を参照してください。

## 関連情報

[IBM MQ classes for JMS のアーキテクチャー](#)

### PROVIDERVERSION 通常モード

通常モードは、IBM MQ キュー・マネージャーのすべての機能を使用して JMS を実装します。このモードは、JMS 2.0 の API と機能を使用するように最適化されています。

以下のフローチャートは、通常モードの接続を作成できるかどうかを判断するために JMS クライアントが実行する検査を示しています。

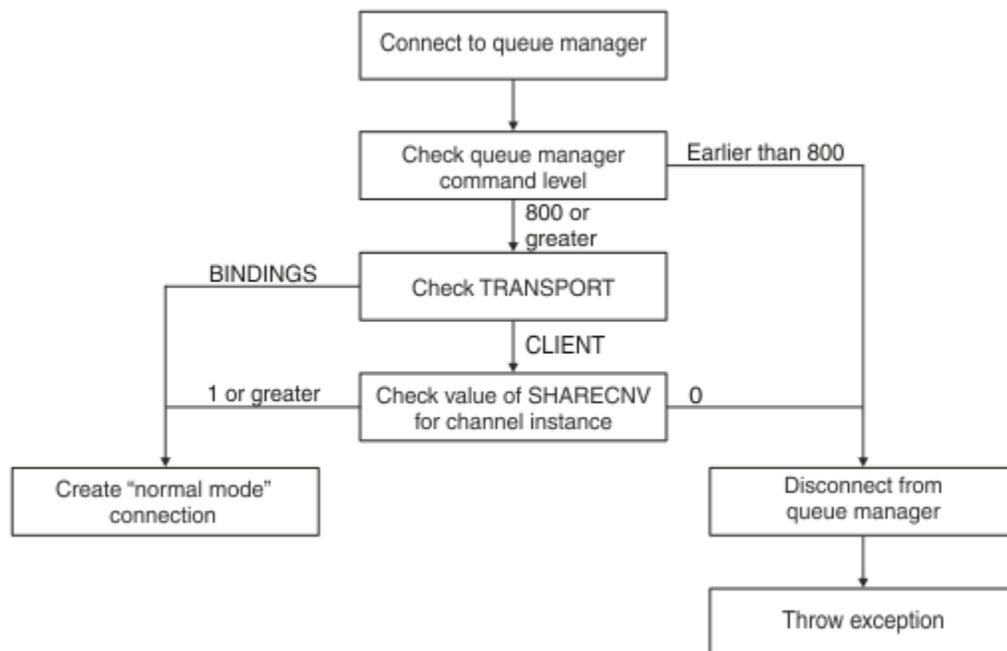


図 92. PROVIDERVERSION 通常モード

接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 800 以上で、接続ファクトリーの **TRANSPORT** プロパティが **BINDINGS** に設定されている場合は、その他のプロパティを検査することなく、通常モードの接続が作成されます。

接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 800 以上で、**TRANSPORT** プロパティが **CLIENT** に設定されている場合は、サーバー接続チャンネルの **SHARECNV** プロパティも検査されます。この検査が必要になるのは、IBM MQ メッセージング・プロバイダー通常モードで共有会話機能を使用するためです。したがって、通常モードの接続の試みを成功させるには、共有可能な会話の数を制御する **SHARECNV** プロパティの値が 1 以上でなければなりません。

フローチャートに示されたすべての検査が成功すると、キュー・マネージャーへの通常モードの接続が作成され、JMS 2.0 のすべての API および機能（つまり、非同期送信、遅延送達、および共有サブスクリプション）を使用できるようになります。

通常モードの接続を作成する試みの失敗は、以下のいずれかの理由によります。

- 接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 800 未満である。この場合、`createConnection` メソッドは例外 `JMSFMQ0003` で失敗します。
- サーバー接続チャンネル上の **SHARECNV** プロパティは、0 に設定されます。プロパティの値が 1 以上でない場合、`createConnection` メソッドは例外 `JMSCC5007` で失敗します。

## 関連情報

[IBM MQ classes for JMS オブジェクトのプロパティ間の依存関係](#)

## DEFINE CHANNEL (SHARECNV プロパティ)

### TRANSPORT

#### PROVIDERVERSION 制限付き通常モード

制限付き通常モードでは、JMS 2.0 API を使用しますが、共有サブスクリプション、遅延送達、非同期送信などの IBM MQ 8.0 以降の新機能は使用しません。

以下のフローチャートは、制限付き通常モードの接続を作成できるかどうかを判断するために JMS クライアントが実行する検査を示しています。

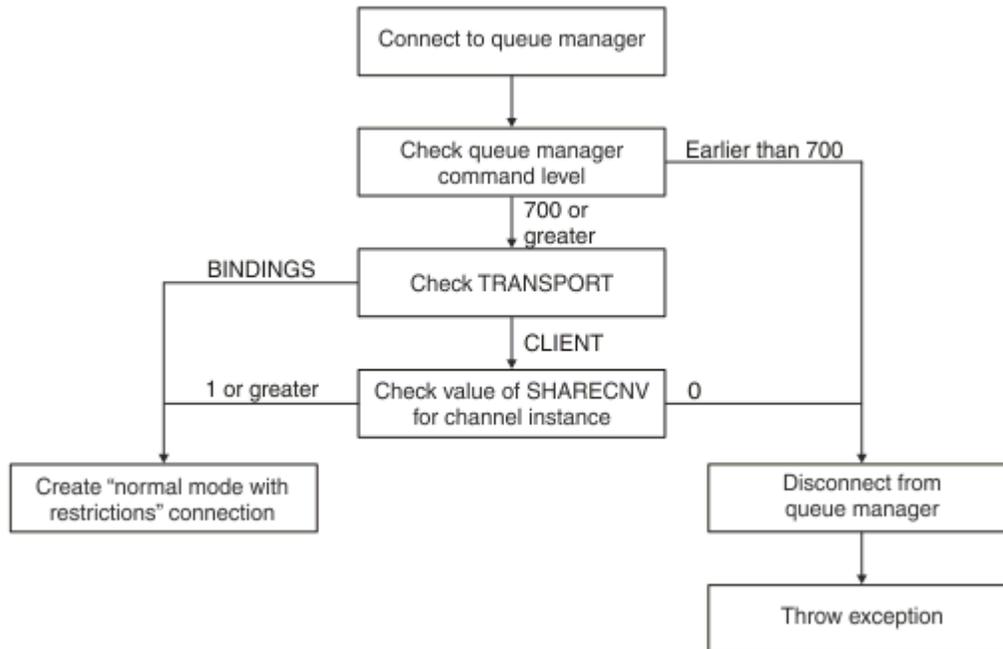


図 93. PROVIDERVERSION 制限付き通常モード

接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 700 以上で、接続ファクトリーの **TRANSPORT** プロパティが BINDINGS に設定されている場合は、その他のプロパティを検査することなく、通常モードの接続が作成されます。

接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 700 以上で、**TRANSPORT** プロパティが CLIENT に設定されている場合は、サーバー接続チャンネルの **SHARECNV** プロパティも検査されます。この検査が必要になるのは、IBM MQ メッセージング・プロバイダー制限付き通常モードで共有会話機能を使用するためです。したがって、制限付き通常モードの接続の試みを成功させるには、共有可能な会話の数を制御する **SHARECNV** プロパティの値が 1 以上でなければなりません。

フローチャートに示されたすべての検査が成功すると、キュー・マネージャーへの制限付き通常モードの接続が作成され、JMS 2.0 API を使用できます。しかし、非同期送信、遅延送達、または共有サブスクリプションの機能は使用できません。

制限付き通常モードの接続を作成する試みの失敗は、以下のいずれかの理由によります。

- 接続ファクトリー設定で指定されたキュー・マネージャーのコマンド・レベルが 700 未満である。この場合、createConnection メソッドは、例外 JMSFCC5008 で失敗します。
- サーバー接続チャンネル上の **SHARECNV** プロパティは、0 に設定されます。プロパティの値が 1 以上でない場合、createConnection メソッドは例外 JMSSC5007 で失敗します。

#### 関連情報

IBM MQ classes for JMS オブジェクトのプロパティ間の依存関係

## DEFINE CHANNEL (SHARECNV プロパティ)

### TRANSPORT

## PROVIDERVERSION 移行モード

移行モードの場合、IBM MQ classes for JMS は、IBM WebSphere MQ 6.0 に付属の機能やアルゴリズムに似た機能やアルゴリズムを使用します。それには、キューに入っているパブリッシュ/サブスクライブ、クライアント・サイドに実装される選択項目、非マルチプレクス・チャンネル、リスナーの実装に使用するポーリングなどがあります。

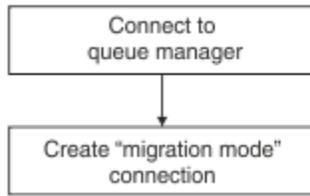


図 94. PROVIDERVERSION 移行モード

IBM MQ Enterprise Transport 6.0 を使用して WebSphere Message Broker 6.0 または 6.1 に接続する場合は、マイグレーション・モードを使用する必要があります。

移行モードを使用して IBM MQ 8.0 キュー・マネージャーに接続できますが、IBM MQ classes for JMS キュー・マネージャーの新機能 (先読みやストリーミングなど) はいずれも使用されません。IBM MQ 8.0 以降のクライアントが分散プラットフォーム上の IBM MQ 8.0 以降のキュー・マネージャー 、または z/OS 上の IBM MQ 8.0 以降のキュー・マネージャーに接続している場合、メッセージの選択は、クライアント・システムではなくキュー・マネージャーによって行われます。

IBM MQ メッセージング・プロバイダー移行モードが指定されていて、IBM MQ classes for JMS で JMS 2.0 のいずれかの API を使用しようとした場合、API メソッド呼び出しは例外 JM5CC5007 で失敗します。

### 関連情報

[IBM MQ classes for JMS オブジェクトのプロパティ間の依存関係](#)

[TRANSPORT](#)

### PROVIDERVERSION の指定なし

接続ファクトリーの **PROVIDERVERSION** プロパティが指定なしの場合、JMS クライアントでは、アルゴリズムを使用して、キュー・マネージャーへの接続に使う操作モードが判別されます。以前のバージョンの IBM MQ classes for JMS を使用して JNDI 名前空間に作成された接続ファクトリーが、新しいバージョンの IBM MQ classes for JMS で使用されるときには、指定なしの値を取ります。

**PROVIDERVERSION** プロパティが指定なしの場合、`createConnection` メソッドが呼び出されるときにアルゴリズムが使用されます。このアルゴリズムは、接続ファクトリーの複数のプロパティを検査して、IBM MQ メッセージング・プロバイダー通常モード、制限付き通常モード、または IBM MQ メッセージング・プロバイダー移行モードのどれが必要かを判別します。常に通常モードが最初に試行され、次に制限付き通常モードが試行されます。どちらのタイプの接続も作成できない場合は、JMS クライアントはキュー・マネージャーから切断してから、キュー・マネージャーに再接続し、移行モードの接続を試行します。

### BROKERVER、BROKERQMGR、PSMODE、BROKERCONQ プロパティの検査

図 1 に示されているように、最初に **BROKERVER** プロパティの値が検査されます。

**BROKERVER** プロパティが V1 に設定されている場合、図 2 に示すように **TRANSPORT** プロパティが次にチェックされます。ただし、**BROKERVER** プロパティが V2 に設定されている場合は、図 1 に示す追加のチェックは、**TRANSPORT** プロパティがチェックされる前に実行されます。

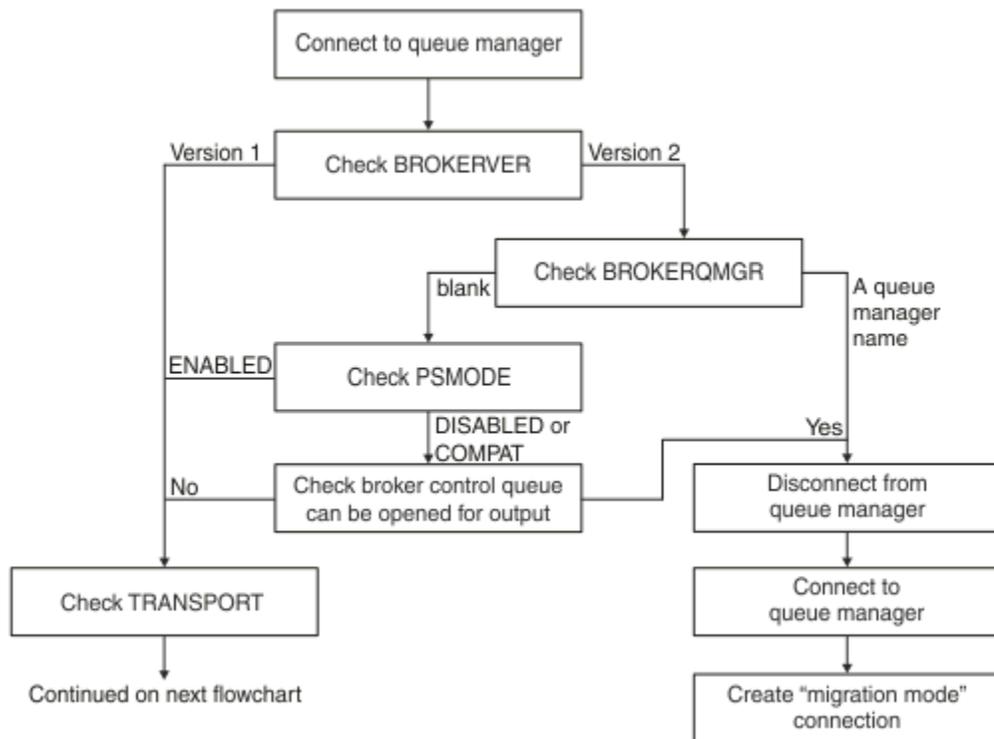


図 95. PROVIDERVERSION の指定なし

**BROKERVER** プロパティが V2 に設定されている場合、通常モードの接続を行えるようにするには、**BROKERQMGR** プロパティが空白でなければなりません。さらに、キュー・マネージャー上の **PSMODE** 属性が **ENABLED** に設定されているか、**BROKERCONQ** プロパティで指定されたブローカー制御キューが出力用に開くことができないようになっていなければなりません。

通常モード接続でプロパティ値が必要に応じて設定されている場合は、次にチェック・マークを付けると、[図 2](#) に示すように **TRANSPORT** プロパティに移動します。

通常モード接続にとって必要なプロパティ値が設定されていない場合は、JMS クライアントはキュー・マネージャーから切断してから再接続して、移行モードの接続を作成します。これは次のような場合に行われます。

- **BROKERQMGR** プロパティが空白で、キュー・マネージャー上の **PSMODE** 属性が **COMPAT** または **DISABLED** に設定されており、**BROKERCONQ** プロパティで指定されたブローカー制御キューを出力用に開くことができる (つまり、出力の **MQOPEN** が正常に実行される) 場合。
- **BROKERQMGR** プロパティでキュー名が指定されている場合。

## TRANSPORT プロパティとコマンド・レベルの検査

[図 2](#) には、**TRANSPORT** プロパティと、キュー・マネージャーのコマンド・レベルを対象に行う検査が示されています。

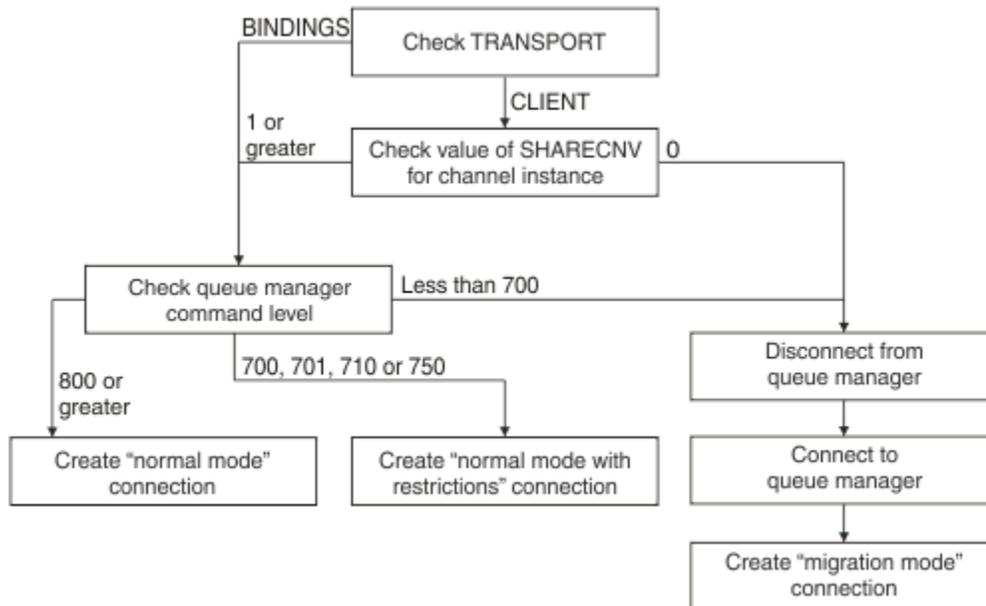


図 96. PROVIDERVERSION の指定なし (続き)

以下のいずれかの場合に、通常モードの接続が作成されます。

- 接続ファクトリーの **TRANSPORT** プロパティが BINDINGS に設定され、キュー・マネージャーのコマンド・レベルが 800 以上の場合。
- **TRANSPORT** プロパティが CLIENT に設定され、サーバー接続チャンネル上の **SHARECNV** プロパティの値が 1 以上で、キュー・マネージャーのコマンド・レベルが 800 以上の場合。

キュー・マネージャーのコマンド・レベルが 700、701、710、または 750 の場合は、キュー・マネージャーに対する制限付き通常モードの接続が作成されます。

**TRANSPORT** プロパティが CLIENT に設定され、サーバー接続チャンネル上の **SHARECNV** プロパティの値が 0 の場合も、移行モードの接続が作成されます。

#### 関連情報

[IBM MQ classes for JMS オブジェクトのプロパティ間の依存関係](#)

[ALTER QMGR \(PSMODE 属性\)](#)

[BROKERCONQ](#)

[BROKERQMGR](#)

[BROKERVER](#)

[DEFINE CHANNEL \(SHARECNV プロパティ\)](#)

[TRANSPORT](#)

## PROVIDERVERSION のデフォルト設定を指定変更する場合

以前のバージョンの IBM MQ classes for JMS を使用して JNDI 名前空間に作成された接続ファクトリーを新しいバージョンの IBM MQ classes for JMS で使用すると、接続ファクトリーの **PROVIDERVERSION** プロパティはデフォルト値 **unspecified** に設定され、使用する IBM MQ メッセージング・プロバイダーの操作モードを判別するアルゴリズムが使用されます。ただし、IBM MQ classes for JMS を正しく機能させるため、**PROVIDERVERSION** プロパティのデフォルト選択を指定変更しなければならない場合が 2 つあります。

**注:** このトピックで説明するマイグレーション・モードは、IBM WebSphere MQ 6.0 から 7.0 へのマイグレーション用です。それより後のリリースからの移行には適用されません。

IBM WebSphere MQ 6.0、WebSphere Application Server 6.0.x、および WebSphere Message Broker 6 はサポート対象外です。したがって、このトピックは参照用により掲載されています。

**PROVIDERVERSION** プロパティがデフォルトの `unspecified` に設定されている場合、599 ページの『**PROVIDERVERSION** の指定なし』で説明されているように、使用する操作モードを判別するアルゴリズムが使用されます。ただし、このアルゴリズムは以下の 2 つのシナリオでは使用できません。

1. WebSphere Message Broker および WebSphere Event Broker が互換モードの場合、WebSphere Message Broker および WebSphere Event Broker が正しく機能するためには、**PROVIDERVERSION** プロパティの値を指定する必要があります。
2. WebSphere Application Server 6.0.1、6.0.2 または 6.1 を使用している場合は、WebSphere Application Server 管理コンソールを使用して接続ファクトリーを定義します。

WebSphere Application Server では、接続ファクトリーの **BROKERVER** プロパティのデフォルト値は V2 です。JMS 管理ツール **JMSAdmin** または IBM MQ Explorer を使用して作成される接続ファクトリーの **BROKERVER** プロパティのデフォルト値は、V1 です。IBM MQ では、このプロパティは現在 `unspecified` です。

**BROKERVER** プロパティが V2 に設定されている場合 (以前に WebSphere Application Server によって作成されたか、以前にパブリッシュ/サブスクライブに接続ファクトリーが使用されており、**BROKERCONQ** プロパティが定義されている既存のキュー・マネージャーがあるため)、IBM MQ メッセージング・プロバイダー移行モードが使用されます。

ただし、アプリケーションでピアツーピア通信を使用する必要があり、パブリッシュ/サブスクライブに使用されたことがある既存のキュー・マネージャーをアプリケーションが使用しており、接続ファクトリーが WebSphere Application Server で作成された場合のデフォルト設定である **BROKERVER** が 2 に設定された接続ファクトリーがある場合は、IBM MQ メッセージング・プロバイダー移行モードが使用されます。この場合に IBM MQ メッセージング・プロバイダー移行モードを使用する必要はありません。代わりに IBM MQ メッセージング・プロバイダー通常モードを使用してください。この回避策として、以下のいずれかの方法を使用できます。

- **BROKERVER** を 1 または `unspecified` に設定します。選択するオプションは、ご使用のアプリケーションによって異なります。
- **PROVIDERVERSION** を、8 または 7 に設定します。これらは、WebSphere Application Server 6.1 のカスタム・プロパティです。

あるいは、クライアント構成プロパティを使用するか、**BROKERCONQ** プロパティが設定されないように接続先のキュー・マネージャーを変更するか、キューを使用不可にします。

## WebSphere Application Server でのプロバイダー・バージョン情報の構成

WebSphere Application Server でプロバイダー・バージョン情報を構成するときに、管理コンソールまたは `wsadmin` コマンドを使用できます。

### 手順

IBM MQ 接続ファクトリーまたはアクティベーション・スペック・オブジェクトのプロバイダー・バージョン情報を WebSphere Application Server で構成するには、WebSphere Application Server 製品資料内の詳細情報へのリンクを「関連情報」で参照してください。

#### WebSphere Application Server 8.5.5 の関連情報

[IBM MQ メッセージング・プロバイダー接続ファクトリー設定](#)

[createWMQConnectionFactory コマンド](#)

[IBM MQ メッセージング・プロバイダー・アクティベーション・スペック設定](#)

[createWMQActivationSpec コマンド](#)

#### WebSphere Application Server 8.0.0 の関連情報

[IBM MQ メッセージング・プロバイダー接続ファクトリー設定](#)

[createWMQConnectionFactory コマンド](#)

[IBM MQ アクティベーション・スペック設定](#)

[createWMQActivationSpec コマンド](#)

## WebSphere Application Server 7.0.0 の関連情報

[IBM MQ メッセージング・プロバイダー接続ファクトリー設定](#)

[createWMQConnectionFactory コマンド](#)

[IBM MQ アクティベーション・スペック設定](#)

[createWMQActivationSpec コマンド](#)

## WebSphere Application Server 永続サブスクリプションの削除

WebSphere Application Server 7.0 および 8.0 で IBM MQ メッセージング・プロバイダーを使用する場合、アクティベーション・スペックにバインドされたメッセージ駆動型 Bean アプリケーションによって作成された永続サブスクリプションは削除されません。永続サブスクリプションは、IBM MQ Explorer または IBM MQ コマンド・ライン・ユーティリティーのいずれかを使用して削除できます。

### 本タスクについて

永続サブスクリプションを除去するメッセージ駆動型 Bean アプリケーションは、リスナー・ポートまたはアクティベーション・スペックのいずれかを使用するように構成できます。ただし、アプリケーションが [WebSphere MQ メッセージング・プロバイダー通常モード](#) を使用して IBM MQ に接続する WebSphere Application Server 7.0 または 8.0 インスタンス内で実行されている場合に限りです。

メッセージ駆動型 Bean アプリケーションがリスナー・ポートにバインドされている場合、IBM MQ メッセージング・プロバイダーが、アプリケーションの初回始動時にアプリケーション用の永続サブスクリプションを作成します。永続サブスクリプションは、メッセージ駆動型 Bean アプリケーションがアプリケーション・サーバーからアンインストールされ、アプリケーション・サーバーが再始動すると削除されます。

活動化仕様にバインドされたメッセージ駆動型 Bean アプリケーションの動作方法は、若干異なります。アプリケーション用の永続サブスクリプションは、アプリケーションの初回始動時に作成されます。ただし、永続サブスクリプションは、アプリケーションがアンインストールされて、アプリケーション・サーバーが再始動しても削除されません。

このため、WebSphere Application Server システムにもうインストールされていないアプリケーションの IBM MQ パブリッシュ/サブスクライブ・エンジンに、いくつかの永続サブスクリプションが残されることになる場合があります。このようなサブスクリプションは「オフファン・サブスクリプション」と呼ばれ、パブリッシュ/サブスクライブ・エンジンの実行時にキュー・マネージャー上で問題を生じさせる可能性があります。

トピックに関するメッセージがパブリッシュされると、IBM MQ パブリッシュ/サブスクライブ・エンジンが、そのトピックに登録されている各永続サブスクリプション用にメッセージのコピーを作成し、内部キューに配置します。その永続サブスクリプションを使用しているアプリケーションは、この内部キューからメッセージを取り出して消費します。

その永続サブスクリプションを使用していたメッセージ駆動型 Bean アプリケーションがもはやインストールされていない場合であっても、そのアプリケーションに対してパブリッシュされたメッセージのコピーは引き続き作成されます。しかし、これらのメッセージは処理されることがなく、つまり削除されることのない大量のメッセージが内部キューに残される可能性があります。

### 開始前に

IBM MQ パブリッシュ/サブスクライブ・エンジンに登録されたサブスクリプションには、関連付けられたサブスクリプション名があります。

活動化仕様にバインドされたメッセージ駆動型 Bean 用の WebSphere Application Server IBM MQ メッセージング・プロバイダーによって作成された永続サブスクリプションには、以下のフォーマットのサブスクリプション名が付けられます。

```
JMS:queue manager name:client identifier:subscription name
```

説明

### **queue manager name**

これは、パブリッシュ/サブスクライブ・エンジンが実行されている IBM MQ キュー・マネージャーの名前です。

### **client identifier**

これは、メッセージ駆動型 Bean がバインドされた活動化仕様のクライアント ID プロパティの値です。

### **subscription name**

これは、メッセージ駆動型 Bean アプリケーションが使用するよう構成されている活動化仕様の、活動化仕様プロパティのサブスクリプション名の値です。

例えば、キュー・マネージャー testQM に接続するためにセットアップされた活動化仕様があるとします。この活動化仕様のプロパティ・セットは次のようになります。

- クライアント ID = testClientID
- サブスクリプション名 = durableSubscription1

永続サブスクリプションを取得するメッセージ駆動型 Bean がこの活動化仕様にバインドされている場合、キュー・マネージャー testQM の IBM MQ パブリッシュ/サブスクライブ・エンジンで、以下のサブスクリプション名を持つサブスクリプションが作成されます。

- JMS:testQM:testClientID:durableSubscription1

所定のキュー・マネージャー用の IBM MQ パブリッシュ/サブスクライブ・エンジンに登録されているサブスクリプションは、以下の 2 つの方法のいずれかで確認できます。

- 最初のオプションは、IBM MQ Explorer を使用する方法です。パブリッシュ/サブスクライブ作業に使用されているキュー・マネージャーに IBM MQ Explorer が接続されている場合、ナビゲーション・ペインの IBM WebSphere MQ ->queue manager name-> Subscriptions 項目をクリックすると、パブリッシュ/サブスクライブ・エンジンに現在登録されているサブスクライバーのリストを表示できます。
- パブリッシュ/サブスクライブ・エンジンに登録されているサブスクリプションを表示するもう 1 つの方法は、IBM MQ コマンド・ライン・ユーティリティー **runmqsc** を使用して、コマンド **display sub** を実行することです。これを行うには、コマンド・プロンプトを表示し、*WebSphere MQ\bin* ディレクトリに移動して、以下のコマンドを入力して **runmqsc** を開始します。

- runmqsc queue manager name

**runmqsc** ユーティリティーが始動したら、以下のコマンドを入力して、**runmqsc** が接続されているキュー・マネージャーで実行しているパブリッシュ/サブスクライブ・エンジンに現在登録されているすべての永続サブスクリプションをリストします。

- display sub(\*) durable

パブリッシュ/サブスクライブ・エンジンに登録されている永続サブスクリプションがまだアクティブであるかどうかを調べるには、以下のようになります。

1. パブリッシュ/サブスクライブ・エンジンに登録されている永続サブスクリプションのリストを生成します。
2. 各永続サブスクリプションで、以下を行います。
  - 永続サブスクライバーのサブスクリプション名を調べ、*client identifier* と *subscription name* の値をメモします。
  - このパブリッシュ/サブスクライブ・エンジンに接続されている WebSphere Application Server システムを調べます。*client identifier* 値に一致するクライアント ID プロパティと、*subscription name* と一致するサブスクリプション名プロパティを持つ活動化仕様が定義されていないかどうかを調べます。
  - IBM MQ サブスクリプション名の *client identifier* フィールドと *subscription name* フィールドに一致するクライアント ID プロパティとサブスクリプション名プロパティを持つ活動化仕様が見つからない場合、この永続サブスクリプションを使用している活動化仕様はありません。その永続サブスクリプションは削除できます。

- 永続サブスクリプション名に一致する活動化仕様が定義されている場合、必要な最終検査は、この活動化仕様を使用しているメッセージ駆動型 Bean アプリケーションがないかどうかを調べることで、そのためには、次のようにします。
  - 現在確認している永続サブスクリプションを取得している活動化仕様の JNDI 名をメモします。
  - インストールされている各メッセージ駆動型 Bean アプリケーションの WebSphere Application Server 管理コンソールで「構成」ペインを起動します。
  - 「構成」ペイン内のメッセージ駆動型 Bean リスナーのバインディング・リンクをクリックします。
  - メッセージ駆動型 Bean アプリケーションに関する情報を示した表が表示されます。「バインディング」列で活動化仕様ラジオ・ボタンが選択され、「ターゲット・リソース JNDI 名」フィールドに永続サブスクリプションを取得した活動化仕様の JNDI 名が指定されている場合、そのサブスクリプションはまだ使用されており、削除できません。
  - 活動化仕様を使用しているメッセージ駆動型 Bean アプリケーションが見つからない場合は、その永続サブスクリプションを削除できます。

## 手順

「孤立した」永続サブスクリプションが識別されると、IBM MQ Explorer または IBM MQ コマンド・ライン・ユーティリティー **runmqsc** を使用して削除できます。

「オーファン化した」永続サブスクリプションを IBM MQ Explorer を使用して削除する手順は、以下のとおりです。

1. サブスクリプション用の項目を強調表示します。
2. エントリーを右クリックし、「削除 ...」を選択します。表示されます。確認ウィンドウが表示されません。
3. 確認ウィンドウに表示されているサブスクリプション名が正しいことを確認し、「はい」をクリックします。

IBM MQ Explorer はパブリッシュ/サブスクライブ・エンジンからサブスクリプションを削除し、それに関連するすべての内部リソース (その永続サブスクリプションが登録されていたトピックに関してパブリッシュされた未処理のメッセージなど) をクリーンアップします。

IBM MQ コマンド・ライン・ユーティリティー **runmqsc** を使用して「孤立した」永続サブスクリプションを削除するには、コマンド **delete sub** を実行する必要があります。

1. コマンド・プロンプト・セッションを開きます。
2. *WebSphere MQ\bin* ディレクトリーに移動します
3. 以下のコマンドを入力して **runmqsc** を始動します。

```
runmqsc queue manager name
```

4. **runmqsc** ユーティリティーが始動したら、以下を入力します。

```
delete sub(Subscription name)
```

ここで、*Subscription name* は、永続サブスクリプションの次の形式のサブスクリプション名です。

- *JMS:queue manager name:client identifier:subscription name*

## V 9.0.1 IBM MQ Console および REST API の構成

IBM MQ Console と REST API をホストする mqweb サーバーには、デフォルトの構成が準備されています。どちらのコンポーネントを使用する場合も、ユーザーにログインを許可するために、セキュリティを構成するなどの複数の構成タスクを実行する必要があります。このトピックでは、使用可能なすべての構成オプションについて説明します。

## 手順

- [606 ページの『セキュリティーの構成』](#)
- [607 ページの『HTTP ホスト名の構成』](#)
- [609 ページの『HTTP および HTTPS ポートの構成』](#)
- [610 ページの『応答タイムアウトの構成』](#)
- [611 ページの『自動始動の構成』](#)
- [612 ページの『ロギングの構成』](#)
- [615 ページの『LTPA トークンの有効期間の構成』](#)
- [617 ページの『messaging REST API の構成』](#)
- [606 ページの『CSRF 保護の構成』](#)

### V 9.0.1 セキュリティーの構成

mqwebuser.xml ファイルを編集して、IBM MQ Console および REST API のセキュリティーを構成できます。基本ユーザー・レジストリー、LDAP レジストリー、または WebSphere Application Server Liberty が備える他のレジストリー・タイプを構成することで、ユーザーを構成し、認証することができます。そして、ユーザーとグループに役割を割り当てることで、それらのユーザーに権限を与えることができます。IBM MQ 9.0.1 には、REST API のためのセキュリティーがありません。IBM MQ 9.0.2 以降では、REST API のためのセキュリティーを構成します。

## このタスクについて

IBM MQ Console、および REST API のためのセキュリティーを構成するには、ユーザーとグループを構成する必要があります。そして、それらのユーザーとグループに、IBM MQ Console を使用するための権限、または REST API、あるいはその両方を与えることができます。ユーザーとグループを構成する方法、およびユーザーを認証して権限を与える方法について詳しくは、[IBM MQ Console および REST API のセキュリティー](#)を参照してください。

ユーザーが IBM MQ Console で認証されると、LTPA トークンが生成されます。REST API でトークン・ベースの認証を使用する場合、HTTP POST メソッドで /login REST API リソースを使用してユーザーがログインする際に別の LTPA トークンが生成されます。このトークンにより、ユーザーは、トークンの有効期限が切れるまで再認証することなく IBM MQ Console を使用できます。トークンの有効期限は構成できます。詳しくは、[615 ページの『LTPA トークンの有効期間の構成』](#)を参照してください。

## 手順

- [IBM MQ Console および REST API セキュリティー](#)
- [615 ページの『LTPA トークンの有効期間の構成』](#)

### V 9.0.4 CSRF 保護の構成

クロスサイト・リクエスト・フォージェリー (CSRF) は、悪意のある Web サイトにより、ユーザーが現在認証されている信頼できるサイトでユーザーのブラウザーが不要なアクションを実行する場合に発生する攻撃のタイプです。

## 始める前に

この手順を実行するには、[特権ユーザー](#)である必要があります。

**V 9.0.4** 以下のコマンドを使用して、CSRF 保護の現在の構成を表示できます。

```
dspmqweb properties -a
```

mqRestCsrftValidation フィールドは、CSRF 検証チェックが実行されるかどうかを示します。詳しくは、[dspmqweb](#)を参照してください。

注: **V 9.0.5** CSRF 有効期限を示すために IBM MQ 9.0.4 で導入された `mqRestCsrfsExpirationInMinutes` フィールドは、IBM MQ 9.0.5 には存在しなくなりました。



重要: **z/OS** **V 9.0.4**

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行するには、その前に `WLP_USER_DIR` 環境変数を設定し、この変数が `mqweb` サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

`WLP_user_directory` は、`crtmqweb.sh` に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

## このタスクについて

**V 9.0.5** IBM MQ 9.0.5 より前は、IBM MQ Console と REST API でシンクロナイザー・トークンを使用して、CSRF 攻撃からシステムを保護していました。IBM MQ 9.0.4 に限って言えば、管理 REST API の CSRF シンクロナイザー・トークンが周期的に再生成されていました。IBM MQ 9.0.5 からは、CSRF シンクロナイザー・トークンが使用されなくなっています。その代わりに、カスタム HTTP ヘッダーを設定して、シンクロナイザー・トークンの使用と同等の保護機能を用意する必要があります。

**setmqweb properties** コマンドを使用して、REST API の CSRF 保護の構成を変更できます。

## 手順

- 以下の方法を使用して、REST API の CSRF トークン検証を構成します。
  - IBM MQ 9.0.4 限定の機能として、トークンの有効期限を変更するために、**setmqweb properties** コマンドを使用できます。

```
setmqweb properties -k mqRestCsrfsExpirationInMinutes -v time
```

ここで、`time` は、CSRF トークンの有効期限が切れるまでの時間を分数で指定します。トークンは、有効期限が切れた後も次の HTTP POST、PATCH、または DELETE メソッドで有効です。その後は、新しいトークンが Cookie として返され、前のトークン値が無効になります。時間値として `-1` を指定すると、CSRF トークンの有効期限が無効になります。一方、`0` を指定すると、POST、PATCH、DELETE の各要求の実行ごとにトークンが変更されます。デフォルト値は 30 分です。

- CSRF 検証チェックを除去するには、**setmqweb properties** コマンドを使用します

```
setmqweb properties -k mqRestCsrfsValidation -v boolean
```

ここで、`boolean` は、CSRF 検証チェックを実行するかどうかを指定します。値が `false` の場合、CSRF トークン検証チェックは除去されます。特にユーザーが Web ブラウザーを使用して REST API にアクセスする場合には、トークンの検証が推奨されます。デフォルト値は `true` で、CSRF トークンは、REST API 経由でのすべての HTTP POST、PATCH、および DELETE 要求について検証されます。

## **V 9.0.1** HTTP ホスト名の構成

デフォルトでは、IBM MQ Console および REST API をホストする `mqweb` サーバーは、ローカル接続のみを許可するように構成されます。つまり、IBM MQ Console と REST API は、IBM MQ Console と REST API がインストールされているシステム上でのみ利用できます。**V 9.0.4** IBM MQ 9.0.4 以降では、**setmqweb** コマンドを使用して、リモート接続を許可するようにホスト名を構成できます。IBM MQ 9.0.3 以前では、`mqwebuser.xml` ファイルを編集することにより、リモート接続を許可するようにホスト名を構成できます。

## 始める前に

この手順を実行するには、[特権ユーザー](#)である必要があります。

**V 9.0.4** IBM MQ 9.0.4 以降では、以下のコマンドを使用して、HTTP ホスト名の現在の構成を表示できます。

```
dspmweb properties -a
```

httpHost フィールドには、HTTP ホスト名が表示されます。詳しくは、[dspmweb](#) を参照してください。



**重要:** **z/OS** **V 9.0.4**

z/OS で **setmqweb** コマンドまたは **dspmweb** コマンドを発行するには、その前に WLP\_USER\_DIR 環境変数を設定し、この変数が mqweb サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

WLP\_user\_directory は、crtmqweb.sh に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

## 手順

### • **V 9.0.4**

以下のいずれかの方法を使用して、ホスト名を構成します。

- IBM MQ 9.0.4 以降では、**setmqweb properties** コマンドを使用します。

```
setmqweb properties -k httpHost -v hostName
```

hostName は IP アドレス、ドメイン名サフィックス付きのドメイン・ネーム・サーバー (DNS) ホスト名、または IBM MQ がインストールされているサーバーの DNS ホスト名を示します。使用可能なすべてのネットワーク・インターフェースを指定するには、アスタリスクを二重引用符で囲んで使用します。ローカル接続のみを許可するには、値 localhost を使用します。

- IBM MQ 9.0.3 以前では、mqwebuser.xml ファイルを編集します。

1. mqwebuser.xml ファイルを開きます。

mqwebuser.xml ファイルは、次のいずれかのディレクトリーにあります。

- **ULW** で UNIX, Linux, and Windows: MQ\_DATA\_DIRECTORY/web/installations/  
installationName/servers/mqweb

- **z/OS** z/OS の場合: WLP\_user\_directory/servers/mqweb

ここで WLP\_user\_directory は、mqweb サーバー定義を作成するために **crtmqweb.sh** スクリプトを実行した際に指定されたディレクトリーです。

2. mqweb サーバーを構成します。

- mqweb サーバーへのリモート接続を許可するには、以下の行を mqwebuser.xml ファイルの <server> タグ内に追加します。

```
<variable name="httpHost" value="hostName"/>
```

hostName は IP アドレス、ドメイン名サフィックス付きのドメイン・ネーム・サーバー (DNS) ホスト名、または IBM MQ がインストールされているサーバーの DNS ホスト名を示します。使用可能なすべてのネットワーク・インターフェースを指定するには、アスタリスク (\*) を使用します。

- mqweb サーバーへのローカル接続のみを許可する場合は、以下の行を mqwebuser.xml ファイルから削除するか、値を localhost に設定します。

```
<variable name="httpHost" value="hostName"/>
```

## V 9.0.1 HTTP および HTTPS ポートの構成

デフォルトでは、IBM MQ Console と REST API をホストする mqweb サーバーは HTTPS ポート 9443 を使用します。HTTP 接続に関連付けられたポートは無効になります。HTTP ポートを有効にしたり、別の HTTPS ポートを構成したり、HTTP や HTTPS ポートを無効にしたりできます。 **V 9.0.4** IBM MQ 9.0.4 以降では、**setmqweb** コマンドを使用して、ポートを構成できます。IBM MQ 9.0.3 以前では、mqwebuser.xml ファイルを編集してポートを構成できます。

### 始める前に

この手順を実行するには、特権ユーザーである必要があります。

HTTP ポートと HTTPS ポートの両方を有効にする場合、HTTPS 要求に対して発行される LTPA トークンを、ブラウザから HTTP 要求で再利用できます。この動作が行われないうまく mqweb サーバーを構成することができ、次の行を mqwebuser.xml ファイルに追加することにより、環境のセキュリティーを向上させることができます。

```
<webAppSecurity ssoRequiresSSL="true"/>
```

**V 9.0.4** IBM MQ 9.0.4 以降では、以下のコマンドを使用して、HTTP や HTTPS ポートの現在の構成を表示できます。

```
dspmweb properties -a
```

httpPort フィールドは HTTP ポートを示し、httpsPort フィールドは HTTPS ポートを示します。詳しくは、[dspmweb](#) を参照してください。



**重要:** **z/OS** **V 9.0.4**

z/OS で **setmqweb** コマンドまたは **dspmweb** コマンドを発行するには、その前に WLP\_USER\_DIR 環境変数を設定し、この変数が mqweb サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

WLP\_user\_directory は、crtmqweb.sh に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

### 手順

#### **V 9.0.4**

以下のいずれかの方法を使用して、ポートを構成します。

- IBM MQ 9.0.4 以降では、**setmqweb properties** コマンドを使用します。

- HTTP ポートを有効にしたり構成したりするには、以下のコマンドを使用します。

```
setmqweb properties -k httpPort -v portNumber
```

ここで、portNumber は HTTP 接続に使用するポートを示します。-1 の値を使用すると、ポートを無効にすることができます。

- HTTPS ポートを構成するには、以下のコマンドを使用します。

```
setmqweb properties -k httpsPort -v portNumber
```

ここで、*portNumber* は HTTPS 接続に使用するポートを示します。-1 の値を使用すると、ポートを無効にすることができます。

- IBM MQ 9.0.3 以前では、mqwebuser.xml ファイルを編集します。

1. mqwebuser.xml ファイルを開きます。

mqwebuser.xml ファイルは、次のいずれかのディレクトリーにあります。

-  **ULW** で UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`

-  **z/OS** の場合: `WLP_user_directory/servers/mqweb`

ここで *WLP\_user\_directory* は、mqweb サーバー定義を作成するために `crtmqweb.sh` スクリプトを実行した際に指定されたディレクトリーです。

2. ポートを構成します。

- HTTP ポートを有効にしたり構成したりするには、mqwebuser.xml ファイルの `<server>` タグ内の次の行を編集または追加します。

```
<variable name="httpPort" value="portNumber" />
```

ここで、*portNumber* は HTTP 接続に使用するポートを示します。-1 の値を使用すると、ポートを無効にすることができます。

- HTTPS ポートを構成するには、mqwebuser.xml ファイルの `<server>` タグ内の次の行を編集または追加します。

```
<variable name="httpsPort" value="portNumber" />
```

ここで、*portNumber* は HTTPS 接続に使用するポートを示します。-1 の値を使用すると、ポートを無効にすることができます。

## 応答タイムアウトの構成

デフォルトでは、IBM MQ Console および REST API は、30 秒以内にクライアントに応答を送り返さないとタイムアウトになります。  IBM MQ 9.0.4 以降では、`setmqweb` コマンドを使用して、別のタイムアウト値を使用するように IBM MQ Console と REST API を構成できます。IBM MQ 9.0.3 以前では、mqwebuser.xml ファイルを編集することにより、別のタイムアウト値を使用するように IBM MQ Console および REST API を構成できます。

### 始める前に

この手順を実行するには、特権ユーザーである必要があります。

 IBM MQ 9.0.4 以降では、以下のコマンドを使用して、REST API の応答タイムアウトの現在の構成を表示できます。

```
dspmweb properties -a
```

mqRestRequestTimeout フィールドは、応答タイムアウトの現行値を示します。詳しくは、[dspmweb](#) を参照してください。



**重要:**  

z/OS で `setmqweb` コマンドまたは `dspmweb` コマンドを発行するには、その前に `WLP_USER_DIR` 環境変数を設定し、この変数が mqweb サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

`WLP_user_directory` は、`crtmqweb.sh` に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

## 手順

### • V 9.0.4

以下のいずれかの方法を使用して、タイムアウトを構成します。

- IBM MQ 9.0.4 以降では、**setmqweb properties** コマンドを使用します。

```
setmqweb properties -k mqRestRequestTimeout -v timeout
```

*timeout* はタイムアウトになるまでの時間 (秒単位) です。

- IBM MQ 9.0.3 以前では、`mqwebuser.xml` ファイルを編集します。

1. `mqwebuser.xml` ファイルを開きます。

`mqwebuser.xml` ファイルは、次のいずれかのディレクトリーにあります。

- **ULW** で UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`

- **z/OS** z/OS の場合: `WLP_user_directory/servers/mqweb`

ここで `WLP_user_directory` は、`mqweb` サーバー定義を作成するために **crtmqweb.sh** スクリプトを実行した際に指定されたディレクトリーです。

2. `mqwebuser.xml` ファイルの `<server>` タグ内の次の行を編集または追加して、タイムアウトを構成します。

```
<variable name="mqRestRequestTimeout" value="timeout" />
```

*timeout* はタイムアウトになるまでの時間 (秒単位) です。

## V 9.0.1 自動始動の構成

デフォルトでは、`mqweb` サーバーを始動すると、IBM MQ Console が自動的に始動します。IBM MQ 9.0.1 では、REST API は自動的に始動しません。IBM MQ 9.0.2 以降、`mqweb` サーバーの始動時に REST API が自動的に開始されるようになりました。 **V 9.0.4** IBM MQ 9.0.4 以降では、**setmqweb** コマンドを使用して、IBM MQ Console と REST API を自動的に開始するかどうかを構成できます。IBM MQ 9.0.3 以前では、`mqwebuser.xml` ファイルを編集して、IBM MQ Console と REST API を自動的に開始するかどうかを構成できます。

## 始める前に

この手順を実行するには、[特権ユーザー](#)である必要があります。

**V 9.0.4** IBM MQ 9.0.4 以降では、以下のコマンドを使用して、REST API の自動開始の現在の構成を表示できます。

```
dspmweb properties -a
```

`mqRestAutostart` フィールドは REST API が自動的に開始されるかどうかを示し、`mqConsoleAutostart` フィールドは IBM MQ Console が自動的に開始されるかどうかを示します。詳しくは、[dspmweb](#) を参照してください。



**重要:** **z/OS** **V 9.0.4**

z/OS で **setmqweb** コマンドまたは **dspmweb** コマンドを発行するには、その前に `WLP_USER_DIR` 環境変数を設定し、この変数が `mqweb` サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

`WLP_user_directory` は、`crtmqweb.sh` に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

## 手順

### V 9.0.4

以下のいずれかの方法を使用して、IBM MQ Console および REST API を自動的に開始するかどうかを構成します。

- IBM MQ 9.0.4 以降では、**setmqweb properties** コマンドを使用します。

- 以下のコマンドを使用して、IBM MQ Console を自動的に開始するかどうかを構成します。

```
setmqweb properties -k mqconsoleAutostart -v start
```

ここで、`start` は、IBM MQ Console を自動的に開始する場合は値 `True`、それ以外の場合は `False` です。

- 以下のコマンドを使用して、REST API に手動開始を必要とするかどうかを構成します。

```
setmqweb properties -k mqRestAutostart -v start
```

ここで、`start` は、REST API を自動的に開始する場合は値 `True`、それ以外の場合は `False` です。

- IBM MQ 9.0.3 以前では、`mqwebuser.xml` ファイルを編集します。

1. `mqwebuser.xml` ファイルを開きます。

`mqwebuser.xml` ファイルは、次のいずれかのディレクトリーにあります。

- **ULW** で UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`

- **z/OS** z/OS の場合: `WLP_user_directory/servers/mqweb`

ここで `WLP_user_directory` は、`mqweb` サーバー定義を作成するために **crtmqweb.sh** スクリプトを実行した際に指定されたディレクトリーです。

2. 以下のようにして自動開始を構成します。

- `mqwebuser.xml` ファイルの `<server>` タグ内の以下の行を追加または更新して、IBM MQ Console を手動開始する必要があるかどうかを構成します。

```
<variable name="mqConsoleAutostart" value="start"/>
```

ここで、`start` は、IBM MQ Console を自動的に開始する場合は値 `True`、それ以外の場合は `False` です。

- `mqwebuser.xml` ファイルの `<server>` タグ内の以下の行を追加または更新して、REST API を手動開始する必要があるかどうかを構成します。

```
<variable name="mqRestAutostart" value="start"/>
```

ここで、`start` は、REST API を自動的に開始する場合は値 `True`、それ以外の場合は `False` です。

### V 9.0.1 ロギングの構成

IBM MQ Console と REST API をホストする `mqweb` サーバーで使用するロギング・レベル、最大ログ・ファイル・サイズ、およびログ・ファイルの最大数を構成できます。 **V 9.0.4** IBM MQ 9.0.4 以降では、

**setmqweb** コマンドを使用して、ロギングを構成できます。IBM MQ 9.0.3 以前では、mqwebuser.xml ファイルを編集してロギングを構成できます。

## 始める前に

この手順を実行するには、[特権ユーザー](#)である必要があります。

**V 9.0.4** IBM MQ 9.0.4 以降では、以下のコマンドを使用して、REST API のロギングの現在の構成を表示できます。

```
dspmqweb properties -a
```

maxTraceFileSize フィールドは最大トレース・ファイル・サイズを示し、maxTraceFiles フィールドはトレース・ファイルの最大数を示し、traceSpec フィールドは使用されるトレースのレベルを示します。詳しくは、[dspmqweb](#) を参照してください。



**重要:** **z/OS** **V 9.0.4**

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行するには、その前に WLP\_USER\_DIR 環境変数を設定し、この変数が mqweb サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

WLP\_user\_directory は、crtmqweb.sh に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

## このタスクについて

mqweb サーバーのログ・ファイルは、次のいずれかのディレクトリーにあります。

- **ULW** で UNIX, Linux, and Windows: MQ\_DATA\_DIRECTORY/web/installations/*installationName*/servers/mqweb/logs
- **z/OS** z/OS の場合: WLP\_user\_directory/servers/mqweb/logs

ここで WLP\_user\_directory は、mqweb サーバー定義を作成するために **crtmqweb.sh** スクリプトを実行した際に指定されたディレクトリーです。

IBM MQ Console および REST API のトレースを有効化する方法について詳しくは、[IBM MQ Console](#) および [REST API のトレース](#)を参照してください。

## 手順

### • **V 9.0.4**

以下のいずれかの方法を使用して、ロギングを構成します。

- IBM MQ 9.0.4 以降では、**setmqweb properties** コマンドを使用します。
  - 最大ログ・ファイル・サイズを設定するには、以下のコマンドを使用します。

```
setmqweb properties -k maxTraceFileSize -v size
```

ここで、size は各ログ・ファイルの上限サイズを MB で示します。デフォルト値は 20 です。
  - ロギングに使用するファイルの最大数を設定するには、以下のコマンドを使用します。

```
setmqweb properties -k maxTraceFiles -v max
```

ここで、max はファイルの最大数を示します。デフォルト値は 2 です。
  - 使用するロギングのレベルを構成するには、以下のコマンドを使用します。

setmqweb properties -k traceSpec -v level

ここで、level は、614 ページの表 37 にリストされているいずれかの値です。この表は、詳細度が高くなる順に各ロギング・レベルの概要を示しています。あるロギング・レベルを有効にすると、それより前のすべてのレベルも有効になります。例えば、**\*=warning** ロギング・レベルを有効にすると、**\*=severe** および **\*=fatal** ロギング・レベルも有効になります。

デフォルト値は **\*=info** です。この値は、IBM サービスから要請された場合に変更してください。

値	適用されるロギング・レベル
*=off	ロギングはオフ。
*=fatal	タスクが続行できず、コンポーネント、アプリケーション、およびサーバーが機能できない。
*=severe	タスクが続行できない。ただし、コンポーネント、アプリケーション、およびサーバーはまだ機能できる。このレベルは、リカバリー不能エラーが差し迫っていることを示すこともあります。
* = 警告	潜在的なエラー、または差し迫ったエラー。このレベルは、障害の進行 (例えば、潜在的なリソース・リーク) を示すこともあります。
*=audit	サーバー状態またはリソースに影響を与える重大なイベント。
*=info	タスクの全体的な進行状況を大まかに示す一般情報
*=config	構成の変更または状況
*=detail	サブタスクの進行の詳細を示す一般情報
*=fine	トレース情報 - 一般トレース + メソッドの開始、終了、戻り値
*=finer	トレース情報 - 詳細トレース
*=finest	トレース情報 - より詳細なトレース。問題をデバッグするために必要な詳細がすべて含まれる
* = すべて	すべてのイベントをログに記録する

- IBM MQ 9.0.3 以前では、mqwebuser.xml ファイルを編集します。

1. mqwebuser.xml ファイルを開きます。

mqwebuser.xml ファイルは、次のいずれかのディレクトリーにあります。

-  UNIX, Linux, and Windows: MQ\_DATA\_DIRECTORY/web/installations/installationName/servers/mqweb
-  z/OS の場合: WLP\_user\_directory/servers/mqweb

ここで WLP\_user\_directory は、mqweb サーバー定義を作成するために **crtmqweb.sh** スクリプトを実行した際に指定されたディレクトリーです。

2. 以下のようにしてロギングを構成します。

- 最大ログ・ファイル・サイズを設定するには、mqwebuser.xml ファイルの <server> タグ内の次の行を編集または追加します。

```
<variable name="maxTraceFileSize" value="size" />
```

ここで、`size` は各ログ・ファイルの上限サイズを MB で示します。デフォルト値は 20 です。

- ログイングに使用するファイルの最大数を設定するには、`mqwebuser.xml` ファイルの `<server>` タグ内の次の行を編集または追加します。

```
<variable name="maxTraceFiles" value="max" />
```

ここで、`max` はファイルの最大数を示します。デフォルト値は 2 です。

- 使用するログイングのレベルを構成するには、`mqwebuser.xml` ファイルの `<server>` タグ内の次の行を編集または追加します。

```
<variable name="traceSpec" value="level" />
```

ここで、`level` は、[614 ページの表 37](#) の表にリストされているいずれかの値です。

この表は、詳細度が高くなる順に各ログイング・レベルの概要を示しています。あるログイング・レベルを有効にすると、それより前のすべてのレベルも有効になります。例えば、**\*=warning** ログイング・レベルを有効にすると、**\*=severe** および **\*=fatal** ログイング・レベルも有効になります。

デフォルト値は **\*=info** です。この値は、IBM サービスから要請された場合に変更してください。

## LTPA トークンの有効期間の構成

LTPA トークンを使用すると、ユーザーは WebSphere Application Server Liberty への要求のたびにユーザー名とパスワードの資格情報を入力する必要がなくなります。LTPA 認証トークンの有効期間間隔を構成できます。

### 始める前に

この手順を実行するには、[特権ユーザー](#)である必要があります。

**V 9.0.4** IBM MQ 9.0.4 以降、`-a` フラグを指定した `dspmqweb properties` コマンドを使用して、トークンの有効期限の現在の構成を表示できます。詳しくは、[dspmqweb](#) を参照してください。トークンの有効期限の値は、`-k` フラグと `-d` フラグを指定した `setmqweb properties` コマンドを使用してリセットできます。詳細については、[setmqweb](#) を参照してください。

**V 9.0.2**

注：IBM MQ Console と REST API を使用したトークン認証の両方を使用している場合、有効期間間隔が共有されます。



**重要：** **z/OS** **V 9.0.4**

z/OS で `setmqweb` コマンドまたは `dspmqweb` コマンドを発行するには、その前に `WLP_USER_DIR` 環境変数を設定し、この変数が `mqweb` サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

`WLP_user_directory` は、`crtmqweb.sh` に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

### このタスクについて

ユーザーが IBM MQ Console にログインすると、LTPA トークンが生成されます。REST API でトークンベースの認証を使用する場合、HTTP POST メソッドで `/login` REST API リソースを使用してユーザーが

ログインする際に LTPA トークンが生成されます。ユーザーの認証にトークンが使用されます。そのトークンの有効期限が切れるまでは、ユーザーはユーザー ID とパスワードを使用して再びログインする必要はありません。デフォルトの有効期間は 120 分です。 **V 9.0.4** IBM MQ 9.0.4 以降では、**setmqweb** コマンドを使用して、トークンの有効期限を構成できます。IBM MQ 9.0.3 以前では、**mqwebuser.xml** ファイルを編集することによって、トークンの有効期限が切れるタイミングを構成できます。

## 手順

### **V 9.0.4**

以下のいずれかの方法を使用して、トークンの有効期限を構成します。

- IBM MQ 9.0.4 以降では、**setmqweb properties** コマンドを使用します。

```
setmqweb properties -k ltpaExpiration -v time
```

ここで、*time* は、LTPA トークンの有効期限が切れてユーザーがログアウトされるまでの時間を分数で指定します。デフォルト値は 120 分です。

- IBM MQ 9.0.3 以前では、**mqwebuser.xml** ファイルを編集します。

1. **mqwebuser.xml** ファイルを開きます。

**mqwebuser.xml** ファイルは、次のいずれかのディレクトリーにあります。

- **ULW** で UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`

- **z/OS** の場合: `WLP_user_directory/servers/mqweb`

ここで *WLP\_user\_directory* は、mqweb サーバー定義を作成するために **crtmqweb.sh** スクリプトを実行した際に指定されたディレクトリーです。

2. LTPA トークンの有効期間を構成するには、**mqwebuser.xml** ファイルの `<server>` タグ内の次の行を編集または追加します。

```
<variable name="ltpaExpiration" value="time" />
```

ここで、*time* は、LTPA トークンの有効期限が切れてユーザーがログアウトされるまでの時間を分数で指定します。デフォルト値は 120 分です。

## **V 9.0.4** administrative REST API ゲートウェイの構成

デフォルトでは、administrative REST API ゲートウェイは有効になっています。administrative REST API ゲートウェイを有効にすると、ゲートウェイ・キュー・マネージャーを使用して REST API によるリモート管理を実行できます。デフォルトのゲートウェイ・キュー・マネージャーとして使用されるキュー・マネージャーを構成したり、**setmqweb properties** コマンドを使用して administrative REST API ゲートウェイを無効にすることによって、リモート管理が行われないようにしたりできます。

### このタスクについて

この手順を実行するには、特権ユーザーである必要があります。

administrative REST API ゲートウェイの現在の構成を表示するには、以下のコマンドを使用します。

```
dspmqweb properties -a
```

**mqRestGatewayEnabled** フィールドにゲートウェイが有効かどうかが表示され、**mqRestGatewayQmgr** フィールドにデフォルトのゲートウェイ・キュー・マネージャーの名前が表示されます。詳しくは、**dspmqweb** を参照してください。

デフォルトのゲートウェイ・キュー・マネージャーは、以下の両方の記述が当てはまる場合に使用されません。

- REST 要求の `ibm-mq-rest-gateway-qmgr` ヘッダーでキュー・マネージャーが指定されていない。

- REST API リソース URL で指定されたキュー・マネージャーがローカル・キュー・マネージャーではない。

REST API を使用したリモート管理の詳細については、[REST API によるリモート管理](#)を参照してください。



**重要:** z/OS V 9.0.4

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行するには、その前に WLP\_USER\_DIR 環境変数を設定し、この変数が mqweb サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

WLP\_user\_directory は、crtmqweb.sh に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

## 手順

- 以下のコマンドを使用して、administrative REST API ゲートウェイを有効にするかどうかを構成します。  
`setmqweb properties -k mqRestGatewayEnabled -v enabled`  
ここで、*enabled* は administrative REST API ゲートウェイを有効にする場合は値 **true**、それ以外の場合は **false** です。
- どのキュー・マネージャーがデフォルトのゲートウェイ・キュー・マネージャーとして使用されるかを構成します。
  - 以下のコマンドを使用して、デフォルトのゲートウェイ・キュー・マネージャーを設定します。  
`setmqweb properties -k mqRestGatewayQmgr -v qmgrName`  
ここで、*qmgrName* は mqweb サーバーと同じインストールのキュー・マネージャーの名前です。
  - 以下のコマンドを使用して、デフォルトのゲートウェイ・キュー・マネージャーを設定解除します。  
`setmqweb properties -k mqRestGatewayQmgr -d`

## V 9.0.4 messaging REST API の構成

IBM MQ Console と REST API をホストする mqweb サーバーでは、デフォルトで messaging REST API が有効になります。 **setmqweb properties** コマンドを使用して、メッセージングを有効にするか無効にするかを構成できます。

### 始める前に

この手順を実行するには、[特権ユーザー](#)である必要があります。

以下のコマンドを使用して、messaging REST API の現在の構成を表示できます。

```
dspmqweb properties -a
```

mqRestMessagingEnabled フィールドは、messaging REST API が有効か無効かを示します。詳しくは、[dspmqweb](#) を参照してください。

messaging REST API を使用するには、呼び出し元が mqweb サーバーで認証を受ける必要があります。また、呼び出し元は MQWebUser 役割のメンバーである必要があります。MQWebAdmin 役割および MQWebAdminRO 役割は、messaging REST API には適用されません。呼び出し元は、OAM/RACF でも許可される必要があります。REST API のセキュリティーについては詳しくは、[IBM MQ コンソールおよび REST API のセキュリティー](#)を参照してください。



重要: z/OS V 9.0.4

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行するには、その前に WLP\_USER\_DIR 環境変数を設定し、この変数が mqweb サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

WLP\_user\_directory は、crtmqweb.sh に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

## 手順

### V 9.0.4

以下の方法を使用して、messaging REST API を構成します。

- **setmqweb properties** コマンドを使用します。

- 以下のコマンドを使用して、messaging REST API を有効にするかどうかを構成します。

```
setmqweb properties -k mqRestMessagingEnabled -v enabled
```

ここで、*enabled* は、messaging REST API を有効にする場合は値 *true*、それ以外の場合は *false* です。

## V 9.0.5 REST API for MFT の構成

IBM MQ Console と REST API をホストする mqweb サーバーでは、デフォルトで MFT REST API が有効になります。 **setmqweb properties** コマンドを使用して、MFT の REST API を有効または無効にし、調整キュー・マネージャーを設定し、MFT 再接続タイムアウトを指定することができます。

### 始める前に

この手順を実行するには、[特権ユーザー](#)である必要があります。

以下のコマンドを使用して、REST API for MFT の現在の構成を表示できます。

```
dspmqweb properties -a
```

mqRestMftEnabled フィールドは、REST API for MFT が有効または無効であるかどうかを示します。mqRestMftCoordinationQmgr フィールドには調整キュー・マネージャーの名前が表示され、mqRestMftReconnectTimeoutInMinutes フィールドには MFT 要求のタイムアウト値が表示されます。詳しくは、[dspmqweb](#) を参照してください。

REST API for MFT を使用するには、呼び出し元が mqweb サーバーで認証を受ける必要があります。さらに呼び出し元は、MFTWebAdmin と MFTWebAdminRO のうち 1 つ以上の役割を持つメンバーでなければなりません。



重要: z/OS V 9.0.4

z/OS で **setmqweb** コマンドまたは **dspmqweb** コマンドを発行するには、その前に WLP\_USER\_DIR 環境変数を設定し、この変数が mqweb サーバー構成を指すようにしておく必要があります。

そのためには、以下のコマンドを実行します。

```
export WLP_USER_DIR=WLP_user_directory
```

`WLP_user_directory` は、`crtmqweb.sh` に渡すディレクトリー名です。以下に例を示します。

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

詳しくは、[Liberty サーバー定義の作成](#)を参照してください。

## このタスクについて

REST API for MFT の構成時に、次の 3 つのプロパティを構成できます。

- REST API for MFT が有効であるかどうか。デフォルトでは無効になっています。
- MFT REST API リソース使用時の情報取得元となる調整キュー・マネージャーの名前。このキュー・マネージャーは、mqweb サーバーと同じマシン上のキュー・マネージャーである必要があります。REST API for MFT は、mqweb サーバーの始動時に、このキュー・マネージャーへのバインディング接続を確立します。

デフォルトでは、このキュー・マネージャー名は空白です。値を設定せず、MFT REST API が呼び出されないと、HTTP 400 が返されます。

- REST API for MFT が調整キュー・マネージャーへの接続の試行を停止するまでのタイムアウト (分単位)。接続を再確立するための最初の試行は、調整キュー・マネージャーへの接続が切れた直後に行われます。それが失敗すると、再接続の試行が 5 分間隔で行われます。

再接続がタイムアウトになった後の最初の再接続の試行は、`/transfer` または `/agent` の REST API リソースが呼び出される時次に行われます。その再接続の試行が失敗すると、再接続がタイムアウトになるまで、MFT が再び 5 分ごとに再接続を試行するようになります。

デフォルトで、タイムアウト値は 30 分です。調整キュー・マネージャーが始動していない状態で MFT REST API が呼び出されると、HTTP 503 が返されます。

## 手順

1. REST API for MFT の構成を調整します。

- 以下のコマンドを使用して、REST API for MFT を有効にするかどうかを構成します。

```
setmqweb properties -k mqRestMftEnabled -v value
```

ここで、*value* は、MFT の REST API を有効にする場合は `true`、それ以外の場合は `false` です。

- 以下のコマンドを使用して、転送の詳細の取得元となる調整キュー・マネージャーを構成します。

```
setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
```

ここで、*qmgrName* は、調整キュー・マネージャーの名前です。調整キュー・マネージャーは、mqweb サーバーが稼働しているマシンに配置する必要があります。

- 以下のコマンドを使用して、REST API for MFT が調整キュー・マネージャーへの接続の試行を停止するまでのタイムアウト (分単位) を構成します。

```
setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -v time
```

ここで、*time* はタイムアウトが発生するまでの時間 (分単位) を指定します。

- 0-5 の間の値は、REST API for MFT が調整キュー・マネージャーへの再接続を 1 回だけ試行することを指定します。接続に失敗すると、REST API が呼び出されるまで接続の再確立は試行されません。
- この値を -1 に指定すると、REST API for MFT は、接続が成功するまで再接続を試みます。

2. 以下のコマンドを入力して、mqweb サーバーを再始動します。

```
endmqweb  
startmqweb
```

## V 9.0.2 mqweb サーバーの JVM のチューニング

デフォルトで、mqweb サーバーの Java 仮想マシン (JVM) では、ヒープの最小サイズと最大サイズにプラットフォーム固有のデフォルトを使用します。デフォルト値の変更が必要になることもあります。例えば、mqweb サーバーによって `java.lang.OutOfMemoryError` がスローされる場合は、ヒープの最大サイズを増やす必要があります。デフォルト値は、`jvm.options` ファイルで変更できます。

### 手順

1. `jvm.options` ファイルを開きます。

`jvm.options` ファイルは、次のいずれかのディレクトリーにあります。

- **ULW** で UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb`
- **z/OS** z/OS の場合: `WLP_user_directory/servers/mqweb`

ここで `WLP_user_directory` は、mqweb サーバー定義を作成するために `crtmqweb.sh` スクリプトを実行した際に指定されたディレクトリーです。

2. オプション: 次の行をファイルに追加して、最大ヒープ・サイズを設定します。

```
-XmxMaxSize
```

ここで、`MaxSize` は、ヒープの最大サイズを指定します (MB 単位)。  
例えば、以下の行は、最大ヒープ・サイズを 1GB に設定します。

```
-Xmx1024m
```

3. オプション: 次の行をファイルに追加して、最小ヒープ・サイズを設定します。

```
-XmsMinSize
```

ここで、`MinSize` は、ヒープの最小サイズを指定します (MB 単位)。  
例えば、次の行は、最小ヒープ・サイズを 512MB に設定します。

```
-Xms512m
```

4. コマンド行で以下のコマンドを入力して、mqweb サーバーを再始動します。

```
endmqweb  
strmqweb
```

## IBM MQ Console および REST API のインストール・コンポーネントのファイル構造

IBM MQ Console および REST API のインストール・コンポーネントには、2 組のディレクトリー構造が関連付けられています。一方のディレクトリー構造には、編集可能なファイルが含まれています。もう一方のディレクトリー構造には、編集不可のファイルが含まれています。

### 編集可能なファイル

ユーザーが編集可能なファイルは、IBM MQ Console および REST API のインストール・コンポーネントの初期インストールの一部として配置されます。これらは編集可能なファイルであるため、保守を適用するときに変更されることはありません。

ユーザーが編集可能なファイルの場所は、以下のようにオペレーティング・システムによって異なります。

- **ULW** で UNIX, Linux, and Windows: `MQ_DATA_DIRECTORY/web/installations/installationName/`

- **z/OS** z/OS の場合: `WLP_user_directory`

ここで `WLP_user_directory` は、mqweb サーバー定義を作成するために `crtmqweb.sh` スクリプトを実行した際に指定されたディレクトリーです。

この最上位ディレクトリーの下に、以下のディレクトリーとファイルがあります。

ディレクトリーおよびファイル	説明
<code>angular.persistence/</code>	IBM MQ Console ダッシュボード構成が保管されるディレクトリー。
<code>servers/</code>	WebSphere Liberty Profile サーバーのディレクトリー。
<code>servers/mqweb</code>	mqweb サーバーのディレクトリー構造が含まれるディレクトリー。
<code>servers/mqweb/logs</code>	mqweb サーバーのログが含まれるディレクトリー。
<code>servers/mqweb/logs/console.log</code>	基本的なサーバー状況および操作メッセージのログ。
<code>servers/mqweb/logs/ffdc</code>	初期障害データ・キャプチャー機能 (FFDC) の出力ディレクトリー。
<code>servers/mqweb/logs/messages.log</code>	IBM MQ Console および REST API を含む、mqweb サーバーのランタイム・メッセージのログ。古いメッセージは、 <code>messages_timestamp.log</code> という名前のファイルに保管されます。
<code>servers/mqweb/logs/trace.log</code>	IBM MQ Console および REST API を含む、mqweb サーバーのトレースのログ。古いトレースは、 <code>trace_timestamp.log</code> という名前のファイルに保管されます。これらのファイルは、トレースが有効になっている場合にのみ存在します。
<code>servers/mqweb/logs/state</code>	サーバー固有の状態。
<code>servers/mqweb/server.xml</code>	メインのサーバー構成ファイル。 このファイルは読み取り専用です。 <code>mqwebuser.xml</code> ファイルを編集して、デフォルトの構成をオーバーライドします。
<code>servers/mqweb/mqwebuser.xml</code>	IBM MQ Console および REST API の構成ファイル。このファイルで構成した設定は、デフォルトの構成をオーバーライドします。  このファイルを編集するには、 <u>特権ユーザー</u> である必要があります。
<code>servers/mqweb/resources</code>	鍵ストアなどのさまざまなサーバー・リソースを含むディレクトリー。
<code>servers/mqweb/workarea</code>	サーバーの稼働中に作成されるディレクトリー。 このディレクトリーは、サーバーが最初に実行された後に作成されます。

## 編集不可のファイル

編集不可のファイルは、IBM MQ Console および REST API のインストール・コンポーネントの初期インストールの一部として配置されます。これらのファイルは、保守を適用すると更新されます。

ユーザーが編集可能なファイルの場所は、以下のようにオペレーティング・システムによって異なります。

- ▶ **ULW** で UNIX, Linux, and Windows: `MQ_INSTALLATION_PATH/web`
- ▶ **z/OS** z/OS の場合: `installation_directory/web/`

`installation_directory` は、IBM MQ UNIX System Services Components のインストール・パスです。

この場所には、以下のディレクトリー構造とファイルがあります。

ディレクトリーおよびファイル	説明
<code>bin/</code>	Liberty コマンドが含まれているディレクトリー。 このディレクトリーのスクリプトを実行するには、 <u>特権ユーザー</u> である必要があります。
<code>mq/</code>	さまざまな IBM MQ リソースが含まれているディレクトリー構造。
<code>mq/apps/</code>	IBM MQ Console および REST API アプリケーションが含まれているディレクトリー。
<code>mq/etc/</code>	
<code>mq/etc/mqweb.xml</code>	mqweb サーバーの読み取り専用の構成ファイル。 mqwebuser.xml ファイルを編集して、構成変更を行います。
<code>mq/libs</code>	IBM MQ Console および REST API で使用される共用ライブラリーが含まれているディレクトリー。
<code>mq/samp</code>	サンプルが含まれているディレクトリー。
<code>mq/samp/configuration</code>	mqwebuser.xml ファイルにコピーできるサンプル構成ファイルが含まれているディレクトリー。

## IBM MQ を Docker で構成

ここでは、Docker を使用して IBM MQ を構成する方法について説明します。

### このタスクについて

Docker を使用すると、IBM MQ キュー・マネージャーや IBM MQ クライアント・アプリケーションをそのすべての依存関係と共に、ソフトウェア開発用に標準化された単位にパッケージ化することができます。

アプリケーションに対する変更を、素早く簡単にテスト・システムおよびステージング・システムにデプロイできます。この機能は、企業の継続的デリバリーに大きなメリットをもたらす可能性があります。

### 手順

- Docker を使用して IBM MQ を構成する方法については、以下のサブトピックを参照してください。
  - ▶ **Linux** 623 ページの『Linux システムでの Docker サポート』
  - 623 ページの『Docker を使用した独自の IBM MQ キュー・マネージャー・イメージの計画』
  - 624 ページの『Docker を使用したサンプル IBM MQ キュー・マネージャー・イメージの作成』
  - 627 ページの『別々のコンテナでのローカル・バインディング・アプリケーションの実行』

## Linux Linux システムでの Docker サポート

Linux システムで Docker を使用する場合は、以下の情報を考慮する必要があります。

- Docker イメージで使用するベース・イメージでは、サポート対象 Linux オペレーティング・システムを使用する必要があります。
- IBM MQ インストーラーを使用して、Docker イメージの内部に製品をインストールする必要があります。
- サポートされるパッケージのリストについては、[Linux システム用の IBM MQ rpm コンポーネント](#)を参照してください。
- **V 9.0.4** 以下のパッケージはサポートされません。

- MQSeriesBCBridge

- MQSeriesRDQM

- キュー・マネージャーのデータ・ディレクトリー (デフォルトでは /var/mqm) は、永続的な状態を保持する Docker ボリューム上に保管する必要があります。

**重要:** union ファイル・システムは使用できません。

ホスト・ディレクトリーをデータ・ボリュームとしてマウントするか、またはデータ・ボリューム・コンテナを使用する必要があります。詳しくは、[Manage data in containers](#) を参照してください。

- コンテナ内で、**endmqm** などの IBM MQ 制御コマンドを実行できる必要があります。
- 診断で使用するために、コンテナ内からファイルやディレクトリーを取得できる必要があります。
- **V 9.0.3** 別のコンテナで実行されているキュー・マネージャーにアプリケーションをローカル・バインドするために、名前空間操作を使用して、キュー・マネージャーのコンテナの名前空間を他のコンテナと共有することができます。詳細については、627 ページの『[別々のコンテナでのローカル・バインディング・アプリケーションの実行](#)』を参照してください。

## Docker を使用した独自の IBM MQ キュー・マネージャー・イメージの計画

ここでは、Docker を使用して IBM MQ を構成する方法について説明します。Docker で IBM MQ キュー・マネージャーを実行する際に考慮すべき要件がいくつかあります。サンプルの Docker イメージはそれらの要件に対処できるように作成されていますが、独自のイメージを使用する場合には、それらの要件に対処する方法を検討する必要があります。

### プロセス監視

Docker コンテナを実行すると、実際には単一のプロセス (コンテナ内の PID 1) が実行されます。このプロセスは、後で複数の子プロセスを作成する可能性があります。

メインのプロセスが終了すると、Docker はコンテナを停止します。IBM MQ キュー・マネージャーでは、複数のプロセスをバックグラウンドで実行する必要があります。

このため、キュー・マネージャーの実行中は、メインプロセスがアクティブな状態であることを確認する必要があります。グッド・プラクティスとして、例えば管理照会などを実行して、キュー・マネージャーがアクティブであることをこのプロセスから確認してください。

### /var/mqm へのデータの取り込み

Docker コンテナには、/var/mqm を Docker ボリュームとして構成する必要があります。

これを行うと、コンテナが最初に始動したときに、このボリュームのディレクトリーは空の状態です。通常、このディレクトリーにはインストール時にデータが追加されますが、Docker を使用する場合、インストール環境とランタイム環境は別のものです。

**V 9.0.3** これを解決するには、コンテナの開始時に、**crtmqdir** コマンドを使用して、初めて実行するときに /var/mqm にデータを取り込むことができます。

## Docker を使用したサンプル IBM MQ キュー・マネージャー・イメージの作成

この情報を使用して、Docker コンテナで IBM MQ キュー・マネージャーを実行するためのサンプルの Docker イメージを作成します。

### このタスクについて

最初に、Ubuntu Linux ファイル・システムと IBM MQ のクリーン・インストール環境を含むベース・イメージを作成します。

次に、このベースの上に、追加の Docker イメージ・レイヤーを作成します。このイメージ・レイヤーによって、ユーザー ID とパスワードによる基本的なセキュリティを可能にする IBM MQ 構成を追加します。

最後に、このイメージを Docker のファイル・システムとして使用して、そのコンテナを実行します。/var/mqm のコンテンツは、Docker のホスト・ファイル・システム上のコンテナ固有の Docker ボリュームから提供されます。

### 手順

- Docker コンテナで IBM MQ キュー・マネージャーを実行するためのサンプルの Docker イメージを作成する方法については、以下のサブピックを参照してください。
  - [624 ページの『サンプルの IBM MQ キュー・マネージャーのベース・イメージのビルド』](#)
  - [625 ページの『サンプルの構成済みの IBM MQ キュー・マネージャーのイメージのビルド』](#)

## サンプルの IBM MQ キュー・マネージャーのベース・イメージのビルド

IBM MQ を Docker で使用するには、まず IBM MQ のクリーン・インストール環境を含むベース・イメージを作成する必要があります。以下の手順は、GitHub でホストされているコードを使用して、サンプルのベース・イメージを作成する方法を示しています。

### このタスクについて

#### Make を使用した Docker イメージの作成

[mq-container GitHub リポジトリ](#)で提供される Make ファイルを使用して実動 Docker イメージを作成する場合は、GitHub の [Building a Docker image](#) に記載されている手順に従ってください。

#### docker を使用して手動で Docker イメージを作成する

docker を使用して手動でイメージを作成する場合は、以下のステップを実行してください。

### 手順

1. 前提条件パッケージをインストールします。  
これらの手順では、インストールが必要ないいくつかの Linux パッケージを使用します。

- Ubuntu の場合:

```
sudo apt-get install python git
```

- Red Hat Enterprise Linux の場合:

```
sudo yum install python git
```

2. コマンド `mkdir downloads` を発行して、`downloads` ディレクトリーを作成します。
3. Passport Advantage®を使用して、Linux 用の IBM MQ サーバー・イメージをダウンロードします。  
詳しくは、[電子ソフトウェア・ダウンロードを使用したインストール](#)を参照してください。

例えば、`WS_MQ_V9.0.5.0_LINUX_ON_X86_64_IM.tar.gz` ファイルを選択し、作成した `downloads` ディレクトリーに配置します。

注: **CD** ベース・イメージとして Ubuntu を使用することを計画している場合は、Debian インストールをダウンロードする必要があります。

4. IBM MQ server for Linux イメージ (tar.gz) ファイルを HTTP サーバーまたは FTP サーバーで使用可能にします。

これは、Docker イメージ・レイヤーのスペースを節約するためです。Docker ファイル内の命令ごとに、新しいイメージ・レイヤーが作成されます。

**ADD** 命令または **COPY** 命令を使用した後に **RUN** 命令を使用してインストールを行うと、追加またはコピーされたファイルが新しいイメージ・レイヤーにコミットされます。

追加されたレイヤーでファイルを削除しても、ファイルは前のレイヤーに存在し続けます。このため、**RUN** コマンド 1 つでダウンロードとインストールを行うことをお勧めします。これは、ファイルをネットワーク上に用意する必要があることを意味します。

例えば、以下のように、Python を使用して HTTP サーバーを実行して現行ディレクトリー内のすべてのファイルを提供することができます。

```
pushd downloads
nohup python -m SimpleHTTPServer 8000 &
popd
```

5. サポートされる Docker イメージを作成するためのサンプル・ファイルを GitHub から取り出します。

- **LTS** 以下のコマンドを発行します。

```
git clone -b mq-9-lts https://github.com/ibm-messaging/mq-docker mq-docker
```

- **CD** 以下のコマンドを発行します。

```
git clone https://github.com/ibm-messaging/mq-container mq-container
```

6. ローカルの IP アドレスを確認します。

使用するアドレスはローカル環境に固有のアドレスです。ただし、以下のコマンドを実行した場合に表示されるものでなければなりません。

```
ip addr show
```

localhost は使用できないことに注意してください。

7. 以下のコマンドを実行して、IBM MQ のベース・イメージを作成します。その際、MQ\_URL の IP アドレスとファイル名を、直前で識別した値で置き換えます。

以下に例を示します。

- **LTS**

```
sudo docker build --tag mq --build-arg MQ_URL=http://10.0.2.15:8000/
WS_MQ_V9.0.0.0_LINUX_ON_X86_64_IM.tar.gz mq-docker
```

- **CD**

```
sudo docker build --tag mq --build-arg MQ_URL=http://10.0.2.15:8000/
WS_MQ_V9.0.0.0_LINUX_ON_X86_64_IM.tar.gz mq-container/Dockerfile-server mq-container
```

## タスクの結果

これで、IBM MQ がインストールされた基本 Docker イメージが作成されました。

## サンプルの構成済みの IBM MQ キュー・マネージャーのイメージのビルド

汎用的な IBM MQ Docker ベース・イメージを作成した後に、独自の構成を適用してセキュアなアクセスを可能にする必要があります。これを行うには、汎用的なイメージを親として使用して、独自の Docker イメ

ージを作成します。以下の手順は、最小のセキュリティー構成でサンプルのイメージを作成する方法を示しています。

## 手順

1. 新しいディレクトリーを作成し、以下を内容とする `config.mqsc` というファイルを追加します。

```
DEFINE CHANNEL(PASSWORD.SVRCONN) CHLTYPE(SVRCONN)
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(BLOCKUSER) USERLIST('nobody') +
DESCR('Allow privileged users on this channel')
SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) DESCR('BackStop rule')
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(CHANNEL) CHCKCLNT(REQUIRED)
ALTER AUTHINFO(SYSTEM.DEFAULT.IDPWOS) AUTHTYPE(IDPWOS) ADOPTCTX(YES)
REFRESH SECURITY TYPE(CONNAUTH)
```

上記の例では、ユーザー ID とパスワードによる単純な認証が使用されることに注意してください。ただし、企業が必要とする任意のセキュリティー構成を適用できます。

2. 以下を内容とする `Dockerfile` というファイルを作成します。

```
FROM mq
RUN useradd johndoe -G mqm && \
    echo johndoe:passw0rd | chpasswd
COPY config.mqsc /etc/mqm/
```

ここで、

- `johndoe` は、追加するユーザー ID です。
- `passw0rd` は、元のパスワードです。

3. 以下のコマンドを使用して、カスタムの Docker イメージを作成します。

```
sudo docker build -t mymq .
```

「.」は、先ほど作成した2つのファイルが含まれているディレクトリーです。

Docker はそのイメージを使用して一時コンテナを作成し、残りのコマンドを実行します。

**RUN** コマンドで、`johndoe` というユーザーを追加してパスワード `passw0rd` を指定し、**COPY** コマンドで、親イメージにとって既知の特定の場所に `config.mqsc` ファイルを追加します。

4. カスタマイズした新しいイメージを実行して、先ほど作成したディスク・イメージを含む新しいコンテナを作成します。

新しいイメージ層では、実行する特定のコマンドを指定しなかったため、親イメージから継承されています。親のエントリー・ポイント (このコードは [GitHub](#) で提供されています):

- キュー・マネージャーを作成する
- キュー・マネージャーを開始する
- デフォルトのリスナーを作成する
- 次に、`/etc/mqm/config.mqsc` から MQSC コマンドを実行します

以下のコマンドを発行して、カスタマイズした新しいイメージを実行します。

```
sudo docker run \
  --env LICENSE=accept \
  --env MQ_QMGR_NAME=QM1 \
  --volume /var/example:/var/mqm \
  --publish 1414:1414 \
  --detach \
  mymq
```

説明:

### 最初の env パラメーター

IBM IBM WebSphere MQ のライセンスに同意したことを知らせる環境変数をコンテナに渡しています。LICENSE 変数を `view` に設定してライセンスを表示することもできます。

IBM MQ ライセンスについて詳しくは、[IBM MQ ライセンス情報](#) を参照してください。

## 2 番目の env パラメーター

使用するキュー・マネージャー名を設定しています。

## volume パラメーター

MQ で /var/mqm に書き込むものはすべて、実際にはホスト上の /var/example に書き込む必要があることをコンテナに知らせています。

このオプションは、後で永続データを保持したままコンテナを簡単に削除できることを意味します。このオプションにより、ログ・ファイルの表示も簡単になります。

## publish パラメーター

ホスト・システムのポートをコンテナのポートにマップしています。デフォルトでは、コンテナはコンテナ自身の内部 IP アドレスを使用して実行されます。つまり、ポートを公開するには、具体的にポートをマップする必要があります。

この例では、ホストのポート 1414 をコンテナのポート 1414 にマップしています。

## detach パラメーター

バックグラウンドでコンテナを実行します。

## タスクの結果

構成済みの Docker イメージを作成しました。docker ps コマンドを使用して実行中のコンテナを表示できます。コンテナで実行されている IBM MQ プロセスは、docker top コマンドを使用して表示できます。



**重要:** docker ps コマンドを使用してもコンテナが表示されない場合、コンテナは失敗した可能性があります。コマンド docker ps -a を使用して、失敗したコンテナを表示できます。

コンテナ ID は、docker ps -a コマンドを使用して表示され、docker run コマンドを発行したときにも表示されます。

docker logs \${CONTAINER\_ID} コマンドを使用して、コンテナのログを表示できます。

一般的な問題は、mqconfig が、Docker ホスト上の特定のカーネル設定が正しくないことを示していることです。カーネル設定は Docker のホストとコンテナの間で共有されるため、正しく設定する必要があります (UNIX and Linux システムでのハードウェア要件とソフトウェア要件を参照)。

例えば、オープン・ファイルの最大数は、コマンド sysctl fs.file-max=524288 を使用して設定できます。

## V 9.0.3 別々のコンテナでのローカル・バインディング・アプリケーションの実行

Docker にコンテナ間でのプロセス名前空間の共有が追加されました。IBM MQ キュー・マネージャーとは別のコンテナの IBM MQ へのローカル・バインディング接続を必要とするアプリケーションを実行できるようになりました。この機能は、IBM MQ 9.0.3 以降のキュー・マネージャーでサポートされます。

### このタスクについて

以下の制約事項に従う必要があります。

- Docker バージョン 1.12 以降を使用する必要があります。
- --pid 引数を使用して、各コンテナの PID 名前空間を共有する必要があります。
- --ipc 引数を使用して、各コンテナの IPC 名前空間を共有する必要があります。
- 以下のいずれかが必要です。
  1. --uts 引数を使用して、各コンテナの UTS 名前空間をホストと共有する。
  2. -h 引数または --hostname 引数を使用して、各コンテナを同じホスト名にする。

- IBM MQ データディレクトリは、`/var/mqm` ディレクトリの下にあるすべてのコンテナが利用可能なボリュームにマウントする必要があります。

この機能は、Docker 1.12 以降が既にインストールされている Linux システムで以下の手順を完了することによって試すことができます。

以下の例では、サンプルの IBM MQ Docker コンテナ・イメージを使用しています。このイメージの詳細については、[Github](#) を参照してください。

## 手順

1. 次のコマンドを発行して、ボリュームとして機能する一時ディレクトリを作成します。

```
mkdir /tmp/dockerVolume
```

2. 次のコマンドを発行して、1つのコンテナ内にキュー・マネージャー (QM1) を作成し、名前 `sharedNamespace` を指定します。

```
docker run -d -e LICENSE=accept -e MQ_QMGR_NAME=QM1 --volume /tmp/dockerVol:/mnt/mqm --uts host --name sharedNamespace ibmcom/mq
```

3. 次のコマンドを発行して、`ibmcom/mq` をベースとする `secondaryContainer` という2つ目のコンテナを始動します。ただし、キュー・マネージャーは作成しません。

```
docker run --entrypoint /bin/bash --volumes-from sharedNamespace --pid container:sharedNamespace --ipc container:sharedNamespace --uts host --name secondaryContainer -it --detach ibmcom/mq
```

4. 次のコマンドを発行して、2つ目のコンテナに対して `dspmq` コマンドを実行し、両方のキュー・マネージャーの状況を調べます。

```
docker exec secondaryContainer dspmq
```

5. 次のコマンドを実行して、もう一方のコンテナで実行されているキュー・マネージャーに対する `MQSC` コマンドを処理します。

```
docker exec -it secondaryContainer runmqsc QM1
```

## タスクの結果

これで、別々のコンテナでローカル・アプリケーションが実行されるようになりました。`dspmq`、`amqsput`、`amqsget`、`runmqsc` などのコマンドを、QM1 キュー・マネージャーへのローカル・バインディングとして、2つ目のコンテナから正常に実行することができます。

予期した結果にならない場合は、628 ページの『名前空間アプリケーションのトラブルシューティング』で詳細を参照してください。

### V 9.0.3 名前空間アプリケーションのトラブルシューティング

共有名前空間を使用する場合は、すべての名前空間 (IPC、PID、および UTS/hostname) とマウントされたボリュームを共有していることを確認する必要があります。共有していない場合、アプリケーションは動作しません。

従う必要がある制約事項のリストについては、627 ページの『別々のコンテナでのローカル・バインディング・アプリケーションの実行』を参照してください。

リストされているすべての制約事項をアプリケーションが満たしていないと、コンテナが始動しても、予期したように機能しないという問題が発生する可能性があります。

以下のリストに、一般的な原因と、制約事項のいずれかを満たすことを忘れた場合に見られる動作をまとめています。

- 名前空間 (UTS/PID/IPC) を共有したり、コンテナのホスト名を同じものとして設定したりするのを忘れてボリュームをマウントした場合、コンテナはキュー・マネージャーを認識できませんが、キュー・マネージャーと対話することはできません。

– **dspmq** コマンドの場合は、以下のようになります。

```
docker exec container dspmq
QMNAME(QM1)                STATUS(Status not available)
```

- **runmqsc** コマンドなど、キュー・マネージャーに接続しようとするコマンドの場合は、AMQ8146 エラー・メッセージを受け取るはずですが。

```
docker exec -it container runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2023.
Starting MQSC for queue manager QM1.
AMQ8146: IBM MQ queue manager not available
```

- 必要なすべての名前空間を共有するが、共有されたボリュームを /var/mqm ディレクトリーにマウントせず、有効な IBM MQ データ・パスがある場合も、コマンドの実行により AMQ8146 エラー・メッセージを受け取ります。

しかし、**dspmq** は、キュー・マネージャーをまったく認識できないので、代わりに空の応答を返します。

```
docker exec container dspmq
```

- 必要なすべての名前空間を共有するが、共有されたボリュームを /var/mqm ディレクトリーにマウントせず、有効な IBM MQ データ・パスがない (または IBM MQ データ・パスがまったくない) 場合は、データ・パスが IBM MQ インストール済み環境のキー・コンポーネントであるためさまざまなエラー・メッセージが表示されます。データ・パスがなければ、IBM MQ は動作できません。

以下のコマンドのいずれかを実行し、これらの例のような応答が表示される場合は、ディレクトリーがマウントされているか、または IBM MQ データ・ディレクトリーが作成されていることを確認する必要があります。

```
docker exec container dspmq
'No such file or directory' from /var/mqm/mqs.ini
AMQ6090: IBM MQ was unable to display an error message FFFFFFFF.
AMQffff
```

```
docker exec container dspmqver
AMQ7047: An unexpected error was encountered by a command. Reason code is 0.
```

```
docker exec container mqrc
<file path>/mqrc.c[1152]
lpi0btainQMDetails --> 545261715
```

```
docker exec container crtmqm QM1
AMQ8101: IBM MQ error (893) has occurred.
```

```
docker exec container strmqm QM1
AMQ6239: Permission denied attempting to access filesystem location '/var/mqm'.
AMQ7002: An error occurred manipulating a file.
```

```
docker exec container endmqm QM1
AMQ8101: IBM MQ error (893) has occurred.
```

```
docker exec container dlmqm QM1
AMQ7002: An error occurred manipulating a file.
```

```
docker exec container strmqweb
<file path>/mqrc.c[1152]
lpi0btainQMDetails --> 545261715
```

## Windows V 9.0.2 Linux IBM Cloud の IBM Cloud Product Insights サービスで使用するための IBM MQ の構成

IBM Cloud® Product Insights サービスは使用可能ではなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## Windows V 9.0.2 Linux IBM Cloud (formerly Bluemix) での IBM Cloud Product Insights サービス・インスタンスの作成

IBM Cloud Product Insights サービスは使用可能ではなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## Windows V 9.0.2 Linux IBM Cloud (formerly Bluemix) の IBM Cloud Product Insights サービス・インスタンスで使用するためのキュー・マネージャーの構成

IBM Cloud Product Insights サービスは使用可能ではなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## V 9.0.4 HTTP プロキシを介した IBM Cloud の IBM Cloud Product Insights への接続

IBM Cloud Product Insights サービスは使用可能ではなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## V 9.0.4 Product Insights への接続のトラブルシューティング

IBM Cloud Product Insights サービスは使用可能ではなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## V 9.0.2 Linux Salesforce プッシュ・トピックおよびプラットフォーム・イベントで使用する IBM MQ の構成

この情報を使用して、IBM MQ Bridge to Salesforce を構成してから実行することにより、Salesforce および IBM MQ ネットワークへのセキュリティーと接続をセットアップします。

### 始める前に

- IBM MQ Bridge to Salesforce は、Linux Linux for System x (64 ビット) で使用できます。IBM WebSphere MQ 6.0 以前で実行されているキュー・マネージャーに接続する場合、このブリッジはサポートされていません。
- **MQSeriesSFBridge** パッケージをインストールします。詳しくは、[Linux での IBM MQ サーバーのインストール](#)を参照してください。

### このタスクについて

Salesforce は、クラウド・ベースのカスタマー・リレーションシップ・マネジメント・プラットフォームです。Salesforce を使用して顧客データと対話を管理している場合、IBM MQ 9.0.2 では、IBM MQ Bridge to Salesforce を使用して Salesforce プッシュ・トピックおよびプラットフォーム・イベントにサブスクライブできます。これらは、その後 IBM MQ キュー・マネージャーにパブリッシュできます。そのキュー・マネージャーに接続されているアプリケーションは、プッシュ・トピックおよびプラットフォーム・イベント・データを有用な仕方で消費できます。

**V 9.0.4** IBM MQ 9.0.4 以降、ブリッジを使用して、Salesforce でのプラットフォーム・イベントのイベント・メッセージを作成することもできます。

IBM MQ Bridge to Salesforce の概要については、[図 1](#) のダイアグラムを参照してください。

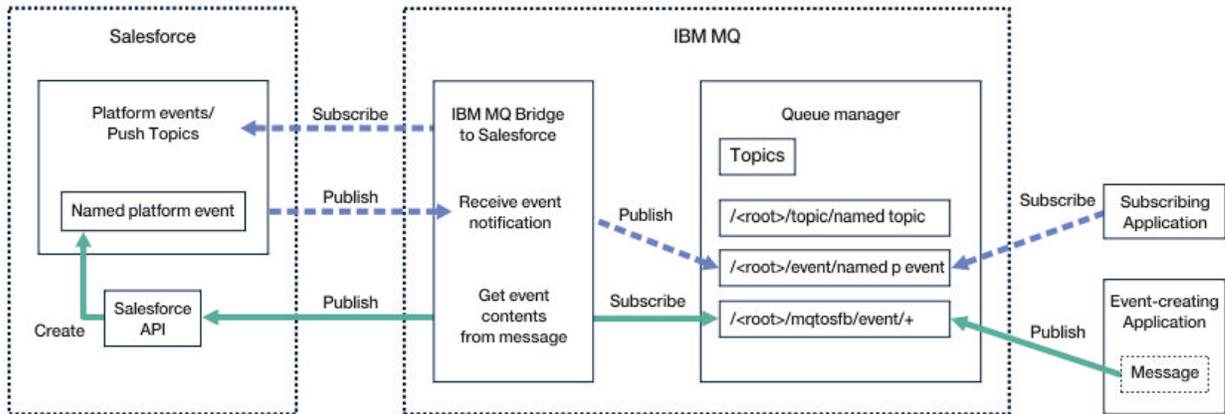


図 97. IBM MQ Bridge to Salesforce

プッシュ・トピックは、Salesforce のレコードに対する変更の通知を受信するために、Force.com ストリーミング API を使用するように定義する照会です。プッシュ・トピックの構成と Streaming API の使用方法について詳しくは、[ストリーミング API の概要](#) および [PushTopic を使った作業](#) を参照してください。

プラットフォーム・イベントはカスタマイズ可能なイベント・メッセージであり、Force.com プラットフォームでの作成またはコンSUME が可能なイベント・データを決定するように定義できます。プラットフォーム・イベントと Salesforce イベントとの違いについて詳しくは、[Enterprise messaging platform events](#) および [What is the difference between the Salesforce events](#) を参照してください。

- プッシュ・トピックとプラットフォーム・イベントのサブスクライブの構成を作成するには、[632 ページの『IBM MQ Bridge to Salesforce の構成』](#)を参照してください。
- **V 9.0.4** Salesforce プラットフォーム・イベントのイベント・メッセージを作成するための構成を作成するには、[637 ページの『Salesforce プラットフォーム・イベントのイベント・メッセージの作成』](#)を参照してください。

ブリッジからのデータをモニターするには、IBM MQ Console を使用する方法と、`amqsrua` コマンドで `-p` パラメーターを使用する方法の 2 つの方法があります。以下のように、ブリッジ全体の状況について一連のデータがパブリッシュされます。

- ある間隔で (STATUS/PUSHTOPIC ツリーの下で) 処理される合計プッシュ・トピック・メッセージ数。
- この間隔で表示されるプッシュ・トピックの数。
- ある間隔で (STATUS/PLATFORM ツリーの下で) 処理される合計プラットフォーム・イベント数。
- この間隔で表示されるプラットフォーム・イベントの数。
- **V 9.0.4** ある間隔で (STATUS/MQPE ツリーの下で) 処理される IBM MQ 作成のプラットフォーム・イベントの総数。
- **V 9.0.4** この間隔で表示される IBM MQ 作成のプラットフォーム・イベントの固有数。
- **V 9.0.4** この間隔で表示される IBM MQ 作成のプラットフォーム・イベントのパブリケーションの失敗数。

構成された Salesforce トピックごとに、さらにメッセージがパブリッシュされます。以下のように、IBM MQ トピックでは完全な Salesforce トピック名が使用され、オブジェクト名には `/event` または `/topic` が使用されます。

- ある間隔で処理されるメッセージの数。

ブリッジ・データをモニターするように IBM MQ Console を構成する場合は、次の [IBM MQ Bridge to Salesforce](#) の構成タスクのステップ 9 と 10 を参照してください。 `amqsrua` コマンドの使用については、[IBM MQ Bridge to Salesforce のモニター](#) を参照してください。

以下のタスクのステップに従い、IBM MQ Bridge to Salesforce を構成して実行します。

## 手順

1. IBM MQ Bridge to Salesforce を構成します。
2. **V 9.0.4**  
Salesforce プラットフォーム・イベントのイベント・メッセージを作成します。
3. IBM MQ Bridge to Salesforce を実行します。

## 関連情報

[runmqsfb \(IBM MQ Bridge to Salesforce の実行\)](#)

[IBM MQ Bridge to Salesforce のトレース](#)

**V 9.0.2**

**Linux**

## IBM MQ Bridge to Salesforce の構成

IBM MQ を構成し、IBM MQ Bridge to Salesforce パラメーターを入力すると、構成ファイルを作成することができ、Salesforce プッシュ・トピックとプラットフォーム・イベントを IBM MQ キュー・マネージャーに接続することができます。

## 始める前に

- x86-64 Linux プラットフォーム上の IBM MQ インストール済み環境に `MQSeriesSFBridge` パッケージをインストールしました。

## このタスクについて

このタスクでは、IBM MQ Bridge to Salesforce 構成ファイルを作成し、正常に Salesforce と IBM MQ に接続して、Salesforce プッシュ・トピックとプラットフォーム・イベントをサブスクライブできるようにするために必要な最小限のセットアップについて説明します。すべてのパラメーターの意味とオプションについては、`runmqsfb` コマンドを参照してください。独自のセキュリティー要件を考慮し、デプロイメントに適したパラメーターをカスタマイズする必要があります。

**V 9.0.4**

Salesforce プラットフォーム・イベントのイベント・メッセージを作成するための構成を作成するには、637 ページの『[Salesforce プラットフォーム・イベントのイベント・メッセージの作成](#)』を参照してください。

## Salesforce プッシュ・トピックとプラットフォーム・イベントのサブスクライブ

IBM MQ Bridge to Salesforce は、Salesforce と IBM MQ の両方への接続を確立すると、Salesforce プッシュ・トピックおよびプラットフォーム・イベントに対するサブスクリプションを作成します。ブリッジによるサブスクライブが必要なプッシュ・トピックやプラットフォーム・イベントの名前は、接続する前に、構成ファイルに組み込むか、コマンド行に追加する必要があります。

構成属性の 1 つは IBM MQ トピック・ツリーのルートであり、イベントはこのルートの下にパブリッシュされます。ブリッジはこのルートにアクセスし、完全な Salesforce トピック名 (例えば、`/MQ/SF/ROOT/topic/EscalatedCases`) を追加します。IBM MQ に接続しているモニタリング・トピックとアプリケーションは、`/topic/EscalatedCases` の下でプッシュ・トピックを検索し、`/event/NewCustomer__e` の下でプラットフォーム・イベントを検索する場合があります。

パブリッシュされたメッセージには、制御情報と、要求されたデータ・フィールドを含むデータ構造が含まれています。プッシュ・トピックの場合、データ構造は **subject** で、プラットフォーム・イベントの場合、この構造は **payload** です。ブリッジでは、Salesforce に定義されていないトピックやイベントにサブスクライブすることはできません。ブリッジでトピックへのサブスクライブを試みたときにエラーが発生すると、ブリッジは停止します。

トピック・オブジェクトは、IBM MQ で定義する必要はありませんが、ツリー内の最も近い親エレメントに基づいて、適切な権限が存在している必要があります。リパブリッシュされたメッセージには、デフォルトで、元のメッセージからの関連するデータ構造のみが含まれます。制御情報は削除されます。プラットフォーム・イベントの場合、パブリケーションはペイロード構造になっています。構成パラメーターの「**ブリッジ・プログラムの動作 (Behaviour of bridge program)**」セットの **Publish control data with the payload** 構成オプションにより、制御データを含むメッセージ全体の再パブリッシュが可能になります。詳しくは、[構成パラメーター](#)を参照してください。

各プッシュ・トピックおよびプラットフォーム・イベントには、Salesforce からのパブリケーションに関連した *ReplayID* があります。*ReplayID* を使用すると、サーバーに接続する際にパブリケーションの開始点を要求できます。Salesforce では、履歴を最大 24 時間維持できるので、最新のプッシュ・トピックおよびプラットフォーム・イベントが生成された時刻にブリッジが開始しなかった場合でも、ブリッジでそれらは失われません。ブリッジでは、以下の 2 つのサービスの品質がサポートされます。

### 最大 1 回

ブリッジでは、再始動に *ReplayId* は使用されません。ブリッジの再始動後、新規に生成されたプッシュ・トピックとプラットフォーム・イベントのみが処理されます。欠落しているパブリケーションのみを処理するように、アプリケーションを準備する必要があります。*ReplayId* はまだブリッジによって追跡され、キューにハード化されているので、他のサービス品質でブリッジを再始動し、現在の状態を把握することができます。

### 最低 1 回

ブリッジは、*ReplayId* を追跡し、キューにハード化します。ブリッジの再始動時には、永続 *ReplayId* を使用して、サーバーからのパブリケーションの開始点が要求されます。ギャップが 24 時間以内であった場合、それより古いパブリケーションが送信されます。トピックの *ReplayId* は、すべてのメッセージでハード化される訳ではありません。定期的な間隔で永続メッセージ内に書き込まれ、ブリッジのシャットダウン時にも書き込まれます。アプリケーションで、重複パブリケーションを確認する準備ができていない必要があります。

*ReplayId* は、新規に定義されたキューにメッセージとして書き込まれます。ブリッジの開始前に、このキュー (**SYSTEM.SALESFORCE.SYNCQ**) を定義しておく必要があります。**SYSTEM.SALESFORCE.SYNCQ** が存在しない場合、ブリッジは、サービスの品質モードに関係なく続行されません。関連する属性を持つキューを作成するための MQSC スクリプトが用意されています。DEFSOPT(EXCL) NOSHARE オプションを使用してキューを構成し、ブリッジ・プログラムの 1 つのインスタンスだけが **SYSTEM.SALESFORCE.SYNCQ** キューを更新できるようにする必要があります。

**V 9.0.4** プラットフォーム・イベントのイベント・メッセージを作成するための構成を作成するには、637 ページの『Salesforce プラットフォーム・イベントのイベント・メッセージの作成』を参照してください。

## 手順

1. キュー・マネージャーを作成して開始します。

- a) キュー・マネージャー (例えば、SQM1) を作成します。

```
crtmqm SQM1
```

- b) キュー・マネージャーを開始します。

```
stimqm SQM1
```

2. **注:** 既存のログインおよびセキュリティーの Salesforce 資格情報および自己署名証明書を使用するには、ステップにスキップしてください [633 ページの『3』](#)。

オプション: ご使用の Salesforce アカウント用のセキュリティー・トークンを作成します。

- a) Salesforce アカウントにログインします。
- b) ヘルプ項目 [Salesforce help: Reset your security token](#) のステップに従って、セキュリティー・トークンを作成またはリセットします。

3. CA 署名のセキュリティー証明書を Salesforce に作成します。

- a) 「**Force.com ホーム**」ページの「**管理**」メニューから「**セキュリティのコントロール**」を選択し、「**証明書と鍵の管理**」を選択します。  
「**証明書と鍵の管理**」ページが開きます。
  - b) **CA 署名済み証明書**をクリックする。  
「**証明書**」ページが開きます。
  - c) 「**Label (ラベル)**」フィールドに証明書の名前を入力し、Tab キーを押してから「**保存**」をクリックします。  
証明書と鍵の詳細情報が表示されます。
  - d) 「**Back to list: Certificates and keys (一覧に戻る: 証明書と鍵の管理)**」をクリックします。
  - e) 「**キーストアにエクスポート**」をクリックします。
  - f) 鍵ストアのパスワードを入力し、「**エクスポート**」をクリックします。
  - g) エクスポートした鍵ストアをローカル・ファイル・システムに保存します。
4. IBM 鍵管理 GUI を使用して、Salesforce からエクスポートした鍵ストアを開き、署名者証明書を追加します。
- a) **strmqikm** コマンドを実行して、IBM 鍵管理 GUI を開きます。  
詳しくは、[runmqckm](#)、[runmqakm](#)、および [strmqikm](#) を使用したデジタル証明書の管理を参照してください。
  - b) 「**鍵データベース・ファイルを開く**」をクリックし、Salesforce 鍵ストアのロケーションを参照します。
  - c) 「**開く**」をクリックして、「**鍵データベース・タイプ**」から「**JKS**」が選択されていることを確認し、「**OK**」をクリックします。
  - d) ステップ 3f で鍵ストア用に作成したパスワードを入力してから、「**OK**」をクリックします。
  - e) 「**鍵データベースの内容**」オプションから「**署名者証明書**」を選択します。
  - f) 「**挿入**」をクリックします。
  - g) 「**CA 証明書の追加**」リストから「**Verisign Inc.**」チェック・ボックスを選択し、「**OK**」をクリックします。
5. オプション: Salesforce アカウントに IBM MQ Bridge to Salesforce のアプリケーション接続を作成することによって、OAuth コンシューマ鍵と秘密情報を生成します。  
IBM MQ Bridge to Salesforce を実稼働環境で使用するときは、「**コンシューマ鍵**」と「**コンシューマの秘密**」のコードが必要です。
- a) 「**作成**」を選択し、「**Force.com ホーム**」ページの「**ビルド**」メニューから「**アプリケーション**」を選択します。  
「**アプリケーション**」ページが開きます。
  - b) 「**接続アプリケーション**」セクションから「**新規**」をクリックします。  
「**新規接続アプリケーション**」ページが開きます。
  - c) ご使用の IBM MQ Bridge to Salesforce の名前 (例えば、**MQBridgeToSalesforce**) を「**接続アプリケーション名**」に入力します。
  - d) 「**API Name (API 名)**」を入力します。  
タブで次のフィールドに移動すると、「**接続アプリケーション名**」が「**API Name (API 名)**」名前フィールドにコピーされます。
  - e) 「**Contact Email (連絡先 E メール)**」を入力します。
  - f) 「**API (OAuth 設定の有効化)**」セクションで「**OAuth 設定の有効化**」オプションを選択します。  
そのセクションの他のオプションが表示されます。
  - g) 「**コールバック URL**」 (例えば、<https://www.ibm.com>) を入力します。
  - h) 「**選択した OAuth 範囲**」サブセクションの「**利用可能な OAuth 範囲**」リストから「**フルアクセス (full)**」オプションを選択し、「**追加**」をクリックして、「**選択した OAuth 範囲**」リストに全アクセス権限を追加します。

- i) 「保存」をクリックします。
  - j) 「次へ進む」をクリックします。
  - k) 「コンシューマ鍵」と「コンシューマの秘密」のコードを書き留めます。
6. キュー・マネージャー上に必要な同期キューを作成します。

```
cat /opt/mqm/mqsf/samp/mqsfbSyncQ.mqsc | runmqsc SQM1
```

同期キューによって、アプリケーション間のイベント状態が維持されたり、キュー・マネージャーが再始動されたりします。キュー上にあると予想されるメッセージは1つだけなので、キュー項目数は少なく済みます。このキューに対して一度に実行できるブリッジのインスタンスは1つだけなので、デフォルト・オプションは排他的アクセス用に設定されます。

7. IBM MQ、Salesforce、および IBM MQ Bridge to Salesforce の動作に対する接続とセキュリティのパラメーターを使用して、構成ファイルを作成します。

```
runmqsfb -o new_config.cfg
```

既存の値が大括弧内に表示されます。Enter を押すと、既存の値を受け入れます。Space を押してから Enter を押すと、値をクリアします。値を入力してから Enter を押すと、新規の値を追加します。

- a) 以下のようにして、キュー・マネージャー SQM1 への接続のための値を入力します。

接続に必要な最小限の値は、キュー・マネージャー名、IBM MQ ベース・トピック・ルート、およびチャンネル名です。

```
Connection to Queue Manager
-----
Queue Manager or JNDI CF      : []SQM1
MQ Base Topic                 : []/sf
MQ Channel                    : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                    : []
V9.0.4 MQ Publication Error Queue : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                   : []
JNDI implementation class    : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL            : []
MQ Userid                     : []
MQ Password                   : []
```

注：ローカル接続を行う場合、チャンネル名は必要ありません。構成ファイルにキュー・マネージャー名とベース・トピックを指定する必要はありません。これらは、後でブリッジを実行するときにコマンド行に含めることができます。

- b) 以下のようにして、Salesforce への接続のための値を入力します。

接続に必要な最小限の値は、Salesforce のユーザー ID、パスワード、セキュリティ・トークン、およびログイン・エンドポイントです。実稼働環境では、OAuth セキュリティのコンシューマ鍵と秘密情報を追加できます。

```
Connection to Salesforce
-----
Salesforce Userid (reqd)      : []salesforce_login_email
Salesforce Password (reqd)    : []salesforce_login_password
Security Token (reqd)         : []Security_Token
Login Endpoint                 : [https://login.salesforce.com]
Consumer ID                    : []
Consumer Secret Key           : []
```

- c) 以下のようにして、TLS 接続のための証明書ストアの値を入力します。

TLS 接続に必要な最小限の値は、TLS 証明書の鍵ストアのパスと、鍵ストアのパスワードです。トラストストアのパスやパスワードが指定されていない場合は、鍵ストアとパスワードのパラメーターがトラストストアとパスワードに使用されます。IBM MQ キュー・マネージャー接続に TLS を使用している場合は、同じ鍵ストアを使用できます。

```
Certificate stores for TLS connections
-----
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
```

```
SQM1/ssl/key.jks
Keystore password      : []keystore_password
Trusted store for signer certificates : []
Trusted store password : []
Use TLS for MQ connection : [N]
```

d) 以下のようにして、IBM MQ Bridge to Salesforce の動作を構成するための値を入力します。

いずれの値も変更したり、指定したりする必要はありませんが、プッシュ・トピックやプラットフォーム・イベント名が分かっている場合は、ここで追加してください。また、ブリッジを実行する準備ができたときに、後でコマンド行に追加することもできます。構成ファイルまたはコマンド行にログ・ファイルを指定する必要があります。

```
Behaviour of bridge program
-----
PushTopic Names      : []
Platform Event Names : []
MQ Monitoring Frequency : [30]
At-least-once delivery? (Y/N) : [Y]
V9.0.4 Subscribe to MQ publications for platform events? (Y/N) : [N]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

8. オプション: IBM MQ サービスを作成して、プログラムの実行を制御します。サンプルの `mqsfbService.mqsc` ファイルを編集して、新しく作成された構成ファイルを指すようにし、コマンド・パラメーターにその他の変更を加えます。

```
cat modified mqsfbService.mqsc | runmqsc SQM1
```

9. **V9.0.1**

オプション: [IBM MQ コンソールの概要](#)の説明に従って、IBM MQ Console をセットアップします。

10. 注: MQ Console でブリッジに関するデータを表示する前に、ブリッジを少なくとも 1 回実行して、始動時に Salesforce と IBM MQ に接続する必要がある場合があります。ブリッジのメタトピックは、ブリッジの始動時にパブリッシュされます。

オプション: IBM MQ Console インスタンスにウィジェットを追加して構成し、Salesforce データを表示します。

- 「**ウィジェットの追加**」をクリックします。  
新しいウィジェットが開きます。
- 「**グラフ**」を選択します。
- 新規ウィジェットのタイトル・バーにある「**ウィジェットの構成**」アイコンをクリックします。
- オプション: 「**ウィジェット・タイトル**」を入力します。
- 「**ソース**」ドロップダウン・メニューの「**モニターするリソース (Resource to monitor)**」から「**Salesforce ブリッジ**」を選択します。
- 「**保存**」をクリックします。

## タスクの結果

IBM MQ Bridge to Salesforce が Salesforce プッシュ・トピックおよびプラットフォーム・イベントをサブスクライブし、それらを IBM MQ ネットワークにパブリッシュするために使用する構成ファイルを作成しました。

## 次のタスク

643 ページの『[IBM MQ Bridge to Salesforce の実行](#)』のステップに従います。

### 関連情報

[runmqsf \(IBM MQ Bridge to Salesforce の実行\)](#)

[IBM MQ Bridge to Salesforce のトレース](#)

[IBM MQ Bridge to Salesforce のモニター](#)

# Salesforce プラットフォーム・イベントのイベント・メッセージの作成

IBM MQ を構成し、IBM MQ Bridge to Salesforce parameters を入力して構成ファイルを作成し、ブリッジを使用して Salesforce プラットフォーム・イベントのイベント・メッセージを作成できます。

## 始める前に

- x86-64 Linux プラットフォーム上の IBM MQ インストール済み環境に **MQSeriesSFBridge** パッケージをインストールしました。

## このタスクについて

このタスクでは、IBM MQ Bridge to Salesforce コンフィギュレーションファイルを作成し、Salesforce プラットフォームイベント用のイベントメッセージを作成できるように、正常に Salesforce と IBM MQ 接続するために必要な最小限のセットアップを行います。すべてのパラメーターの意味とオプションについて詳しくは、[runmqsfb](#) コマンドを参照してください。独自のセキュリティー要件を考慮し、デプロイメントに適したパラメーターをカスタマイズする必要があります。

プッシュ・トピックとプラットフォーム・イベントのサブスクライブの構成を作成するには、[632 ページ](#)の『[IBM MQ Bridge to Salesforce の構成](#)』を参照してください。

## Salesforce プラットフォーム・イベントのイベント・メッセージの作成

IBM MQ 9.0.4 以降では、IBM MQ アプリケーションを使用して、キュー・マネージャー・トピック / `root/mqtosfb/event/+` に書き込まれるメッセージを作成できます。ブリッジはトピックをサブスクライブし、メッセージからコンテンツを取得し、それを使用して Salesforce プラットフォーム・イベントのイベント・メッセージをパブリッシュします。プラットフォーム・イベントの詳細については、Salesforce 開発者資料の [Delivering custom notifications with platform event](#) を参照してください。

ブリッジを有効にしてイベント・メッセージを作成するには、IBM MQ 9.0.2 でプッシュ・トピックおよびプラットフォーム・イベントのサブスクライブに使用された属性に加えて 2 つの属性を指定する必要があります。

- キュー・マネージャーへの接続のブリッジ構成属性に **MQ Publication Error Queue** の名前を作成して追加します。
- 「**Behavior of bridge program**」を定義するブリッジ構成属性で **Subscribe to MQ publications for platform events** オプションを Y に設定します。

ブリッジを使用してそのプラットフォーム・イベントのイベント・メッセージを作成できるようにするには、まず、Salesforce でプラットフォーム・イベントを作成し、コンテンツ・フィールドを定義する必要があります。プラットフォーム・イベント名およびそのコンテンツによって、ブリッジによって処理される IBM MQ メッセージをどのようにフォーマット設定する必要があるかが決定されます。例えば、Salesforce プラットフォーム・イベント **Object name** が `MQPlatformEvent1` であり、2 つのカスタム定義フィールドが **API name** `MyText__c` および `Name__c` のテキスト・フィールドである場合、`/root/mqtosfb/event/MQPlatformEvent1__e` トピックにパブリッシュされる IBM MQ メッセージは、以下のように正しい形式の JSON でなければなりません。

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

メッセージは、IBM MQ Bridge to Salesforce が MQFMT\_STRING 形式のメッセージ本体として認識できるようにフォーマット設定する必要があります。

ステップ [640 ページ](#)の『[7](#)』を参照して、Salesforce でプラットフォーム・イベントを作成するか、イベント・メッセージを作成するプラットフォーム・イベントが既にある場合はこのステップをスキップします。IBM MQ メッセージを、Salesforce プラットフォーム・イベントで設定されているフィールドと一致するようにフォーマット設定する必要があります。Salesforce プラットフォーム・イベント内のフィールドは、オプションまたは必須として指定できます。詳細については、Salesforce 開発者資料の [Platform Event Field](#) を参照してください。

ブリッジが実行されている場合、ブリッジは指定された IBM MQ トピックをサブスクライブします。

- ブリッジ構成で **At-most-once** のサービス品質を指定すると、ブリッジが作成するサブスクリプションが非永続になります。ブリッジが実行されていないときに IBM MQ アプリケーションによって作成されたパブリケーションは処理されません。
- ブリッジ構成で **At-least-once** のサービス品質を指定すると、ブリッジが作成するサブスクリプションは永続的になります。これは、ブリッジが実行中でないときに IBM MQ アプリケーションによって作成されたパブリケーションをブリッジが処理できることを意味します。永続サブスクリプションには、既知のサブスクリプションおよびクライアント ID が必要です。ブリッジは、サブスクリプション名として `D_SUB_RUNMQSFB` を使用し、クライアント ID として `runmqsfb_1` を使用します。

ブリッジが Salesforce プッシュ・トピックおよびプラットフォーム・イベントのサブスクライブのために使用され、イベント・メッセージの作成のためには使用されない場合、構成が変更されると、ブリッジは永続サブスクリプションの削除を試行し、サブスクリプションが孤立します。

ブリッジが作成した永続サブスクリプションを削除するには、以下のようになります。

#### IBM MQ Explorer を使用します。

ブリッジが使用するキュー・マネージャーの「サブスクリプション」フォルダーを開き、末尾が `:D_SUB_RUNMQSFB` のサブスクリプション名を探します。ここで、トピック文字列は `/sf/mqtosfb/event+` です。サブスクリプション名を右クリックして、「削除」をクリックします。サブスクリプションが使用中であることを示すエラーが表示された場合は、ブリッジが実行中である可能性があります。ブリッジを停止し、サブスクリプションの削除を再試行してください。

#### runmqsc を使用した、サブスクリプションの検索と削除。

**runmqsc** インターフェースを開始し、`DISPLAY SUB (*)` を実行します。末尾が `:D_SUB_RUNMQSFB` のサブスクリプション名 **SUB** を検索します。 `delete sub` コマンドを実行し、削除するサブスクリプションの **SUBID** を含めます。例: `DELETE SUB SUBID(414D5120514D3120202020202020205C589459987E8620)`

#### At-most-once のサービス品質を指定した、ブリッジの停止と開始。

**At-least-once** のサービス品質である `At-least-once delivery? (Y/N) :[Y]` を指定してブリッジを開始すると、作成されるサブスクリプションは永続的になります。サブスクリプションを削除するには、構成ファイルでサービス品質を `At-least-once delivery? (Y/N) :[N]` に変更し、ブリッジを再始動します。永続サブスクリプションが削除され、非永続サブスクリプションが作成されます。

## 手順

1. キュー・マネージャーを作成して開始します。

a) キュー・マネージャー (例えば、PEQM1) を作成します。

```
crtmqm PEQM1
```

b) キュー・マネージャーを開始します。

```
strmqm PEQM1
```

2. 注: 既存のログインとセキュリティの Salesforce 資格情報、および自己署名証明書を使用する場合は、ステップ 4 にスキップします。

オプション: ご使用の Salesforce アカウント用のセキュリティ・トークンを作成します。

a) Salesforce アカウントにログインします。

b) ヘルプ項目 [Salesforce help: Reset your security token](#) のステップに従って、セキュリティ・トークンを作成またはリセットします。

3. Salesforce で自己署名セキュリティ証明書を作成します。

a) 「**Force.com ホーム**」ページの「管理」メニューから「**セキュリティのコントロール**」を選択し、「**証明書と鍵の管理**」を選択します。  
「**証明書と鍵の管理**」ページが開きます。

- b) 「自己署名証明書の作成」をクリックします。  
「証明書」ページが開きます。
  - c) 「Label (ラベル)」フィールドに証明書の名前を入力し、Tab キーを押してから「保存」をクリックします。  
証明書と鍵の詳細情報が表示されます。
  - d) 「Back to list: Certificates and keys (一覧に戻る: 証明書と鍵の管理)」をクリックします。
  - e) 「キーストアにエクスポート」をクリックします。
  - f) 鍵ストアのパスワードを入力し、「エクスポート」をクリックします。
  - g) エクスポートした鍵ストアをローカル・ファイル・システムに保存します。
4. IBM 鍵管理 GUI を使用して、Salesforce からエクスポートした鍵ストアを開き、署名者証明書を追加します。
- a) `strmqikm` コマンドを実行して、IBM 鍵管理 GUI を開きます。詳しくは、[runmqckm](#)、[runmqakm](#)、および [strmqikm](#) を使用したデジタル証明書の管理を参照してください。
  - b) 「鍵データベース・ファイルを開く」をクリックし、Salesforce 鍵ストアのロケーションを参照します。
  - c) 「開く」をクリックして、「鍵データベース・タイプ」から「JKS」が選択されていることを確認し、「OK」をクリックします。
  - d) ステップ 3f で鍵ストア用に作成したパスワードを入力してから、「OK」をクリックします。
  - e) 「鍵データベースの内容」オプションから「署名者証明書」を選択します。
  - f) 「挿入」をクリックします。
  - g) 「CA 証明書の追加」リストから「Verisign Inc.」チェック・ボックスを選択し、「OK」をクリックします。
5. オプション: Salesforce アカウントに IBM MQ Bridge to Salesforce のアプリケーション接続を作成することによって、OAuth コンシューマ鍵と秘密情報を生成します。  
IBM MQ Bridge to Salesforce を実稼働環境で使用するときには、「コンシューマ鍵」と「コンシューマの秘密」のコードが必要です。
- a) 「作成」を選択し、「Force.com ホーム」ページの「ビルド」メニューから「アプリケーション」を選択します。  
「アプリケーション」ページが開きます。
  - b) 「接続アプリケーション」セクションから「新規」をクリックします。  
「新規接続アプリケーション」ページが開きます。
  - c) ご使用の IBM MQ Bridge to Salesforce の名前 (例えば、MQBridgeToSalesforce) を「接続アプリケーション名」に入力します。
  - d) 「API Name (API 名)」を入力します。  
タブで次のフィールドに移動すると、「接続アプリケーション名」が「API Name (API 名)」名前フィールドにコピーされます。
  - e) 「Contact Email (連絡先 E メール)」を入力します。
  - f) 「API (OAuth 設定の有効化)」セクションで「OAuth 設定の有効化」オプションを選択します。  
そのセクションの他のオプションが表示されます。
  - g) 「コールバック URL」 (例えば、https://www.ibm.com) を入力します。
  - h) 「選択した OAuth 範囲」サブセクションの「利用可能な OAuth 範囲」リストから「フルアクセス (full)」オプションを選択し、「追加」をクリックして、「選択した OAuth 範囲」リストに全アクセス権限を追加します。
    - i) 「保存」をクリックします。
    - j) 「次へ進む」をクリックします。
    - k) 「コンシューマ鍵」と「コンシューマの秘密」のコードを書き留めます。
6. 必要な同期キューおよびエラー・キューをキュー・マネージャー上に作成します。

```
cat /opt/mqm/mqsf/samp/mqsfSyncQ.mqsc | runmqsc PEQM1
```

同期キューによって、アプリケーション間のイベント状態が維持されたり、キュー・マネージャーが再始動されたりします。キュー上にあると予想されるメッセージは1つだけなので、キュー項目数は少なくても済みます。このキューに対して一度に実行できるブリッジのインスタンスは1つだけなので、デフォルト・オプションは排他的アクセス用に設定されます。ブリッジを使用してプラットフォーム・イベントのイベント・メッセージを作成できるようにするには、まず、エラー・キューを作成する必要があります。エラー・キューは、Salesforce で正常に処理できないメッセージに使用されます。ステップ 640 ページの『8.a』に示すように、ブリッジ構成パラメーター・セクション **Connection to Queue Manager** にエラー・キュー名を追加する必要があります。

7. オプション: ご使用の Salesforce アカウントでプラットフォーム・イベント・オブジェクトを作成します。

a) 「**Force.com ホーム (Force.com Home)**」ページの「**開発 (Develop)**」メニューから「**プラットフォーム・イベント (Platform Events)**」を選択し、「**新規プラットフォーム・イベント (New Platform Event)**」をクリックします。

「**新規プラットフォーム・イベント (New Platform Event)**」ページが開きます。

b) 「**ラベル (Label)**」フィールドおよび「**複数ラベル (Plural Label)**」フィールドに入力します。

c) 「**保存**」をクリックします。

「**プラットフォーム・イベント定義の詳細 (Platform Event Definition Detail)**」ページが開きます。

d) **Custom Fields & Relationships** を定義します。

例えば、ラベルが *MyText* および *Name* の 2 つのフィールドを追加し、「**データ・タイプ (Data Type)**」のフィールド長をそれぞれ *Text(64)* および *Text(32)* に設定する場合があります。

プラットフォーム・イベントを作成し、それに対して **Custom Fields and Relationships** を定義しました。ブリッジで処理するメッセージを書き込むことができる IBM MQ トピックとして、プラットフォーム・イベントの *Platform Object name* または *API name* を使用します。例えば、**AMQSPUBA** サンプルを使用して以下の JSON 形式のメッセージを */sf/mqtosfb/event/Salesforce Platform Object Name/API name* トピックに追加する場合があります。

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

**AMQSPUBA** サンプルを実行して、ブリッジの開始後にメッセージを作成できます。MQ *installation location/samp/bin* ディレクトリーから、以下のコマンドを実行します。

```
./amqspub /sf/mqtosfb/event/Salesforce Platform Object Name/API name PEQM1
```

プロンプトが表示されたら、JSON 形式のメッセージを入力します。

8. IBM MQ、Salesforce、および IBM MQ Bridge to Salesforce の動作に対する接続とセキュリティーのパラメーターを使用して、構成ファイルを作成します。

```
runmqsf -o new_config.cfg
```

既存の値が大括弧内に表示されます。Enter を押すと、既存の値を受け入れます。Space を押してから Enter を押すと、値をクリアします。値を入力してから Enter を押すと、新規の値を追加します。

a) 以下のようにして、キュー・マネージャー PEQM1 への接続のための値を入力します。

接続に必要な最小値は、キュー・マネージャー名、IBM MQ ベース・トピック・ルート、エラー・キュー名、およびチャンネル名です。

```
Connection to Queue Manager
-----
Queue Manager or JNDI CF   : []PEQM1
MQ Base Topic             : []/sf
MQ Channel                : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                : []
MQ Publication Error Queue : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL               : []
JNDI implementation class  : [com.sun.jndi.fscontext.RefFSContextFactory]
```

```
JNDI provider URL      : []
MQ Userid              : []
MQ Password            : []
```

注: ローカルに接続している場合は、チャンネル名は必要ありません。構成ファイルにキュー・マネージャー名とベース・トピックを指定する必要はありません。これらは、後でブリッジを実行するときにコマンド行に含めることができます。

- b) 以下のようにして、Salesforce への接続のための値を入力します。

接続に必要な最小限の値は、Salesforce のユーザー ID、パスワード、セキュリティー・トークン、およびログイン・エンドポイントです。実稼働環境では、OAuth セキュリティーのコンシューマー鍵と秘密情報を追加できます。

```
Connection to Salesforce
-----
Salesforce Userid (reqd)  : []salesforce_login_email
Salesforce Password (reqd) : []salesforce_login_password
Security Token (reqd)    : []Security_Token
Login Endpoint           : [https://login.salesforce.com]
Consumer ID              : []
Consumer Secret Key     : []
```

- c) 以下のようにして、TLS 接続のための証明書ストアの値を入力します。

TLS 接続に必要な最小限の値は、TLS 証明書の鍵ストアのパスと、鍵ストアのパスワードです。トラストストアのパスやパスワードが指定されていない場合は、鍵ストアとパスワードのパラメーターがトラストストアとパスワードに使用されます。IBM MQ キュー・マネージャー接続に TLS を使用している場合は、同じ鍵ストアを使用できます。

```
Certificate stores for TLS connections
-----
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/PEQM1/ssl/key.jks
Keystore password                     : []keystore_password
Trusted store for signer certificates : []
Trusted store password                 : []
Use TLS for MQ connection             : [N]
```

- d) 以下のようにして、IBM MQ Bridge to Salesforce の動作を構成するための値を入力します。

ブリッジを使用してイベント・メッセージを作成するには、**Subscribe to MQ publications for platform events** オプションをデフォルトの *N* から *Y* に変更する必要があります。また、構成ファイルまたはコマンド行で、ログ・ファイルを指定する必要もあります。

```
Behaviour of bridge program
-----
PushTopic Names           : []
Platform Event Names      : []
MQ Monitoring Frequency   : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

9. オプション: IBM MQ サービスを作成して、プログラムの実行を制御します。サンプルの `mqsfbService.mqsc` ファイルを編集して、新しく作成された構成ファイルを指すようにし、コマンド・パラメーターにその他の変更を加えます。

```
cat modified mqsfbService.mqsc | runmqsc PEQM1
```

## 10. **V9.0.1**

オプション: [IBM MQ コンソールの概要](#)の説明に従って、IBM MQ Console をセットアップします。

11. オプション: IBM MQ Console インスタンスにウィジェットを追加して構成し、Salesforce データを表示します。

- a) 「**ウィジェットの追加**」をクリックします。

新しいウィジェットが開きます。

- b) 「グラフ」を選択します。
- c) 新規ウィジェットのタイトル・バーにある「ウィジェットの構成」アイコンをクリックします。
- d) オプション: 「ウィジェット・タイトル」を入力します。
- e) 「モニターするリソース (Resource to monitor)」の「ソース」ドロップダウン・メニューから「Salesforce ブリッジ」を選択します。
- f) 「リソース・クラス」ドロップダウン・メニューから「ブリッジ状況 (Bridge Status)」を選択します。
- g) 「リソース・タイプ」ドロップダウン・メニューから「MQ 作成のプラットフォーム・イベント (MQ-created Platform Events)」を選択します。
- h) 「リソース・エレメント」ドロップダウン・メニューから、「MQ 作成のプラットフォーム・イベントの合計 (Total MQ-created Platform Events)」を選択します。
- i) 「保存」をクリックします。

IBM MQ 作成のプラットフォーム・イベントの合計数を表示するために IBM MQ Console を構成しました。ブリッジが実行中で、`/sf/mqtosfb/event/Salesforce Platform Object Name/API name` トピックへのメッセージの書き込みを開始すると、このウィジェットに、ブリッジが作成したメッセージ・イベントの総数が表示されます。

## V9.0.4 IBM MQ Bridge to Salesforce のメッセージの形式およびエラー・メッセージ

IBM MQ Bridge to Salesforce によって処理されるメッセージのフォーマット設定に関する情報。

アプリケーションは、`/root/mqtosfb/event/MQPlatformEvent1__e` など、特定のキュー・マネージャー・トピックにメッセージを書き込みます。ブリッジはトピックをサブスクライブし、メッセージからコンテンツを取得し、それを使用して Salesforce プラットフォーム・イベントのイベント・メッセージをパブリッシュします。

ブリッジを使用してそのプラットフォーム・イベントのイベント・メッセージを作成できるようにするには、まず、Salesforce でプラットフォーム・イベントを作成し、コンテンツ・フィールドを定義する必要があります。プラットフォーム・イベント名およびそのコンテンツによって、ブリッジによって処理される IBM MQ メッセージをどのようにフォーマット設定する必要があるかが決定されます。例えば、Salesforce プラットフォーム・イベント **Object name** が `MQPlatformEvent1` であり、2つのカスタム定義フィールドが **API name** `MyText__c` および `Name__c` のテキスト・フィールドである場合、`/root/mqtosfb/event/MQPlatformEvent1__e` トピックにパブリッシュされる IBM MQ メッセージは、以下のように正しい形式の JSON でなければなりません。

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

ブリッジによって消費または生成されるメッセージは、JSON 形式のテキスト (MQSTR) メッセージです。入力メッセージは単純な JSON であり、プログラムでは文字列の連結を使用して、入力メッセージを生成できます。

## エラー・メッセージ

例えば、メッセージがテキスト形式でない場合、または Salesforce によって、例えば、プラットフォーム・イベント名が存在しない場合に、ブリッジによってエラーが検出されることがあります。入力メッセージの処理中にエラーが発生した場合、メッセージは、エラーを記述するプロパティとともにブリッジ・エラー・キューに移動されます。エラーは、ブリッジの `stderr` ストリームにも書き込まれます。

Salesforce によって生成されるエラーは、JSON です。以下に、形式が無効なメッセージによって発生するいくつかのエラーを示します。

プラットフォーム・イベントの内容が正しくありません。状況 400 テキスト

```
{ "message": "No such column 'Name__c' on subject of type MQPlatformEvent2__e", "errorCode": "INVALID_FIELD" }
```

プラットフォーム・イベント名が無効です。状況 404 テキスト

```
{"errorCode":"NOT_FOUND","message":"The requested resource does not exist"}
```

JSON が正しくありません。状況 400 テキスト

```
{"errorCode":"NOT_FOUND","message":"The requested resource does not exist"}
```

メッセージが JSON ではありません。状況 400 テキスト

```
[{"message":  
  "Unexpected character ('h' (code 104)): expected a valid value (number, String, array,  
  object, 'true', 'false' or 'null') at [line:1, column:2]",  
  "errorCode":"JSON_PARSER_ERROR"}]
```

テキスト・メッセージではありません (Salesforce に送信されません)。

```
Error: Publication on topic ' /sf/mqtosfb/event/MQPlatformEvent1' does not contain a text  
formatted message
```

V 9.0.2

Linux

## IBM MQ Bridge to Salesforce の実行

IBM MQ Bridge to Salesforce を実行して、Salesforce および IBM MQ に接続します。接続されると、ブリッジによって、Salesforce トピックに対するサブスクリプションが作成され、IBM MQ トピックにメッセージがリパブリッシュされます。 **V 9.0.4** From IBM MQ 9.0.4 以降では、ブリッジによって Salesforce プラットフォーム・イベントのイベント・メッセージが作成されることもあります。

### 始める前に

以下のタスクの構成ステップを完了しておきます。

- 632 ページの『IBM MQ Bridge to Salesforce の構成』
- **V 9.0.4** 637 ページの『Salesforce プラットフォーム・イベントのイベント・メッセージの作成』

### このタスクについて

前のステップで作成した構成ファイルを使用して、IBM MQ Bridge to Salesforce を実行します。構成ファイルにすべての必須パラメーターを組み込んでいない場合、コマンド行に必ずそれらを組み込んでください。

### 手順

1. サブスクリプション先の Salesforce のプッシュ・トピックまたはプラットフォーム・イベント **V 9.0.4**、あるいはイベント・メッセージを作成するプラットフォーム・イベントを定義します。
2. IBM MQ Bridge to Salesforce を開始して、Salesforce とキュー・マネージャーに接続します。ブリッジを実行して Salesforce イベントをサブスクリプションする場合は、ステップ 1 で定義したプッシュ・トピックまたはプラットフォーム・イベントの名前を組み込みます。

```
runmqsfb -f new_config.cfg -r logfile -p PushtopicName -e eventName
```

ブリッジが接続されると、次のようなメッセージが返されます。

IBM MQ 9.0.2 時

```
Successful connection to queue manager QM1  
Successful login to Salesforce at https://eu11.salesforce.com  
Ready to process events.
```

- ブリッジを使用して Salesforce プッシュ・トピックおよびプラットフォーム・イベントをサブスクライブする場合:

```
Successful connection to queue manager QM1
Warning: Subscribing to MQ-created platform events is not enabled.
Successful login to Salesforce at https://eu11.salesforce.com
Ready to process events.
```

- ブリッジを使用して、Salesforce プラットフォーム・イベントのイベント・メッセージを作成する場合:

```
Successful connection to queue manager QM1
Successful login to Salesforce at https://eu11.salesforce.com
Successful subscription to '/sf/mqtosfb/event/+' for MQ-created platform events
Ready to process events.
```

3. オプション: ブリッジの実行後に、接続が正常に行われなかったことを示すメッセージが返された場合は、キュー・マネージャーと Salesforce への接続のトラブルシューティングを行います。

- a) デバッグ・オプション 1 を指定して、デバッグ・モードでコマンドを発行します。

```
runmqsfcb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 1
```

ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが簡潔モードで表示されます。

- b) デバッグ・オプション 2 を指定して、デバッグ・モードでコマンドを発行します。

```
runmqsfcb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 2
```

ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが冗長モードで表示されます。全出力がログ・ファイルに書き込まれます。

4. Salesforce インターフェースを使用してイベントを生成し、データベース内のレコードを変更します。
5. IBM MQ Console に移動し、前のタスクで構成したウィジェットに、プッシュ・トピックに対する変更が表示されていることを確認します。

## 次のタスク

`MQSFB_EXTRA_JAVA_OPTIONS` 変数を使用して、例えば、IBM MQ トレースを有効化する JVM プロパティを渡します。詳しくは、[IBM MQ Bridge to Salesforce のトレース](#)を参照してください。

### 関連情報

[runmqsfcb \(IBM MQ Bridge to Salesforce の実行\)](#)

[IBM MQ Bridge to Salesforce のモニター](#)

## の構成

IBM MQ Advanced キュー・マネージャーと IBM Blockchain を安全に接続するために、IBM MQ Bridge to blockchain をセットアップして実行します。IBM MQ Advanced キュー・マネージャーに接続されているメッセージ・アプリケーションを使用すると、ブリッジを使用して、ブロックチェーンのリソースの状態に非同期で接続し、この状態を検索および更新できます。

### 始める前に

- IBM MQ Bridge to blockchain は、IBM MQ Advanced キュー・マネージャーへの接続にのみ使用できません。
- キュー・マネージャーは、ブリッジと同じコマンド・レベルである必要があります (IBM MQ 9.0.4 など)。

- IBM MQ Bridge to blockchain は、Hyperledger Fabric 1.0 architecture に基づくブロックチェーン・ネットワークでの使用がサポートされています。

## このタスクについて

ブロックチェーンとは、分散型の共有デジタル台帳であり、ネットワーク上のピアの間で合意されたトランザクションを表すブロックがチェーン状に連なったもので構成されます。チェーン内のそれぞれのブロックは前のブロックにリンクしているため、最終的には最初のトランザクションにリンクしています。

IBM Blockchain は、Hyperledger Fabric 上に構築され、Docker でローカルに開発するか、IBM Cloud (formerly Bluemix®) のコンテナ・クラスターで開発できます。また、実稼働環境で IBM Blockchain ネットワークをアクティブ化して使用し、セキュリティ、プライバシー、およびパフォーマンスのレベルの高いビジネス・ネットワークを構築し、管理することもできます。詳細については、[IBM Blockchain Platform](#) を参照してください。

Hyperledger Fabric は、Hyperledger Project のメンバー (初期コード・コントリビューターとして IBM を含む) によって共同で開発された、オープン・ソースのエンタープライズ・ブロックチェーン・フレームワークです。Hyperledger Project、または Hyperledger は、業界横断のブロックチェーン・テクノロジーを推進するための Linux Foundation オープン・ソースのグローバルなコラボレーション・イニシアチブです。詳細については、[IBM Blockchain](#)、[Hyperledger Projects](#)、および [Hyperledger Fabric](#) を参照してください。

IBM MQ Advanced および IBM Blockchain を既に使用している場合は、IBM MQ Bridge to blockchain を使用して、単純な照会や更新を送信し、ブロックチェーン・ネットワークから応答を受信できます。そのようにして、オンプレミスの IBM ソフトウェアをクラウドのブロックチェーン・サービスに統合できます。

ブリッジ操作プロセスの概要は、[Figure 1](#) に記載されています。ユーザー・アプリケーションは、JSON 形式のメッセージを IBM MQ Advanced キュー・マネージャー上の入力/要求キューに入れます。ブリッジはキュー・マネージャーに接続し、入力/要求キューからメッセージを取得し、JSON が正しくフォーマット設定されていることを確認してから、照会または更新をブロックチェーンに発行します。ブロックチェーンによって返されるデータは、ブリッジによって解析され、元の IBM MQ 要求メッセージに定義されているように応答キューに入れられます。ユーザー・アプリケーションは、キュー・マネージャーに接続して、応答キューから応答メッセージを取得し、その情報を使用できます。

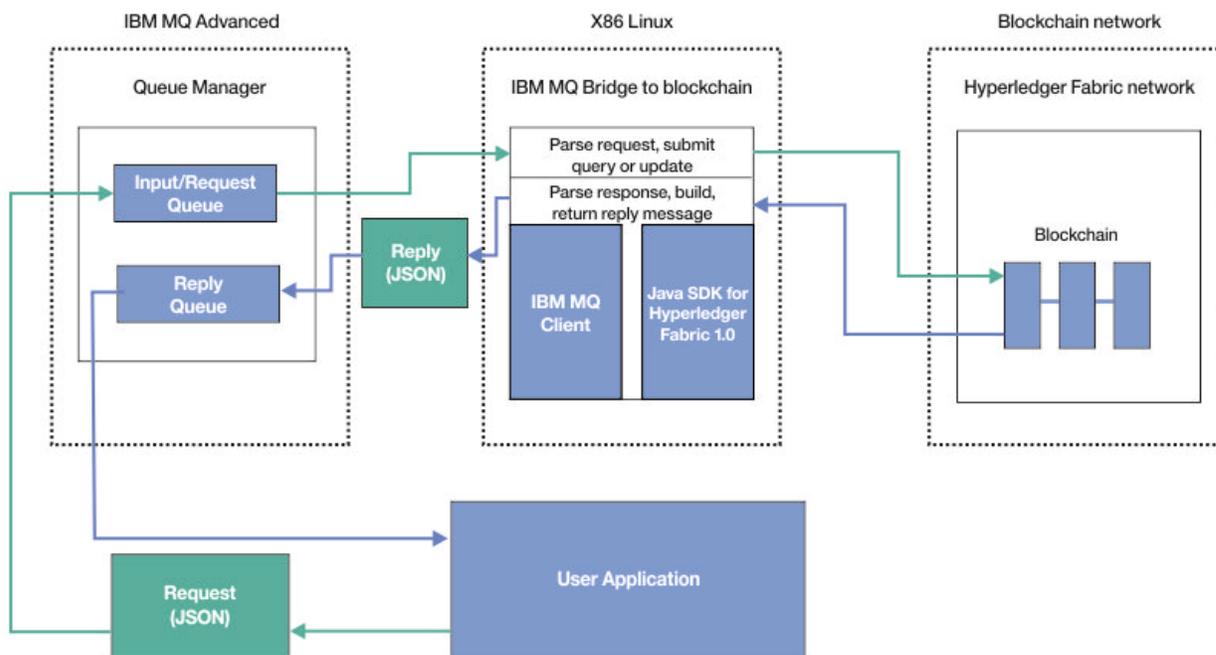


図 98. IBM MQ Bridge to blockchain

参加者またはピアとしてブロックチェーン・ネットワークに接続するように IBM MQ Bridge to blockchain を構成できます。ブリッジの実行中に、メッセージング・アプリケーションからブリッジに、リソースの

状態を照会または更新して結果を応答としてメッセージング・アプリケーションに返すチェーンコード・ルーチンを実行するように要求します。

## 手順

1. キュー・マネージャーを作成して開始するか、または IBM MQ Bridge to blockchain で使用する既存のキュー・マネージャーを開始します。

キュー・マネージャーの作成:

```
crtmqm adv_qmgr_name
```

キュー・マネージャーの開始:

```
strmqm adv_qmgr_name
```

2. **DefineQ.mqsc** スクリプトで定義されているブリッジ用のキューを作成します。

以下に使用される指定されたデフォルト・キューには、サンプル・ブリッジ・キュー定義が用意されています。

- ユーザー資格情報 (例: SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE)
- ブリッジへのメッセージ入力 (例: APPL1.BLOCKCHAIN.INPUT.QUEUE)
- ブロックチェーンからの応答 (例: APPL1.BLOCKCHAIN.REPLY.QUEUE)

/opt/mqm/mqbc/samp ディレクトリーから以下のコマンドを実行します。

```
runmqsc adv_qmgr_name < ./DefineQ.mqsc
```

アプリケーションごとに同じ入力キューを使用できますが、アプリケーションごとに1つずつ、複数の応答キューを指定することができます。定義済み応答キューを使用する必要はありません。応答に動的キューを使用する場合は、そのセキュリティー構成を考慮する必要があります。

## タスクの結果

ブリッジが IBM MQ およびブロックチェーン・ネットワークからのメッセージを処理するために必要なキューを作成しました。

## 次のタスク

ブロックチェーン・ネットワークからの IBM MQ Advanced キュー・マネージャー情報および資格情報を使用して、IBM MQ Bridge to blockchain の構成ファイルを作成します。

## V 9.0.4 IBM MQ Bridge to blockchain の構成ファイルの作成

キュー・マネージャーおよびブロックチェーン・ネットワークのパラメーターを入力して、IBM MQ Bridge to blockchain が IBM MQ および IBM Blockchain ネットワークに接続するための構成ファイルを作成します。

## 始める前に

- ブロックチェーン・ネットワークを作成し、構成しておきます。
- ブロックチェーン・ネットワークからの資格情報ファイルを入手します。
- x86Linux 環境に、IBM MQ Bridge to blockchain をインストールしました。
- IBM MQ Advanced キュー・マネージャーを開始しておきます。

## このタスクについて

このタスクでは、IBM MQ Bridge to blockchain 構成ファイルを作成し、正常に IBM Blockchain および IBM MQ ネットワークに接続するために必要な最小限のセットアップについて説明します。

ブリッジを使用して、Hyperledger Fabric 1.0 architecture を基盤にしたブロックチェーン・ネットワークに接続できます。ブリッジを使用するには、ブロックチェーン・ネットワークからの構成情報が必要です。このタスクの各ステップで、2つの異なる構成済みブロックチェーン・ネットワークに基づく構成の詳細の例を示します。

- Docker で実行される Hyperledger Fabric ネットワーク。詳細については、[Getting started with Hyperledger Fabric](#)、[Writing your first application](#)、および [649 ページの『Hyperledger Fabric ネットワーク資格情報ファイルの例』](#)を参照してください。
- IBM Cloud (formerly Bluemix)の Kubernetes クラスター内で稼働する Hyperledger Fabric ネットワーク。詳細については、[Develop in a cloud sandbox on IBM Blockchain Platform](#) および [651 ページの『Kubernetes コンテナ・クラスター・ネットワーク構成ファイルの例』](#)を参照してください。

すべての IBM MQ Bridge to blockchain パラメーターの意味とオプションの詳細については、[runmqbcb コマンド](#)を参照してください。独自のセキュリティー要件を考慮し、デプロイメントに適したパラメーターをカスタマイズする必要があります。

## 手順

1. ブリッジを実行して構成ファイルを作成します。

ブロックチェーン・ネットワークの資格情報ファイルおよび IBM MQ Advanced キュー・マネージャーのパラメーターが必要です。

```
runmqbcb -o config_file_name.cfg
```

以下の例に示すように、既存の値は大括弧に囲まれて表示されます。既存の値を受け入れる場合は、Enter キーを押します。値をクリアする場合は、Space キーを押してから Enter キーを押します。新しい値を追加する場合は、大括弧で囲んで入力してから Enter キーを押します。値(ピアなど)のリストをコマンドで区切ることも、新しい行にそれぞれの値を入力して区切ることもできます。ブランク行はリストの終わりを示します。

**注:** 既存の値は編集できません。それらを保持、置換、またはクリアすることはできます。

2. IBM MQ Advanced キュー・マネージャーへの接続の値を入力します。

接続に必要な最小値は、キュー・マネージャー名、ブリッジ入力の名前、およびユーザーが定義したアイデンティティー・キューの名前です。リモート・キュー・マネージャーに接続する場合は、**MQ Channel**と**MQ Conname**(キュー・マネージャーが実行されているホスト・アドレスとポート)も必要です。ステップ [648 ページの『6』](#)で IBM MQ に接続するために TLS を使用するには、JNDI または CCDT を使用し、それに応じて **MQ CCDT URL** または **JNDI implementation class** および **JNDI provider URL** を指定する必要があります。

```
Connection to Queue Manager
-----
Queue Manager                : [adv_qmgr_name]
Bridge Input Queue           : [APPL1.BLOCKCHAIN.INPUT.QUEUE]
Bridge User Identity Queue   : [SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE]
MQ Channel                   : []
MQ Conname                   : []
MQ CCDT URL                  : []
JNDI implementation class    : []
JNDI provider URL           : []
MQ Userid                   : []
MQ Password                  : []
```

3. ブロックチェーン・ネットワークの認証局のログインの詳細を入力します。

ローカル Hyperledger Fabric および Kubernetes クラスターの例のデフォルト値は、**Userid** が *admin*、**Enrollment Secret** が *adminpw* です。ご使用のブロックチェーン・ネットワーク用にこれらの値を変更する場合は、適切な値を使用してブリッジを構成していることを確認します。

```
Blockchain - User Identification
-----
```

```
Blockchain Userid          : []admin
Enrollment Secret        : []*****
```

4. ブロックチェーン・ネットワークのメンバーシップおよびアイデンティティ・ルールを管理するメンバーシップ・サービス・プロバイダー ID (MSPID) を入力します。

資格情報ファイルから、**Organisation Name** および **Organisation MSPID** の **msp\_id** パラメータを指定します。649 ページの『[Hyperledger Fabric ネットワーク資格情報ファイルの例](#)』から、ファイルの **peer** セクションの **CORE\_PEER\_LOCALMSPID** 値を使用します。651 ページの『[Kubernetes コンテナ・クラスター・ネットワーク構成ファイルの例](#)』から、**mSPID** 値を使用します。

```
Blockchain - Organisation Identification
-----
Organisation Name          : []Org1MSP
Organisation MSPID        : []Org1MSP
```

5. ブロックチェーン・ネットワーク・サーバーのロケーション値を入力します。  
649 ページの『[Hyperledger Fabric ネットワーク資格情報ファイルの例](#)』から、認証局、ピア、および順序付けプログラムの各エレメントの名前とサーバー: ポートの場所を指定します。

```
Blockchain server locations
-----
Certificate Authority servers : [ca.example.com Docker_container_host:7054] (for
example ca.example.com localhost:7054)
Peer servers                 : [peer0 localhost:7051]
Orderer servers              : [orderer0 localhost:7050]
Peer Event servers           : [peer0 localhost:7053]
Location of PEM file for Blockchain certificate : []
```

651 ページの『[Kubernetes コンテナ・クラスター・ネットワーク構成ファイルの例](#)』から、認証局、ピア、および順序付けプログラムの各エレメントの名前とサーバー: ポートの場所を指定します。

```
Blockchain server locations
-----
Certificate Authority servers : [CA1
your_blockchain_network_public_ip_address:30000] (for example CA1 123.456.789.10:30000)
Peer servers                 : [blockchain-org1peer1
your_blockchain_network_public_ip_address:30110]
Orderer servers              : [blockchain-orderer
your_blockchain_network_public_ip_address:31010]
Peer Event servers           : [blockchain-org1peer1
your_blockchain_network_public_ip address:30111]
Location of PEM file for Blockchain certificate : []
```

6. TLS 接続の証明書ストア値を入力します。

ブリッジは、キュー・マネージャーに接続している IBM MQ Java クライアントとして機能します。つまり、TLS セキュリティーを使用して、その他すべての IBM MQ Java クライアントと同様に安全に接続するようにブリッジを構成できます。TLS 接続の構成の詳細は、ステップ [647 ページの『2』](#) で JNDI または CCDT 情報を指定した後にのみ公開されます。

```
Certificate stores for TLS connections
-----
Personal keystore           : []
Keystore password           : []
Trusted store for signer certs : []
Trusted store password      : []
Use TLS for MQ connection   : [N]
Timeout for Blockchain operations : [12]
```

7. IBM MQ Bridge to blockchain のログ・ファイルの場所を入力します。

構成ファイルまたはコマンド行でログ・ファイルの名前と場所を指定する必要があります。

```
Behavior of bridge program
-----
Runtime logfile for copy of stdout/stderr : [/var/mqm/errors/runmqbcb.log]
Done.
```

## タスクの結果

IBM MQ Bridge to blockchain が IBM Blockchain ネットワークおよび IBM MQ Advanced キュー・マネージャーに接続するために使用する構成ファイルを作成しました。

## 次のタスク

651 ページの『[IBM MQ Bridge to blockchain の実行](#)』のステップに従います。

## Hyperledger Fabric ネットワーク資格情報ファイルの例

Docker で動作しているローカルにインスタンス化された Hyperledger Fabric ブロックチェーンネットワークの .yaml ファイルの内容で、IBM MQ Bridge to blockchain が使用することができます。

[Hyperledger Fabric の使用を開始するチュートリアル](#)を学習し、[で何が起きているのか](#)を理解し、[Hyperledger Fabric のサンプル](#)のいずれかを使用して、ネットワークを起動した後は、/blockchain/fabric-samples/basic-network フォルダー内に以下の構成ファイルをインストールしておく必要があります。

ブロックチェーン・ネットワークに接続する場合は、646 ページの『[IBM MQ Bridge to blockchain の構成ファイルの作成](#)』の際にこのファイルの構成詳細を使用する必要があります。

```
#
# Copyright IBM Corp All Rights Reserved
#
# SPDX-License-Identifier: Apache-2.0
#
version: '2'

networks:
  basic:

services:
  ca.example.com:
    image: hyperledger/fabric-ca
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca.example.com
    ports:
      - "7054:7054"
    command: sh -c 'fabric-ca-server start --ca.certfile /etc/hyperledger/fabric-ca-server-config/ca.org1.example.com-cert.pem --ca.keyfile /etc/hyperledger/fabric-ca-server-config/f329434b83a06f32f17a300fef841cfd16ff58f3185fb744aae047207b01a9e_sk -b admin:adminpw -d'
    volumes:
      - ./crypto-config/peerOrganizations/org1.example.com/ca/:/etc/hyperledger/fabric-ca-server-config
    container_name: ca.example.com
    networks:
      - basic

  orderer.example.com:
    container_name: orderer.example.com
    image: hyperledger/fabric-orderer
    environment:
      - ORDERER_GENERAL_LOGLEVEL=debug
      - ORDERER_GENERAL_LISTENADDRESS=0.0.0.0
      - ORDERER_GENERAL_GENESIMETHOD=file
      - ORDERER_GENERAL_GENESISFILE=/etc/hyperledger/configtx/genesis.block
      - ORDERER_GENERAL_LOCALMSPID=OrdererMSP
      - ORDERER_GENERAL_LOCALMSPDIR=/etc/hyperledger/msp/orderer/msp
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric/orderer
    command: orderer
    ports:
      - 7050:7050
    volumes:
      - ./config/:/etc/hyperledger/configtx
      - ./crypto-config/ordererOrganizations/example.com/orderers/orderer.example.com/:/etc/hyperledger/msp/orderer
      - ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/:/etc/hyperledger/msp/peerOrg1
    networks:
      - basic
```

```

peer0.org1.example.com:
  container_name: peer0.org1.example.com
  image: hyperledger/fabric-peer
  environment:
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    - CORE_PEER_ID=peer0.org1.example.com
    - CORE_LOGGING_PEER=debug
    - CORE_CHAINCODE_LOGGING_LEVEL=DEBUG
    - CORE_PEER_LOCALMSPID=Org1MSP
    - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/msp/peer/
    - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
    # # the following setting starts chaincode containers on the same
    # # bridge network as the peers
    # # https://docs.docker.com/compose/networking/
    - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=${COMPOSE_PROJECT_NAME}_basic
    - CORE_LEDGER_STATE_STATEDATABASE=CouchDB
    - CORE_LEDGER_STATE_COUCHDBCONFIG_COUCHDBADDRESS=couchdb:5984
    # The CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME and
  CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD
    # provide the credentials for ledger to connect to CouchDB. The username and password
  must
    # match the username and password set for the associated CouchDB.
    - CORE_LEDGER_STATE_COUCHDBCONFIG_USERNAME=
    - CORE_LEDGER_STATE_COUCHDBCONFIG_PASSWORD=
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric
  command: peer node start
  # command: peer node start --peer-chaincodedev=true
  ports:
    - 7051:7051
    - 7053:7053
  volumes:
    - /var/run/:/host/var/run/
    - ./crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/
msp:/etc/hyperledger/msp/peer
    - ./crypto-config/peerOrganizations/org1.example.com/users:/etc/hyperledger/msp/users
    - ./config:/etc/hyperledger/configtx
  depends_on:
    - orderer.example.com
    - couchdb
  networks:
    - basic

couchdb:
  container_name: couchdb
  image: hyperledger/fabric-couchdb
  # Populate the COUCHDB_USER and COUCHDB_PASSWORD to set an admin user and password
  # for CouchDB. This will prevent CouchDB from operating in an "Admin Party" mode.
  environment:
    - COUCHDB_USER=
    - COUCHDB_PASSWORD=
  ports:
    - 5984:5984
  networks:
    - basic

cli:
  container_name: cli
  image: hyperledger/fabric-tools
  tty: true
  environment:
    - GOPATH=/opt/gopath
    - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
    - CORE_LOGGING_LEVEL=DEBUG
    - CORE_PEER_ID=cli
    - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
    - CORE_PEER_LOCALMSPID=Org1MSP
  CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp
    - CORE_CHAINCODE_KEEPALIVE=10
  working_dir: /opt/gopath/src/github.com/hyperledger/fabric/peer
  command: /bin/bash
  volumes:
    - /var/run/:/host/var/run/
    - ../chaincode:/opt/gopath/src/github.com/
    - ./crypto-config:/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/
  networks:
    - basic
  #depends_on:
  # - orderer.example.com

```

```
# - peer0.org1.example.com
# - couchdb
```

## Kubernetes コンテナ・クラスター・ネットワーク構成ファイルの例

IBM Cloud (formerly Bluemix) の Kubernetes クラスターで実行される Hyperledger Fabric ブロックチェーン・ネットワークからの構成ファイルの内容。IBM MQ Bridge to blockchain を構成するために使用できません。

[IBM Blockchain Prepare and setup](#)、[Simple install](#)、[Interacting with your blockchain](#) の各チュートリアルを終了した後に、接続プロファイル・フォルダーに JSON ファイルがある必要があります。

ブロックチェーン・ネットワークに接続する場合は、[646 ページの『IBM MQ Bridge to blockchain の構成ファイルの作成』](#)の際にこのファイルの構成詳細を使用する必要があります。

```
{
  "name": "ibm-bc-org1",
  "description": "Connection profile for IBM Blockchain Platform",
  "type": "hlfv1",
  "orderers": [
    {
      "url": "grpc://INSERT_PUBLIC_IP:31010"
    }
  ],
  "ca": {
    "url": "http://INSERT_PUBLIC_IP:30000",
    "name": "CA1"
  },
  "peers": [
    {
      "requestURL": "grpc://INSERT_PUBLIC_IP:30110",
      "eventURL": "grpc://INSERT_PUBLIC_IP:30111"
    }
  ],
  "keyValStore": "INSERT_CREDENTIALS_PATH",
  "channel": "channel1",
  "mspID": "Org1MSP",
  "timeout": 300
}
```

### V 9.0.4 IBM MQ Bridge to blockchain の実行

IBM MQ Bridge to blockchain を実行して、IBM Blockchain と IBM MQ に接続します。接続すると、ブリッジが、照会を処理し、メッセージを更新してブロックチェーン・ネットワークに送信し、応答を受信および処理できるようになります。

#### このタスクについて

前のステップで作成した構成ファイルを使用して、IBM MQ Bridge to blockchain を実行します。

#### 手順

1. ブリッジで使用する IBM MQ Advanced キュー・マネージャーを始動します。
2. IBM MQ Bridge to blockchain を開始して、ブロックチェーン・ネットワークと IBM MQ Advanced キュー・マネージャーに接続します。  
ブリッジ・コマンドを実行します。

```
runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
```

ブリッジが接続されると、以下のような出力が返されます。

```
Fri Oct 06 06:32:11 PDT 2017 IBM MQ Bridge to Blockchain
5724-H72 (C) Copyright IBM Corp. 2017, 2023.
```

Fri Oct 06 06:32:17 PDT 2017 Ready to process input messages.

3. オプション: ブリッジの実行後に返されたメッセージが、接続が成功しなかったことを示している場合、IBM MQ Advanced キュー・マネージャーおよびブロックチェーン・ネットワークへの接続をトラブルシューティングします。

- a) デバッグ・オプション 1 を指定して、デバッグ・モードでコマンドを発行します。

```
runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log -d 1
```

ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが簡潔モードで表示されます。

- b) デバッグ・オプション 2 を指定して、デバッグ・モードでコマンドを発行します。

```
runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log -d 2
```

ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが冗長モードで表示されます。全出力がログ・ファイルに書き込まれます。

## タスクの結果

IBM MQ Bridge to blockchain が開始され、キュー・マネージャーとブロックチェーン・ネットワークに接続されます。

## 次のタスク

- 797 ページの『[IBM MQ Bridge to blockchain クライアント・サンプルの実行](#)』のステップに従って、照会または更新メッセージをフォーマット設定し、ブロックチェーン・ネットワークに送信します。
- `MQBCB_EXTRA_JAVA_OPTIONS` 変数を使用して、例えば、IBM MQ トレースを有効化する JVM プロパティを渡します。詳しくは、[IBM MQ Bridge to blockchain のトレース](#)を参照してください。

## V 9.0.4 IBM MQ Bridge to blockchain のメッセージの形式

IBM MQ Bridge to blockchain によって送受信されるメッセージのフォーマット設定に関する情報。

アプリケーションは、IBM MQ Bridge to blockchain が照会を実行するか、またはブロックチェーンで保持されている情報を更新することを要求します。アプリケーションでは、ブリッジ要求キューに要求メッセージを配置することによって、これを行います。照会または更新の結果は、ブリッジによって応答メッセージの形式に設定されます。ブリッジでは、応答メッセージの宛先として、要求メッセージの `MQMD` の `ReplyToQ` および `ReplyToQMGR` フィールドに格納されている情報を使用します。

ブリッジによって消費または生成されるメッセージは、JSON 形式のテキスト (MQSTR) メッセージです。入力メッセージは単純な JSON であり、プログラムでは文字列の連結を使用して、入力メッセージを生成できます。 `args` 以外のすべてのフィールドが必須です。このフィールドの引数リストには、保管されているチェーンコードの関数の知識が必要です。

## 要求メッセージの形式

入力メッセージの形式:

```
{ "function": functionName,
  "channel" : chainName,
  "chaincodeName" : codeName,
  "args" : [argument list]
}
```

機能する `Fabcar` サンプルを含むローカルの Hyperledger ネットワーク例の場合。

- ブロックチェーンに保持されている車両の詳細を表す JSON オブジェクトのリストを返す fabcar チェーンコードの queryAllCars 関数を呼び出す照会メッセージを使用するには、メッセージを以下のようフォーマット設定します。

```
{ "function": "queryAllCars",
  "channel": "mychannel",
  "chaincodeName": "fabcar",
  "args": []
}
```

応答例:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": [
    {"Record": {"owner": "Tomoko", "colour": "blue", "model": "Prius", "make": "Toyota"}, "Key": "CAR0"},
    {"Record": {"owner": "Brad", "colour": "red", "model": "Mustang", "make": "Ford"}, "Key": "CAR1"},
    {"Record": {"owner": "Jin Soo", "colour": "green", "model": "Tucson", "make": "Hyundai"}, "Key": "CAR2"},
    {"Record": {"owner": "Max", "colour": "yellow", "model": "Passat", "make": "Volkswagen"}, "Key": "CAR3"},
    {"Record": {"owner": "Adriana", "colour": "black", "model": "S", "make": "Tesla"}, "Key": "CAR4"},
    {"Record": {"owner": "Michel", "colour": "purple", "model": "205", "make": "Peugeot"}, "Key": "CAR5"},
    {"Record": {"owner": "Aarav", "colour": "white", "model": "S22L", "make": "Chery"}, "Key": "CAR6"},
    {"Record": {"owner": "Pari", "colour": "violet", "model": "Punto", "make": "Fiat"}, "Key": "CAR7"},
    {"Record": {"owner": "Valeria", "colour": "indigo", "model": "Nano", "make": "Tata"}, "Key": "CAR8"},
    {"Record": {"owner": "Shotaro", "colour": "brown", "model": "Barina", "make": "Holden"}, "Key": "CAR9"}
  ]
}
```

応答メッセージには、現在ブロックチェーンに保持されているすべての車両レコードが含まれます。

- ブロックチェーン台帳で新しい車両項目を作成する fabcar チェーンコード例の createCar 関数を呼び出す更新メッセージを使用するには、メッセージを以下のようフォーマット設定します。

```
{ "function": "createCar",
  "channel": "mychannel",
  "chaincodeName": "fabcar",
  "args": ["CAR10", "Ford", "Mustang GT", "Blue", "Bob"]
}
```

応答例:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
```

ブロックチェーンに新しい車両項目が追加されたことを確認するには、すべての車両を返す初期メッセージを再度使用します。

機能する [example02](#) デモを含む Kubernetes クラスター・ネットワーク例の場合。

- ブロックチェーン台帳内のエンティティ "a" の値を返す example02 チェーンコードの query 関数を呼び出す照会メッセージを使用するには、メッセージを以下のようフォーマット設定します。

```
{ "function": "query",
  "channel": "channel1",
  "chaincodeName": "example02",
  "args": ["a"]
}
```

応答例:

```
{
  "statusCode": 200,
```

```
"statusType": "SUCCESS",
"message": "OK",
"data": "100"
}
```

- 最初の引数に指定されるエンティティを減少させ、2番目の引数に指定されているエンティティを、3番目の引数に指定されている値だけ増分させる `invoke` 関数の `example02` チェーンコードを呼び出すメッセージを使用するには、メッセージを以下のようにフォーマット設定します。

```
{ "function": "invoke",
  "channel": "channel1",
  "chaincodeName": "example02",
  "args": ["a", "b", "10"]
}
```

値は以下のとおりです。

– 前: a=100、b=200

– 後: a=90、b=210

応答例:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
```

新しい値を検査するには、**"a"** および **"b"** の値を検索する新規メッセージ照会メッセージを実行依頼します。

## 応答メッセージの形式

応答メッセージでは、その関連 ID がインバウンド・メッセージのメッセージ ID に設定されています。ユーザー定義のプロパティが入力メッセージから出力メッセージにコピーされます。応答のユーザー ID は、`set-identity` コンテキストによって発信元のユーザー ID に設定されます。

正常な処理の例:

```
{ "data": "500", "message": "OK", "statusCode": 200, "statusType": "SUCCESS" }
```

このメッセージの応答データは、チェーンコード応答によって生成されたものです (UTF-8 文字列に変換されるバイト数)。

すべてのエラー応答には、それがブリッジ自体によって生成されたか、またはブロックチェーンの呼び出しやチェーンコード呼び出しから生成されたかにかかわらず、同じフィールドがあります。以下に例を示します。

- 誤ったチャンネル名

```
{
  "message": "Bad newest block expected status 200 got 404, Chain myUnknownChannel",
  "statusCode": 404,
  "statusType": "FAILURE"
}
```

- 誤った JSON 入力メッセージ

```
{
  "message": "Error: Cannot parse message contents.",
  "statusCode": 2110,
  "statusType": "FAILURE"
}
```

- チェーンコードに対する誤ったパラメーター

```
{
  "message": "Sending proposal to fabric-peer-1a failed because of gRPC
failure=Status{code=UNKNOWN, description={\"Error\": \"Nil amount for c\"}, cause=null}",
  "statusCode": 500,
  "statusType": "FAILURE"
}
```

アプリケーションでは、**statusType** 文字列またはデータ・フィールドの存在を確認することによって、要求が成功したか失敗したかがわかります。入力メッセージの処理でエラーが発生し、ブリッジがブロックチェーンに送信しない場合、ブリッジから返される値はMQRC 値で、通常、**MQRC\_FORMAT\_ERROR** です。

## V 9.0.4 IBM MQ Bridge to blockchain クライアント・サンプルの実行

IBM MQ Bridge to blockchain に用意されている JMS クライアント・サンプルを使用して、Blockchain Bridge が検査している入力キューにメッセージを書き込み、受け取った応答を表示できます。

### 始める前に

IBM MQ Bridge to blockchain が実行中で、IBM MQ Advanced キュー・マネージャーおよびブロックチェーン・ネットワークに接続されており、入力メッセージを処理する準備ができています。

### このタスクについて

IBM MQ Bridge to blockchain の samp ディレクトリーで JMS サンプル・アプリケーションを見つけます。

### 手順

1. クライアント・サンプルの Java ソース・ファイルを編集します。

サンプル内の指示に従って、ご使用の IBM MQ 環境およびブロックチェーン・ネットワークに合わせて構成します。サンプルの以下のコードは、ブリッジに送信する JSON 要求メッセージを定義しています。

```
// Create the JSON request message.
// Modify "query", "exampleBlockchainChannelName", and "exampleChaincodeName" to
// match your deployed blockchain chaincode.
// The "operation" field is optional, but recommended. It should be set to QUERY
// or UPDATE to match what the chaincode is going to do.

JSONObject inputMsg = new JSONObject();
inputMsg.put("operation", "QUERY");

inputMsg.put("function", "query");
inputMsg.put("channel", "exampleBlockchainChannelName");
inputMsg.put("chaincodeName", "exampleChaincodeName");

// Create the JSON arguments for the request message.
// Modify "a" to match your deployed blockchain chaincode
// requirements, and add further arguments as necessary

JSONArray myArgs = new JSONArray();
myArgs.add("a");
inputMsg.put("args", myArgs);

TextMessage message = session.createTextMessage(inputMsg.serialize());
message.setJMSReplyTo(replyToQueue);
```

2. サンプルをコンパイルします。

ブリッジ・ディレクトリーに出荷される IBM MQ クライアント・クラスおよび JSON4j.jar ファイルを指します。

```
javac -cp $MQ_JAVA_INSTALL_PATH/lib/*:../prereqs/JSON4J.jar SimpleBCBClient.java
```

### 3. コンパイルされたクラスを実行します。

```
java -cp $MQ_JAVA_INSTALL_PATH/lib/*:../prereqs/JSON4J.jar:. SimpleBCBClient
```

```
Starting Simple MQ Blockchain Bridge Client
Created the message. Starting the connection
Sent message:

JMSMessage class: jms_text
JMSType: null
JMSDeliveryMode: 2
JMSDeliveryDelay: 0
JMSDeliveryTime: 1508427559117
JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:414d5120424342514d202020202020209063e859ea36aa24
JMSTimestamp: 1508427559117
JMSCorrelationID: null
JMSDestination: queue:///APPL1.BLOCKCHAIN.INPUT.QUEUE
JMSReplyTo: queue:///APPL1.BLOCKCHAIN.REPLY.QUEUE
JMSRedelivered: false
  JMSXAppID: java
  JMSXDeliveryCount: 0
  JMSXUserID: USER1
  JMS_IBM_PutApplType: 6
  JMS_IBM_PutDate: 20171019
  JMS_IBM_PutTime: 15391912
{"args":
["a"],"function":"query","channel":"exampleBlockchainChannelName","operation":"QUERY","chaincodeName":"exampleChaincodeName"}
```

応答メッセージ:

```
JMSMessage class: jms_text
JMSType: null
JMSDeliveryMode: 1
JMSDeliveryDelay: 0
JMSDeliveryTime: 0
JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:c3e2d840e2e2f0f840404040404040d2afa27229838af2
JMSTimestamp: 1497439784000
JMSCorrelationID: ID:414d5120424342514d202020202020209063e859ea36aa24 *(JMSMessageID of
the input message)
JMSDestination: null
JMSReplyTo: null
JMSRedelivered: false
  JMSXAppID: java
  JMSXDeliveryCount: 1
  JMSXUserID: USER1
  JMS_IBM_Character_Set: UTF-8
  JMS_IBM_Encoding: 273
  JMS_IBM_Format: MQSTR
  JMS_IBM_MsgType: 8
  JMS_IBM_PutApplType: 2
  JMS_IBM_PutDate: 20171019
  JMS_IBM_PutTime: 15392014
{
  "data": "20",
  "message": "OK",
  "statusCode": 200,
  "statusType": "SUCCESS"
}
Response text:
{
  "data": "20",
  "message": "OK",
  "statusCode": 200,
  "statusType": "SUCCESS"
}
SUCCESS
```

応答の待機中にクライアントがタイムアウト・エラーを受け取った場合は、ブリッジが実行されていることを確認します。

IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

## 始める前に

IBM MQ を設定する前に、[IBM MQ for z/OS](#) で IBM MQ for z/OS の概念についてお読みください。

[z/OS での IBM MQ 環境の計画](#)での IBM MQ for z/OS 環境の計画方法について説明します。

## このタスクについて

IBM MQ をインストールした後、それを使用できるようにするためには、多くの作業を行う必要があります。

## 手順

- IBM MQ for z/OS でキュー・マネージャーを構成する方法については、以下のサブトピックを参照してください。

### 関連タスク

5 ページの『[マルチプラットフォームでのキュー・マネージャーの作成と管理](#)』

メッセージおよびキューを使用するには、事前に少なくとも 1 つのキュー・マネージャーとそれに関連するオブジェクトを作成し、開始しておく必要があります。キュー・マネージャーは、関連したリソース、特にキュー・マネージャー自体が所有しているキューを管理します。キュー・マネージャーは、メッセージ・キューイング・インターフェース (MQI) 呼び出しおよびコマンドで IBM MQ オブジェクトを作成、修正、表示、および削除できるように、アプリケーションに対してキューイング・サービスを提供します。

140 ページの『[分散キューイングの構成](#)』

このセクションでは、キュー定義、チャンネル定義、トリガー操作および同期点プロシーチャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

16 ページの『[サーバーとクライアント間の接続の構成](#)』

IBM MQ MQI clients とサーバーの間の通信リンクを構成するには、通信プロトコルを決定し、リンクの両端の接続を定義し、リスナーを開始し、チャンネルを定義します。

### 関連情報

[IBM MQ for z/OS の概念](#)

[セキュリティ](#)

[管理 IBM MQ for z/OS](#)

[計画](#)

[コマンドの実行](#)

[IBM MQ for z/OS ユーティリティー](#)

## z/OS でのキュー・マネージャーのカスタマイズ準備

このトピックでは、インストール可能な機能に関する詳細、各国語機能、およびテストに関する詳細を使用してキュー・マネージャーをカスタマイズし、セキュリティをセットアップすることについて説明しています。

## カスタマイズの準備

Program Directory は、IBM MQ インストール・テープの内容のリスト、IBM MQ のプログラムとサービス・レベル情報、およびシステム修正変更プログラム拡張版 (SMP/E) を使用して IBM MQ for z/OS をインストールする方法を示します。IBM MQ for z/OS のプログラム・ディレクトリーは、[IBM Publications Center](#) からダウンロードできます ([IBM MQ 9.0 PDF 資料](#)を参照してください)。

IBM MQ をインストールした後、それを使用できるようにするためには、多くの作業を行う必要があります。それらの作業の説明については、以下の項を参照してください。

- [661 ページの『IBM MQ for z/OS のセットアップ』](#)
- [719 ページの『z/OS でのキュー・マネージャーのテスト』](#)
- [z/OS でのセキュリティーのセットアップ](#)

旧バージョンの IBM MQ for z/OS から移行する場合、大部分のカスタマイズ・タスクについては実行する必要がありません。実行しなければならないタスクの詳細については、[保守とマイグレーション](#)を参照してください。

## IBM MQ for z/OS のインストール可能な機能

IBM MQ for z/OS には、以下の機能が含まれます。

### 基本機能

この機能は必須です。これには、以下の機能を含む、主な機能のすべてが含まれています。

- 管理およびユーティリティー
- CICS、IMS、および IBM MQ アプリケーション・プログラミング・インターフェースまたは C++ を使用するバッチ・タイプ・アプリケーションのサポート
- 分散キューイング機能 (TCP/IP および APPC 通信の両方をサポート)

### 各国語機能

エラー・メッセージとパネルは、サポートされるすべての各国語で表現されています。各国語は言語を示す文字で表します。各国語と言語を示す文字の対応は、次のとおりです。

#### C

中国語 (簡体字)

#### E

U.S. 英語 (大/小文字混合)

#### F

フランス語

#### K

日本語

#### U

U.S. 英語 (大文字)

米国英語 (大/小文字混合) オプションをインストールする必要があります。また、1つ以上の他の言語もインストールできます。(他の言語のインストール・プロセスには、米国英語 (大/小文字混合) を使用しない場合であっても、米国英語 (大/小文字混合) がインストールされている必要があります。)

## IBM MQ for z/OS Unix システム・サービス・コンポーネント

このフィーチャーはオプションです。Java Message Service (JMS) を使用して IBM MQ for z/OS に接続する Java アプリケーションをビルドして実行する場合、または HTTP を使用して IBM MQ for z/OS に接続する HTTP アプリケーションをビルドして実行する場合は、この機能を選択します。

### **V 9.0.1** IBM MQ for z/OS Unix System Services Web Components

このフィーチャーはオプションです。

IBM MQ Console または REST API を使用する場合は、このフィーチャーを選択します。

このフィーチャーをインストールするには、IBM MQ for z/OS Unix System Services Components フィーチャーをインストールする必要があります。

## インストール後に存在するライブラリー

IBM MQ には、複数のロード・ライブラリーが添付されています。[659 ページの表 38](#) は、IBM MQ をインストールした後に存在する可能性があるライブラリーを示しています。

表 38. インストール後に存在する IBM MQ ライブラリー	
名前	説明
thlqual.SCSQANLC	IBM MQ の中国語 (簡体字) 版のロード・モジュールが入っています。
thlqual.SCSQANLE	米国のロード・モジュールを含む。英語(大/小文字混合)の IBM MQ バージョン。
thlqual.SCSQANLF	IBM MQ のフランス語版のロード・モジュールが入っています。
thlqual.SCSQANLK	IBM MQ の日本語版のロード・モジュールが入っています。
thlqual.SCSQANLU	米国のロード・モジュールを含む。IBM MQ の英語(大文字)のバージョン。
thlqual.SCSQASMS	アセンブラー・サンプル・プログラムのソースが入っています。
thlqual.SCSQAUTH	すべての IBM MQ 製品ロード・モジュールのメイン・リポジトリ。これには、デフォルト・パラメーター・モジュール CSQZPARM も含まれます。このライブラリーは、APF 許可が必要で、PDS-E 形式でなければなりません。
thlqual.SCSQCICS	CICS DFHRPL 連結に入れる必要のある追加のロード・モジュールが入っています。このライブラリーは、APF 許可が必要で、PDS-E 形式でなければなりません。
thlqual.SCSQCLST	サンプル・プログラムで使用する CLIST が入っています。
thlqual.SCSQCOBC	サンプル・プログラムに必要なサンプル集などの COBOL サンプル集が入っています。
thlqual.SCSQCOBS	COBOL サンプル・プログラムのソースが入っています。
thlqual.SCSQCPPS	C++ サンプル・プログラムのソースが入っています。
thlqual.SCSQC37S	C サンプル・プログラムのソースが入っています。
thlqual.SCSQC370	サンプル・プログラムに必要なヘッダーを含む、C ヘッダーが入っています。
thlqual.SCSQDEFS	共有キューイング用の C++ および Db2 DBRM のサイド定義が入っています。
thlqual.SCSQEXEC	IBM MQ 操作と操作パネルを使用している場合に SYSEXEC または SYSPROC 連結に組み込む REXX 実行可能ファイルが入っています。
thlqual.SCSQHPPS	C++ 用のヘッダー・ファイルが入っています。
thlqual.SCSQINST	インストール・ジョブ用の JCL が入っています。
thlqual.SCSQLINK	早期コード・ライブラリー。システム初期プログラム・ロード (IPL) 時にロードされるロード・モジュールが入っています。ライブラリーは、APF 許可が必要です。
thlqual.SCSQLOAD	ロード・ライブラリー。APF 以外のコード、ユーザー出口、ユーティリティ、サンプル、インストール検査プログラム、およびアダプター・スタブのためのロード・モジュールが入っています。このライブラリーは、APF 許可の必要はなく、リンク・リスト内にある必要もありません。このライブラリーは、PDS-E 形式でなければなりません。
thlqual.SCSQMACS	サンプル・マクロ、製品マクロ、およびシステム・パラメーター・マクロを含むアセンブラー・マクロが入っています。

表 38. インストール後に存在する IBM MQ ライブラリー (続き)	
名前	説明
thlqual.SCSQMAPS	サンプル・プログラムに使用する CICS マップ・セットが入っています。
thlqual.SCSQMSGC	IBM MQ 操作パネルと制御パネルに中国語 (簡体字) 機能を使用している場合に ISPMLIB 連結に組み込む ISPF メッセージが入っています。
thlqual.SCSQMSGE	米国を使用している場合は、ISPMLIB 連結に組み込まれる ISPF メッセージを含む IBM MQ 操作および制御パネルの英語(大/小文字混合)の言語機能。
thlqual.SCSQMSGF	IBM MQ 操作パネルと制御パネルにフランス語機能を使用している場合に ISPMLIB 連結に組み込む ISPF メッセージが入っています。
thlqual.SCSQMSGK	IBM MQ 操作パネルと制御パネルに日本語機能を使用している場合に ISPMLIB 連結に組み込む ISPF メッセージが入っています。
thlqual.SCSQMSGU	米国を使用している場合は、ISPMLIB 連結に組み込まれる ISPF メッセージを含む IBM MQ 操作および制御パネルの英語(大文字)の言語機能。
thlqual.SCSQMVR1	分散キューイングのロード・モジュールが入っています。このライブラリーは、APF 許可が必要で、PDS-E 形式でなければなりません。
thlqual.SCSQPLIC	PL/I 組み込みファイルが入っています。
thlqual.SCSQPLIS	PL/I サンプル・プログラムのソースが入っています。
thlqual.SCSQPNLA	ISPPLIB 連結に組み込む、ダンプ・フォーマッター用の IPCS パネルが入っています。また、IBM MQ サンプル・プログラム用のパネルも入っています。
thlqual.SCSQPNLC	IBM MQ 操作パネルと制御パネルに中国語 (簡体字) 機能を使用している場合に ISPPLIB 連結に組み込む ISPF パネルが入っています。
thlqual.SCSQPNLE	米国を使用している場合、ISPPLIB 連結に組み込まれる ISPF パネルが入っています。IBM MQ 操作および制御パネルの英語(大/小文字混合)の言語機能。
thlqual.SCSQPNLF	IBM MQ 操作パネルと制御パネルにフランス語機能を使用している場合に ISPPLIB 連結に組み込む ISPF パネルが入っています。
thlqual.SCSQPNLK	IBM MQ 操作パネルと制御パネルに日本語機能を使用している場合に ISPPLIB 連結に組み込む ISPF パネルが入っています。
thlqual.SCSQPNLU	米国を使用している場合、ISPPLIB 連結に組み込まれる ISPF パネルが入っています。IBM MQ 操作および制御パネルの英語(大文字)の言語機能。
thlqual.SCSQPROC	サンプル JCL とデフォルトのシステム初期設定データ・セットが入っています。
thlqual.SCSQSNLC	IBM MQ モジュールの中国語 (簡体字) 版の専用機能 (例えば、早期コード)に必要なロード・モジュールが入っています。
thlqual.SCSQSNLE	米国のロード・モジュールを含む。特殊な目的関数(例えば、早期コード)に必要な IBM MQ モジュールの英語(大/小文字混合)のバージョン。

表 38. インストール後に存在する IBM MQ ライブラリー (続き)	
名前	説明
thlqual.SCSQSNLF	IBM MQ モジュールのフランス語版の専用機能 (例えば、早期コード) に必要なロード・モジュールが入っています。
thlqual.SCSQSNLK	IBM MQ モジュールの日本語版の専用機能 (例えば、早期コード) に必要なロード・モジュールが入っています。
thlqual.SCSQSNLU	米国のロード・モジュールを含む。特殊目的関数(例えば、初期コード)に必要な IBM MQ モジュールの英語(大文字)のバージョン。
thlqual.SCSQTBLK	IBM MQ 操作パネルと制御パネルに中国語 (簡体字) 機能を使用している場合に ISPTLIB 連結に組み込む ISPF 表が入っています。
thlqual.SCSQTBLE	米国を使用している場合は、ISPTLIB 連結に組み込まれる ISPF テーブルが入っています IBM MQ 操作および制御パネルの英語(大/小文字混合)の言語機能。
thlqual.SCSQTBFL	IBM MQ 操作パネルと制御パネルにフランス語機能を使用している場合に ISPTLIB 連結に組み込む ISPF 表が入っています。
thlqual.SCSQTBLL	IBM MQ 操作パネルと制御パネルに日本語機能を使用している場合に ISPTLIB 連結に組み込む ISPF 表が入っています。
thlqual.SCSQTBUL	米国を使用している場合は、ISPTLIB 連結に組み込まれる ISPF テーブルが入っています IBM MQ 操作および制御パネルの英語(大文字)の言語機能。

注: これらのライブラリーの変更、またはカスタマイズをしないでください。変更する場合はライブラリーをコピーし、そのコピーで変更を行ってください。

#### 関連概念

##### [728 ページの『他のキュー・マネージャーとの通信のセットアップ』](#)

このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

##### [759 ページの『IMS での IBM MQ の使用』](#)

IBM MQ-IMS アダプターおよび IBM MQ-IMS ブリッジは、IBM MQ が IMS と相互作用できるようにする 2 つのコンポーネントです。

##### [767 ページの『CICS での IBM MQ の使用』](#)

CICS で IBM MQ を使用するには、IBM MQ CICS アダプターおよびオプションで IBM MQ CICS bridge コンポーネントを構成する必要があります。

##### [770 ページの『IMS における OTMA 出口の使用』](#)

このトピックは、IBM MQ for z/OS で IMS Open Transaction Manager Access 出口を使用する場合に使用します。

#### 関連資料

##### [768 ページの『言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用』](#)

必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

#### 関連情報

[IBM MQ for z/OS の概念](#)

[管理 IBM MQ for z/OS](#)

## ▶ z/OS IBM MQ for z/OS のセットアップ

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

キュー・マネージャーを構成する最善の方法は、以下のステップをこの順序で実行することです。

1. 基本キュー・マネージャーを構成します。
2. キュー・マネージャーからキュー・マネージャーへの通信、およびリモート・クライアント・アプリケーション通信を実行するチャンネル・イニシエーターを構成します。
3. メッセージを暗号化または保護する場合は、Advanced Message Security for z/OS を構成します。
4. IBM MQ を使用してファイルを転送する場合は、Managed File Transfer for z/OS を構成します。
5. 管理用またはメッセージング用の REST API、あるいは MQ Console を使用して Web ブラウザーから IBM MQ を管理する場合は、mqweb サーバーを構成します。

このトピックでは、IBM MQ を正常にインストールした後のセットアップのさまざまな段階を、順を追って示します。インストール・プロセスは「プログラム・ディレクトリー」で説明されています。IBM MQ for z/OS のプログラム・ディレクトリーは、[IBM Publications Center](#) からダウンロードできます ([IBM MQ 9.0 PDF 資料](#)を参照してください)。

IBM MQ では、カスタマイズに役立つようにサンプルが提供されています。サンプル・データ・セットのメンバーは、CSQ4 の 4 文字で始まる名前になっています。また、このメンバーは thlqual.SCSQPROC ライブラリーに格納されています。

このトピックに記載したカスタマイズ作業を実行する前に、さまざまな構成オプションについて検討する必要があります。これらのオプションは、IBM MQ for z/OS のパフォーマンスおよびリソース要件に影響します。例えば、使用するグローバルバージョン・ライブラリーを決定する必要があります。

カスタマイズ・ステップの一部を自動化する場合は、774 ページの『[IBM z/OSMF を使用した IBM MQ の自動化](#)』を参照してください。

## 構成オプション

これらのオプションについての詳細は、[z/OS での計画](#)を参照してください。

このセクションの各作業の説明で、以下の点を示しています。

- 作業が IBM MQ のセットアップ・プロセスの一部であるか。つまり、z/OS システム上での IBM MQ のカスタマイズ時に作業を 1 回実行するかどうかということです。(並列シスプレックスでは、シスプレックス内のそれぞれの z/OS システムごとに作業を実行し、それぞれの z/OS システムがまったく等しくセットアップされていることを確認する必要があります。)
- 作業がキュー・マネージャーの追加作業の一部であるか。つまり、キュー・マネージャーの追加時にキュー・マネージャーごとに作業を 1 回実行するかどうかということです。

各種 z/OS システム・パラメーターを変更するコマンドを使用し、674 ページの『[SYS1.PARMLIB メンバーを更新する](#)』を推奨どおりに実行するなら、どの作業でも z/OS システムの IPL を実行する必要はありません。

操作を単純化し、問題判別を支援するために、シスプレックス内の z/OS システムをすべて、まったく等しくセットアップし、緊急の際に、どのシステムでも迅速にキュー・マネージャーを作成できるようにします。

メンテナンスを容易にするために、ご使用の IBM MQ ライブラリーを参照する別名を定義することを検討してください。詳しくは、[IBM MQ ライブラリーを参照する別名の使用](#)を参照してください。

### 関連概念

[728 ページの『他のキュー・マネージャーとの通信のセットアップ』](#)

このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

[759 ページの『IMS での IBM MQ の使用』](#)

IBM MQ-IMS アダプターおよび IBM MQ-IMS ブリッジは、IBM MQ が IMS と相互作用できるようにする 2 つのコンポーネントです。

[767 ページの『CICS での IBM MQ の使用』](#)

CICS で IBM MQ を使用するには、IBM MQ CICS アダプターおよびオプションで IBM MQ CICS bridge コンポーネントを構成する必要があります。

## 770 ページの『IMS における OTMA 出口の使用』

このトピックは、IBM MQ for z/OS で IMS Open Transaction Manager Access 出口を使用する場合に使用します。

### 関連資料

#### 768 ページの『言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用』

必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

### 関連情報

[IBM MQ for z/OS の概念](#)

[IBM MQ for z/OS の管理](#)

[IBM MQ for z/OS のプログラム・ディレクトリー](#)

## z/OS IBM MQ を使用するための z/OS システムの構成

これらのトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

### z/OS z/OS システム・パラメーターを識別する

作業の一部には z/OS システム・パラメーターの更新が含まれます。システム IPL が実行されたとき、どのシステム・パラメーターが指定されたかを知っておく必要があります。

- この作業は、IBM MQ を実行するそれぞれの z/OS システムごとに実行する必要があります。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

SYS1.PARMLIB (IEASYSpp) には、SYS1.PARMLIB の他のメンバーを指すパラメーターのリストが入っています (ここで、pp は、システムの IPL を実行するために使用された z/OS システム・パラメーター・リストを表します)。

検索する必要がある項目は、次のとおりです。

#### 663 ページの『IBM MQ ロード・ライブラリーの APF 許可を行う』の場合:

PROG=xx または APF=aa は許可プログラム機能 (APF) で許可されたライブラリー・リストを指します (メンバー PROGxx または IEFAPFaa)。

#### 664 ページの『z/OS リンク・リストおよび LPA を更新する』の場合:

LNK=kk はリンク・リストを指します (メンバー LNKLSTkk)。LPA=mm は LPA リストを指します (メンバー LPALSTmm)。

#### 666 ページの『z/OS プログラム特性表を更新する』の場合:

SCH=xx はプログラム特性表 (PPT) を指します (メンバー SCHEDxx)。

#### 667 ページの『IBM MQ サブシステムを z/OS に定義する』の場合:

SSN=ss は定義されたサブシステム・リストを指します (メンバー IEFSSNss)。

### z/OS IBM MQ ロード・ライブラリーの APF 許可を行う

各種ライブラリーの APF 許可を行います。ロード・モジュールの中にはすでに許可されているものもあります。

- この作業は、IBM MQ を実行するそれぞれの z/OS システムごとに実行する必要があります。
- キュー共有グループを使用する場合、シスプレックス内のそれぞれの z/OS システムの IBM MQ の設定値を同一の値に設定する必要があります。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。
- ライブラリー・ルックアサイド (LLA) の使用:
  - IBM MQ の使用方法によっては、ライブラリーからモジュールをロードする際に高入出力 (IO) になる場合があります。この IO は、オペレーティング・システムの LLA 機能を使用することによって軽減できます。
  - この高 IO は、以下の場合に発生する可能性があります。

- MQCONN/MQDISC 率の高いアプリケーション (例えば、WLM ストアード・プロシージャー内)。
- チャネル出口のロード。開始と停止を頻繁に行うチャネルがあり、チャネル出口を使用する場合。

- SYS1.PARMLIB 内のメンバー CSVLLAx は LLA セットアップを指定します。ライブラリー名を LIBRARIES ステートメントに含めるということは、プログラム・コピーは常に VLF (仮想ルックアサイド機能) から取られるので、使用率が高い場合に通常は I/O を必要としないということを意味します。

FREEZE ステートメントに含めるということは、関係のある DD ステートメント連結ディレクトリーを取得する (しばしばプログラム・ロード自体よりも I/O が大きくなることもある) ための I/O が発生しないということを意味します。

これらのライブラリーのいずれかを変更したら、オペレーティング・システム・コマンド「F LLA,REFRESH」を使用してください。

IBM MQ ロード・ライブラリーの thlqual.SCSQAUTH と thlqual.SCSQLINK には、APF 許可が必要です。さらに、各国語機能用のライブラリー (thlqual.SCSQANLx、thlqual.SCSQSNLx) および分散キューイング機能用のライブラリー (thlqual.SCSQMVR1) についても APF 許可が必要です。Advanced Message Security を使用する場合は、ライブラリー thlqual.SDRQAUTH についても APF 許可が必要です。

ただし、LPA 内のすべてのロード・モジュールの APF 許可は自動的に行われます。SYS1.PARMLIB メンバーの IEASYSpp に次のステートメントが含まれている場合は、リンク・リストのすべてのメンバーについても同様です。

```
LNKAUTH=LNKLST
```

LNKAUTH を指定しない場合は、LNKAUTH=LNKLST がデフォルト値になります。

LPA またはリンク・リスト (664 ページの『z/OS リンク・リストおよび LPA を更新する』を参照) に入れるものによっては、ライブラリーを APF リンク・リストに入れる必要はありません。

注: IBM MQ STEPLIB に入れたすべてのライブラリーを APF 許可する必要があります。APF 許可されていないライブラリーを STEPLIB に入れると、ライブラリー連結全体の APF 許可が失われます。

APF リストは、SYS1.PARMLIB のメンバーの PROGxx または IEAAPFaa 内にあります。これらのリストには、APF 許可された z/OS ライブラリーの名前が入っています。リスト内の項目の順序には意味がありません。APF リストについては、「z/OS MVS 初期設定およびチューニング解説書」を参照してください。

システムのチューニングについて詳しくは、SupportPac MP16 を参照してください。

動的フォーマットで PROGxx メンバーを使用する場合は、変更を有効にするために z/OS コマンド SETPROG APF,ADD,DSNAME=h1q.SCSQ XXXX,VOLUME= YYYYYY を発行するだけで済みます。ここで、XXXX はライブラリー名によって異なり、YYYYYY はボリュームです。このコマンドを発行しないで、IEAAPFaa メンバーまたは静的形式を使用した場合は、システムで IPL を実行しなければなりません。

APF リストでは、ライブラリーの実際の名前を使用する必要があります。ライブラリーのデータ・セット別名を使用しようとすると、許可は失敗します。

## 関連概念

664 ページの『z/OS リンク・リストおよび LPA を更新する』

早期コード・ライブラリーの新しいバージョンで LPA ライブラリーを更新します。その他のコードは、リンク・リストまたは LPA に入ることができます。

657 ページの『z/OS でのキュー・マネージャーのカスタマイズ準備』

このトピックでは、インストール可能な機能に関する詳細、各国語機能、およびテストに関する詳細を使用してキュー・マネージャーをカスタマイズし、セキュリティーをセットアップすることについて説明しています。

## z/OS リンク・リストおよび LPA を更新する

早期コード・ライブラリーの新しいバージョンで LPA ライブラリーを更新します。その他のコードは、リンク・リストまたは LPA に入ることができます。

- この作業は、IBM MQ を実行するそれぞれの z/OS システムごとに実行する必要があります。
- キュー共有グループを使用している場合は、キュー・マネージャーを IBM MQ 9.0 にマイグレーションする前に、QSG 内の各キュー・マネージャーの早期コードを IBM MQ 9.0 レベルにリフレッシュする必要があります。

各 LPAR に最新の初期コードをインストールしてから、マイグレーション前のある時点でキュー・マネージャーを一度に 1 つずつリフレッシュします。すべてのキュー・マネージャーを同時にマイグレーションする必要はありません。

- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。詳細については、プログラム・ディレクトリー IBM MQ for z/OS のプログラム・ディレクトリーは、[IBM Publications Center](#) からダウンロードできます ([IBM MQ 9.0 PDF 資料](#)を参照してください)。を参照してください。

注：LPA のデータ・セットは、バージョン固有のものです。システム内で既存の LPA を使用している場合は、使用する LPA を決定するためにシステム管理者に連絡してください。

## 早期コード

サブシステムとして動作させるためには、IBM MQ ロードモジュールを MVS 追加する IBM MQ 必要があるものもあります。これらのモジュールは早期コードと呼ばれ、キュー・マネージャーがアクティブでないときにも実行できます。例えば、IBM MQ コマンド接頭部を使用して、コンソールでオペレーター・コマンドが実行されると、この早期コードが制御を取得し、キュー・マネージャーを開始する必要があるか、または実行中のキュー・マネージャーに要求を受け渡す必要があるかをチェックします。このコードはリンク・パック域 (LPA) にロードされます。早期モジュールは 1 セットあり、すべてのキュー・マネージャーに使用され、IBM MQ の最上位にある必要があります。あるバージョンの IBM MQ の早期コードを、それより前のバージョンの IBM MQ のキュー・マネージャーに使用することはできますが、その逆は不可能です。

早期コードは、以下のロード・モジュールを構成します。

- ライブラリー thqual.SCSQLINK 内の CSQ3INI および CSQ3EPX
- ライブラリー thqual.SCSQSNL x 内の CSQ3ECMX。ここで、x は、ご使用の言語を表す文字です。
  - thlqual.SCSQSNLE (米国英語、大/小文字混合)
  - thlqual.SCSQSNLU (米国英語、大文字)
  - thlqual.SCSQSNLK (日本語)
  - thlqual.SCSQSNLF (フランス語)
  - thlqual.SCSQSNLC (中国語)

IBM MQ には、thqual.SCSQSNL*i* ライブラリーの内容を thqual.SCSQLINK に移動して、SMP/E を通知するユーザー変更が含まれています。このユーザー変更は CSQ8UERL と呼ばれ、Long Term Support または Continuous Delivery については *IBM MQ for z/OS* のプログラム・ディレクトリーで説明されています。これは、[IBM Publications Center](#) からダウンロードできます。

LPA ライブラリー内の早期コードを更新した場合、その早期コードは、次の z/OSIPL (CLPA オプションを使用) の時点から、IPL 中に SYS1.PARMLIB 内の IEFSSNss メンバーにある定義から追加されたすべてのキュー・マネージャー・サブシステムで使用することができます。

以下のように早期コードを LPA に追加することにより、後で追加されたどの新規キュー・マネージャー・サブシステム (667 ページの『[IBM MQ サブシステムを z/OS に定義する](#)』を参照) に対しても、IPL なしで即時に早期コードを使用可能にできます。

- CSQ8UERL を使用しなかった場合は、以下の z/OS コマンドを実行します。

```
SETPROG LPA,ADD,MODNAME=(CSQ3INI,CSQ3EPX),DSNAME=thqual.SCSQLINK
SETPROG LPA,ADD,MODNAME=(CSQ3ECMX),DSNAME=thqual.SCSQSNL x
```

- CSQ8UERL を使用した場合は、次の z/OS コマンドを使用して早期コードを LPA にロードできます。

```
SETPROG LPA,ADD,MASK=*,DSNAME=thqua1.SCSQLINK
```

- Advanced Message Security を使用する場合は、以下の z/OS コマンドを発行して LPA に追加モジュールを組み込む必要もあります。

```
SETPROG LPA,ADD,MODNAME=(CSQ0DRTM),DSNAME=thqua1.SCSQLINK
```

保守を適用した場合、または IBM MQ の新しいバージョンまたはリリースでキュー・マネージャーを再始動する予定の場合は、既に定義されているキュー・マネージャー・サブシステムに対して早期コードを使用可能にすることができます。早期コードを使用可能にするには、以下のステップを実行します。

1. z/OS SETPROG コマンドを使用して、早期コードを LPA に追加します (このトピック内の上記を参照)。
2. IBM MQ コマンド STOP QMGR を使用して、キュー・マネージャーを停止します。
3. qmgr.REFRESH.QMGR セキュリティー・プロファイルがセットアップされていることを確認します。  
[MQSC コマンドとプロファイルとアクセス権のレベルを参照してください。](#)
4. IBM MQ コマンド REFRESH QMGR TYPE(EARLY) を使用して、キュー・マネージャーの早期コードをリフレッシュします。
5. IBM MQ コマンド START QMGR を使用して、キュー・マネージャーを再始動します。

IBM MQ コマンドの STOP QMGR、REFRESH QMGR、および START QMGR については、[MQSC コマンド](#)で説明しています。

## その他のコード

次のライブラリーの IBM MQ 提供のロード・モジュールはすべて再入可能であり、LPA に置くことができます。

- SCSQAUTH
- SCSQANLx (x は言語を表す文字)
- SCSQMVR1

**重要:** ただし、ライブラリーを LPA に置く場合は、保守を適用するたびに、変更したモジュールを手動で LPA にコピーする必要があります。したがって、IBM MQ ロード・ライブラリーをリンク・リストに置く方がよく、これは、保守後に z/OS コマンド REFRESH LLA を発行することによって更新できます。

特に SCSQAUTH の場合には、これを複数の STEPLIB に含めないようにするために、LPA に置くことをお勧めします。LPA またはリンク・リストには、1つの言語ライブラリー SCSQANLx のみを配置します。リンク・リスト・ライブラリーは、SYS1.PARMLIB の LNKLISTkk メンバーで指定されます。

分散キューイング機能および CICS bridge (キュー・マネージャーそのものではない) からは、言語環境プログラム (LE) ランタイム・ライブラリー SCEERUN へアクセスする必要があります。これらのいずれかの機能を使用する場合は、SCEERUN をリンク・リストに組み込む必要があります。

## 関連概念

666 ページの『z/OS プログラム特性表を更新する』

IBM MQ キュー・マネージャーには、いくつかの追加 PPT 項目が必要です。

## z/OS プログラム特性表を更新する

IBM MQ キュー・マネージャーには、いくつかの追加 PPT 項目が必要です。

- この作業は、IBM MQ を実行するそれぞれの z/OS システムごとに実行する必要があります。
- キュー共有グループを使用する場合、シスプレックス内のそれぞれの z/OS システムの IBM MQ の設定値を同一の値に設定する必要があります。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。
- Advanced Message Security を必要とする場合には、この作業の CSQ0DSRV 部分を実行する必要があります。

必要なすべての PPT 項目を含むサンプルが、thlqual.SCSQPROC(CSQ4SCHD) に用意されています。必要な項目が PPT に追加されていることを確認してください。これは SYS1.PARMLIB(SCHEDxx) にあります。

1In z/OS 1.12 and later versions, CSQYASCP is already defined to the operating system with the attributes detailed and no longer needs to be included in a SCHEDxx member of PARMLIB.

IBM MQ キュー・マネージャー自体によって、スワッピングが制御されます。しかし、負荷の高い IBM MQ ネットワークで応答時間が重要な場合、CSQXJST PPT エントリを追加して、IBM MQ チャネルイニシエータをスワップ不可にすることが、z/OS システムの他の部分のパフォーマンスに影響を与えるリスクを考慮すると、有利になる場合があります。

Advanced Message Security が必要な場合は、CSQ0DSRV PPT エントリを追加します。

これらの変更を有効にするために、z/OS コマンド SET SCH= を発行します。

## 関連概念

667 ページの『[IBM MQ サブシステムを z/OS に定義する](#)』

サブシステム名の表を更新し、コマンド接頭部ストリング用の規則を決めます。

## **z/OS** キュー・マネージャーとチャネル・イニシエーターの構成

これらのトピックを、キュー・マネージャーとチャネル・イニシエーターを構成するためのステップバイステップ・ガイドとしてご利用ください。

## **z/OS** IBM MQ サブシステムを z/OS に定義する

サブシステム名の表を更新し、コマンド接頭部ストリング用の規則を決めます。

この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。旧バージョンから移行する際は、この作業を実行する必要はありません。

## 関連概念

671 ページの『[IBM MQ キュー・マネージャー用のプロシージャーを作成する](#)』

各 IBM MQ サブシステムでは、キュー・マネージャーを始動するためのカタログ式プロシージャーが必要です。独自のプロシージャー・ライブラリーを作成するか、あるいは IBM 提供のプロシージャー・ライブラリーを使用することができます。

## **z/OS** サブシステム名表の更新

IBM MQ サブシステムを定義する場合、項目をサブシステム名表に追加する必要があります。

SYS1.PARMLIB メンバーの IEFSSNss から最初に取り出される z/OS のサブシステム名表には、正式に定義された z/OS サブシステムの定義が入っています。個々の IBM MQ サブシステムを定義するには、この表に項目を追加する必要があります。それには、SYS1.PARMLIB の IEFSSNss メンバーを変更する方法と、z/OS コマンド SETSSI を使用する方法 (この方法が望ましい) があります。

IBM MQ サブシステムの初期設定は並列処理をサポートしているため、IBM MQ サブシステム定義ステートメントを、z/OS V1.12 以降で使用可能な IEFSSNss 表の BEGINPARALLEL キーワードの上下に追加することができます。

SETSSI コマンドを使用する場合は変更がただちに有効になるので、システムの IPL を実行する必要はありません。以降の IPL 後も変更が有効になるように、674 ページの『[SYS1.PARMLIB メンバーを更新する](#)』の説明に従って SYS1.PARMLIB も更新してください。

IBM MQ サブシステムを動的に定義するための SETSSI コマンドは、次のとおりです。

```
SETSSI ADD,S=ssid,I=CSQ3INI,P='CSQ3EPX,cpf,scope'
```

IEFSSNss 内の対応する情報は、次の 2 つのいずれかの方法で指定できます。

- IEFSSNss 内の IBM MQ サブシステム定義のキーワード・パラメーター形式。この方法をお勧めします。

```
SUBSYS SUBNAME(ssid) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,cpf,scope')
```

- IBM MQ サブシステム定義の定位置パラメーター形式。

```
ssid,CSQ3INI,'CSQ3EPX,cpf,scope'
```

これら 2 種類の書式を 1 つの IEFSSNss メンバーの中に混在させないでください。異なる書式が必要な場合は、新しいメンバーの SSN オペランドを IEASYSpp SYS1.PARMLIB メンバーに追加し、書式の種類ごとに別の IEFSSNss メンバーを使用します。複数の SSN を指定するには、SSN=(aa,bb,...) を使用します。IEASYSpp に格納されます。

これらの例について以下に説明します。

#### ssid

サブシステム ID。長さは最大 4 文字です。すべての文字は大文字の英数字 (大文字の A ~ Z、0 ~ 9) で、先頭は英字でなければなりません。キュー・マネージャーはサブシステムと同じ名前になるので、z/OS サブシステム名と IBM MQ オブジェクト名の両方で許可される文字のみを使用できます。

#### cpf

コマンド接頭部ストリング (CPF については、669 ページの『コマンド接頭部ストリング (CPF) の定義』を参照)。

#### scope

z/OS シスプレックスで実行している場合に使用されるシステムの有効範囲 (システムの有効範囲については、670 ページの『シスプレックス環境での CPF』を参照)。

668 ページの図 99 は、IEFSSNss ステートメントのいくつかの例を示しています。

```
CSQ1,CSQ3INI,'CSQ3EPX,+mqs1cpf,S'  
CSQ2,CSQ3INI,'CSQ3EPX,+mqs2cpf,S'  
CSQ3,CSQ3INI,'CSQ3EPX,++,S'
```

図 99. サブシステム定義のためのサンプル IEFSSNss ステートメント

注: オブジェクトをサブシステム内に作成したら、サブシステム名を変更したり、あるサブシステムのページ・セットを別のサブシステムの中で使用したりすることができなくなります。これらの作業を行う場合は、すべてのオブジェクトとメッセージを 1 つのサブシステムからアンロードして、別のシステムに再ロードする必要があります。

668 ページの表 39 は、668 ページの図 99 のステートメントで定義された、サブシステム名とコマンド接頭部ストリング (CPF) の関連を示すいくつかの例を示しています。

IBM MQ サブシステム名	CPF
CSQ1	+mqs1cpf
CSQ2	+mqs2cpf
CSQ3	++

注: z/OS コマンド SETSSI の ACTIVATE 機能と DEACTIVATE 機能は、IBM MQ ではサポートされません。

変更の状況を確認するには、SDSF でコマンド /D SSI,L を発行します。「ACTIVE」状況で作成された新規サブシステムが表示されます。

## z/OS コマンド接頭部ストリング (CPF) の定義

IBM MQ の各サブシステム・インスタンスには、そのサブシステムを識別するためのコマンド接頭部ストリングを含めることができます。

CPF には、システム全域に適用される規則を採用し、すべてのサブシステムが対立しないようにします。以下のガイドラインに従ってください。

- CPF を最大 8 文字のストリングとして定義する。
- すでに他のサブシステムで使用されている CPF は、使用しないでください。また、システムに定義されている JES バックスペース文字をストリングの最初の文字として使用しないでください。
- CPF は、669 ページの表 41 に示されている有効な文字セットから選んだ文字を使用して定義してください。
- すでに定義済みのプロセスの省略形である CPF や、コマンド構文と混同する可能性のある CPF は使用しないでください。例えば、「D」などの CPF は、DISPLAY などの z/OS コマンドと対立します。これを回避するには、CPF ストリングの最初の文字あるいは単独の文字として (669 ページの表 41 に示されている) 特殊文字のいずれかを使用してください。
- 既存の CPF のサブセットまたはスーパーセットである CPF を定義しないでください。例については、669 ページの表 40 を参照してください。

表 40. CPF のサブセットおよびスーパーセットの規則例

サブシステム名	定義された CPF	コマンドの宛先
MQA	!A	MQA
MQB	!B	MQB
MQC1	!C1	MQC1
MQC2	!C2	MQC2
MQB1	!B1	MQB

Commands intended for subsystemMQB1(using CPF!B1)は、サブシステムの CPF が存在するため、サブシステム MQB に経路指定されます!B、部分集合!B1。例えば、次のコマンドを入力すると、

```
!B1 START QMGR
```

サブシステム MQB は、次のコマンドを受け取ります。

```
1 START QMGR
```

(この場合、これは処理できません)。

z/OS コマンドの DISPLAY OPDATA を発行すると、存在する接頭部を表示できます。

シスプレックスで実行している場合、z/OS は CPF 登録時にこの種の対立があるかどうかを診断します (CPF の登録については、670 ページの『シスプレックス環境での CPF』を参照してください)。

669 ページの表 41 は、CPF ストリングを定義する際に使用できる文字を示しています。

文字セット	目次
英字	大文字の A ~ Z、小文字の a ~ z
数字	0 ~ 9
国別 (注を参照)	@ \$ # (16 進値として表せる文字)
特殊	. [ ] * & + - = <   ! ; % _ ? : >

## 注:

システムは、次の国別文字の 16 進表示を認識します。@ (X'7C' として)、\$ (X'5B' として) および # (X'7B' として)。U.S 以外の国の場合: 端末キーボードに表示される U.S 国別文字により、別の 16 進表記が生成され、エラーが発生する可能性があります。たとえば、一部の国では、\$ 文字は X'4A' を生成することがあります。

セミコロン (;) は CPF として有効ですが、ほとんどのシステムではコマンド区切り文字として使用されません。

## シスプレックス環境での CPF

このトピックでは、シスプレックス内で CPF を使用方法について知ることができます。

シスプレックス環境で使用する場合、IBM MQ は CPF を登録します。これにより、シスプレックス内の任意のコンソールからコマンドを入力し、そのコマンドを適切なシステムへ実行のために経路指定できるようになります。コマンドの応答は、コマンドが出されたコンソールに戻されます。

## シスプレックス操作の有効範囲の定義

有効範囲は、IBM MQ がシスプレックス環境で実行されている場合に、IBM MQ サブシステムが実行する CPF 登録の種類を決定するために使用されます。

可能な有効範囲の値は次のとおりです。

### M

システムの有効範囲

CPF は、システム IPL 時に IBM MQ によって z/OS に登録され、z/OS システムが活動状態の間は登録されています。

IBM MQ コマンドは、ターゲット・サブシステムを実行している z/OS イメージへ接続されたコンソールで入力するか、ROUTE コマンドを使用してコマンドをそのイメージへ送る必要があります。

シスプレックスで実行していない場合は、このオプションを使用してください。

### S

シスプレックスの開始有効範囲

CPF は、IBM MQ サブシステムの開始時に z/OS に登録され、IBM MQ サブシステムが終了するまで活動状態のままです。

元の START QMGR コマンドをターゲット・システムに送るためには、ROUTE コマンドを使用する必要がありますが、それ以降のすべての IBM MQ コマンドはシスプレックスに接続された任意のコンソールから入力でき、ターゲット・システムに自動的に送られます。

IBM MQ が終了した後、ROUTE コマンドを使用して、それ以降の START コマンドをターゲット IBM MQ サブシステムへ送る必要があります。

### X

シスプレックス IPL 有効範囲

CPF は、システム IPL 時に IBM MQ によって z/OS に登録され、z/OS システムが活動状態の間は登録されています。

IBM MQ コマンドは、シスプレックスへ接続された任意のコンソールで入力できます。また、このコマンドはターゲット・システムを実行しているイメージへ自動的に送られます。

有効範囲 S の CPF を持っている IBM MQ サブシステムは、シスプレックス内の 1 つ以上の z/OS イメージについて定義できるので、これらのイメージは単一のサブシステム名表を共有できます。ただし、初回の START コマンドは、IBM MQ サブシステムが実行される z/OS イメージに対して発行する (またはそのイメージにルーティングされる) ことが必要です。このオプションを使用する場合、シスプレックス内の別の z/OS イメージで IBM MQ サブシステムを停止し、それを再始動できますが、その際にサブシステム名表を変更したり、z/OS システムの IPL を実行したりする必要はありません。

有効範囲 X の CPF を持っている IBM MQ サブシステムは、シスプレックス内の 1 つの z/OS イメージについてのみ定義できます。このオプションを使用する場合は、スコープ X の CPF を使用する IBM MQ サブシステムが必要な z/OS イメージごとに、固有のサブシステム名テーブルを定義する必要があります。

z/OS 自動再始動管理プログラム (ARM) を使用していくつかの異なる z/OS イメージでキュー・マネージャーを自動的に再始動する場合は、キュー・マネージャーが再始動される z/OS イメージごとに、キュー・マネージャーをそれぞれ定義する必要があります。それぞれのキュー・マネージャーは、有効範囲 S の CPF をもつシスプレックス全体に固有な 4 文字のサブシステム名で定義する必要があります。

## **z/OS** IBM MQ キュー・マネージャー用のプロシージャーを作成する

各 IBM MQ サブシステムでは、キュー・マネージャーを始動するためのカタログ式プロシージャーが必要です。独自のプロシージャー・ライブラリーを作成するか、あるいは IBM 提供のプロシージャー・ライブラリーを使用することができます。

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- 前のバージョンから移行する際に、カタログ式プロシージャーを変更しなければならない場合があります。

サブシステム名表に定義された IBM MQ サブシステムごとに、カタログ式プロシージャーを、キュー・マネージャー開始用のプロシージャー・ライブラリーで作成してください。IBM 提供のプロシージャー・ライブラリーの名前は SYS1.PROCLIB ですが、インストール先によっては、独自の命名規則を使用している場合があります。

キュー・マネージャー開始タスク・プロシージャーの名前は、サブシステム名に文字 MSTR を連結して作られます。例えば、サブシステム CSQ1 では、プロシージャー名は CSQ1MSTR になります。定義したサブシステムごとに、1 つのプロシージャーが必要です。

選択した言語のメッセージを入れたライブラリーを組み込む必要があります。

- thlqual.SCSQSNLE (米国英語、大/小文字混合)
- thlqual.SCSQSNLU (米国英語、大文字)
- thlqual.SCSQSNLK (日本語)
- thlqual.SCSQSNLF (フランス語)
- thlqual.SCSQSNLC (中国語)

この製品資料の多くの例と説明では、CSQ1 という名前のサブシステムを使用していることを前提としています。インストール検査とテストのために、CSQ1 という名前のサブシステムを最初に作成すると、これらの例を活用しやすくなります。

2 つのサンプル開始タスク・プロシージャーが thlqual.SCSQPROC 内にあります。メンバー CSQ4MSTR はメッセージの各クラスごとに 1 つのページ・セットを使用し、メンバー CSQ4MSRR はメッセージの主要クラスに複数のページ・セットを使用します。SYS1.PROCLIB (SYS1.PROCLIB を使用していない場合は、ご使用のプロシージャー・ライブラリー) のメンバー xxxxMSTR (この xxxx は IBM MQ サブシステムの名前) に、これらのプロシージャーの 1 つをコピーしてください。各 IBM MQ サブシステムごとに、定義したサンプル・プロシージャーを、プロシージャー・ライブラリーのメンバーにコピーします。

メンバーをコピーした後、メンバー内の説明を使用して、これらのメンバーを各サブシステムの要件に合わせて調整できます。16 MB 境界より下、16 MB 境界より上、および 2 GB バーよりも上の領域サイズの指定に関する詳細は、[提案される領域サイズ](#)を参照してください。さらに、JCL で記号パラメーターを使用して、プロシージャーを開始時に変更できます。複数の IBM MQ サブシステムがある場合は、将来の保守を単純化するために、プロシージャーの共通部分に JCL 組み込みグループを使用すると有利な場合があります。

キュー共有グループを使用する場合は、STEPLIB 連結に Db2 ランタイム・ターゲット・ライブラリー SDSNLOAD を含め、APF 許可する必要があります。このライブラリーは、リンク・リストまたは LPA からアクセスできない場合は、STEPLIB 連結でのみ必要です。

Advanced Message Security を使用する場合は、STEPLIB 連結に thlqual.SDRQAUTH が含まれている必要があります。されにそれが APF 許可されていなければなりません。

注: JCL で使用するためにブートストラップ・データ・セット (BSDS)、ログ、およびページ・セットの名前をメモしておき、プロセスの後の手順でこれらのセットを定義することができます。

## 関連概念

672 ページの『[チャンネル・イニシエーター用のプロシージャーを作成する](#)』

IBM MQ サブシステムごとに、CSQ4CHIN のコピーを調整します。他のどの製品を使用しているかに応じて、他のデータ・セットへのアクセスを許可しなければならないことがあります。

## チャンネル・イニシエーター用のプロシージャーを作成する

IBM MQ サブシステムごとに、CSQ4CHIN のコピーを調整します。他のどの製品を使用しているかに応じて、他のデータ・セットへのアクセスを許可しなければならないことがあります。

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- 前のバージョンから移行する際に、カタログ式プロシージャーを変更しなければならない場合があります。

分散キューイングを使用する IBM MQ サブシステムには、1つのサブシステムにつき1つずつ、チャンネル・イニシエーターの開始タスク・プロシージャーを作成する必要があります。

そのためには、次のようにします。

1. サンプルの開始タスク・プロシージャー `thlqual.SCSQPROC(CSQ4CHIN)` をプロシージャー・ライブラリーにコピーします。プロシージャーに `xxxxx CHIN` という名前を付けます。ここで、`xxxxx` は IBM MQ サブシステムの名前です (例えば、`CSQ1CHIN` は、キュー・マネージャー `CSQ1` のチャンネル・イニシエーター開始タスク・プロシージャーです)。
2. 使用する IBM MQ サブシステムごとに1つずつコピーを作成します。
3. サンプル・プロシージャー `CSQ4CHIN` 内の命令を使用して、要件に合うようにプロシージャーを調整します。さらに、JCL で記号パラメーターを使用して、プロシージャーを開始時に変更できます。これについては、[IBM MQ for z/OS の管理](#)で開始オプションとともに説明しています。

分散キューイング・ライブラリー `thlqual.SCSQMVR1` を連結します。

LE ランタイム・ライブラリー `SCEERUN` へのアクセスが必要です。このライブラリーがリンク・リスト (`SYS1.PARMLIB(LNKLSTkk)`) にない場合は、`STEPLIB DD` ステートメントの中で連結してください。

4. 外部セキュリティ管理プログラムの下で実行できるようにプロシージャーに許可を与えます。
5. 選択した言語のメッセージを入れたライブラリーを組み込む必要があります。

- `thlqual.SCSQSNLE` (米国英語、大/小文字混合)
- `thlqual.SCSQSNLU` (米国英語、大文字)
- `thlqual.SCSQSNLK` (日本語)
- `thlqual.SCSQSNLF` (フランス語)
- `thlqual.SCSQSNLC` (中国語)

チャンネル・イニシエーターは、実行時間の長いアドレス・スペースです。制限された量の CPU が消費されてから終了しないように、以下のいずれかが必要です。

- z/OS システムにおける開始タスクのデフォルトが、CPU 無制限であること。これは、`TIME=(1440,00)` を指定した、`JOBCLASS(STC)` の `JES2` 構成ステートメントで実現します。または、
- `CSQXJST` の `EXEC` ステートメントに、`TIME=1440` または `TIME=NOLIMIT` パラメーターを明示的に追加する。

チャンネル出口を使用する場合は、あとからこのプロシージャーに出口ライブラリー (`CSQXLIB`) を追加できます。そのためにはチャンネル・イニシエーターを停止して再始動する必要があります。

TLS を使用する場合は、システム TLS ランタイム・ライブラリーへのアクセスが必要です。このライブラリーは、`SIEALNKE` と呼ばれます。ライブラリーは、APF 許可が必要です。

TCP/IP を使用する場合は、チャンネル・イニシエーターのアドレス・スペースは、TCP/IP システム・パラメーターが入っている `TCPIP.DATA` データ・セットにアクセスできるようにする必要があります。データ・

セットをセットアップする方法は、使用している TCP/IP 製品とインターフェースによって異なります。次の方法があります。

- 環境変数 RESOLVER\_CONFIG
- HFS ファイル /etc/resolv.conf
- //SYSTCPD DD ステートメント
- //SYSTCPDD DD ステートメント
- jobname/userid.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)
- zapname.TCPIP.DATA

この一部は、開始タスク・プロシージャー JCL に影響を及ぼします。詳しくは、[z/OS 通信サーバー: IP 構成ガイド](#)を参照してください。

## 関連概念

673 ページの『[IBM MQ サブシステムを z/OS WLM サービス・クラスに定義する](#)』

z/OS システムで IBM MQ に適切なパフォーマンス優先度を適用するには、キュー・マネージャーとチャンネル・イニシエーター・アドレス・スペースを適切な z/OS ワークロード管理 (WLM) サービス・クラスに割り当てる必要があります。これを明示的に行わないと、不適切なデフォルトが適用されることがあります。

### **z/OS** IBM MQ サブシステムを z/OS WLM サービス・クラスに定義する

z/OS システムで IBM MQ に適切なパフォーマンス優先度を適用するには、キュー・マネージャーとチャンネル・イニシエーター・アドレス・スペースを適切な z/OS ワークロード管理 (WLM) サービス・クラスに割り当てる必要があります。これを明示的に行わないと、不適切なデフォルトが適用されることがあります。

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。

WLM で提供される ISPF ダイアログを使用して、以下のタスクを実行します。

- WLM 結合データ・セットから z/OS WLM ポリシー定義を取り出す。
- このポリシー定義を更新するため、キュー・マネージャーおよびチャンネル・イニシエーター開始タスク・プロシージャーの名前を、選択したサービス・クラスに追加する。
- WLM 結合データ・セットに、変更済みのポリシーをインストールする。

この後、次の z/OS コマンドを使用してこのポリシーをアクティブ化します。

```
V WLM,POLICY=policyname,REFRESH
```

パフォーマンス・オプションの設定について詳しくは、[こちら](#)を参照してください。

## 関連概念

708 ページの『[Db2 環境のセットアップ](#)』

キュー共有グループを使用する場合は、いくつかのサンプル・ジョブをカスタマイズして実行することで、必要な Db2 オブジェクトを作成する必要があります。

### **z/OS** ESM セキュリティー管理を実施する

キュー・マネージャーとチャンネル・イニシエーターのセキュリティー管理を実装します。

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

外部セキュリティー管理プログラムとして RACF® を使用している場合は、[z/OS でのセキュリティーのセットアップ](#)に、これらのセキュリティー管理の実施方法についての説明があります。

チャンネル・イニシエーターを使用する場合は、次のようにする必要もあります。

- サブシステムが接続セキュリティーを活動状態にしている場合は、接続セキュリティー・プロファイル `ssid.CHIN` を外部セキュリティー管理プログラムに定義します (詳細は、[チャンネル・イニシエーターのための接続セキュリティー・プロファイル](#)を参照)。
- Transport Layer Security (TLS) またはソケット・インターフェースを使用する場合は、ユーザー ID (そのユーザーの権限でチャンネル・イニシエーターが実行されている) が UNIX システム・サービスを使用する設定になっていることを確認してください (設定方法については、「[OS/390®UNIX システム・サービスの計画](#)」を参照してください)。
- TLS を使用する場合は、チャンネル・イニシエーターの実行に使用するユーザー ID が、ALTER QMGR コマンドの SSLKEYR パラメーターに指定されている鍵リングにアクセスする設定になっていることを確認してください。

キュー・マネージャーを始動する前に、次の方法によって IBM MQ のデータ・セットとシステム・セキュリティーを設定する必要があります。

- キュー・マネージャーの開始タスク・プロシージャーが、外部セキュリティー管理プログラムの下で実行できるように許可を与える。
- キュー・マネージャーのデータ・セットへのアクセスの許可を与える。

この方法の詳細は、「[z/OS\(r\) のセキュリティーのインストール作業](#)」を参照してください。

RACF を使用する場合は、RACF STARTED クラスを使用するなら、システムの IPL を実行する必要はありません (開始済みタスク・プロシージャーに関する RACF 権限を参照)。

## 関連概念

674 ページの『[SYS1.PARMLIB メンバーを更新する](#)』

IPL の実行後も変更を有効な状態に保つには、SYS1.PARMLIB のいくつかのメンバーを更新する必要があります。

712 ページの『[キュー共有グループの ESM セキュリティー管理の実装](#)』

キュー共有グループ内のすべてのキュー・マネージャーが Db2 とカップリング・ファシリティー・リスト構造にアクセスするためのセキュリティー管理を実装します。

## **SYS1.PARMLIB メンバーを更新する**

IPL の実行後も変更を有効な状態に保つには、SYS1.PARMLIB のいくつかのメンバーを更新する必要があります。

- この作業は、IBM MQ を実行するそれぞれの z/OS システムごとに実行する必要があります。
- キュー共有グループを使用する場合、シスプレックス内のそれぞれの z/OS システムの IBM MQ の設定値を同一の値に設定する必要があります。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

SYS1.PARMLIB メンバーを以下のように更新します。

1. メンバー IEFSSNss を更新します。これについては、667 ページの『[IBM MQ サブシステムを z/OS に定義する](#)』で説明されています。
2. IPL の実行時に以下のメンバーが使用されるように IEASYSpp を変更します。
  - PROGxx または IEAAPFaa メンバー (663 ページの『[IBM MQ ロード・ライブラリーの APF 許可を行う](#)』で使用)
  - LNKLSTkk および LPALSTmm メンバー (664 ページの『[z/OS リンク・リストおよび LPA を更新する](#)』で使用)
  - SCHEDxx メンバー (666 ページの『[z/OS プログラム特性表を更新する](#)』で使用)
  - IEFSSNss メンバー (667 ページの『[IBM MQ サブシステムを z/OS に定義する](#)』で使用)

## 関連概念

675 ページの『[初期設定入力データ・セットをカスタマイズする](#)』

サンプル初期設定入力データ・セットの作業用コピーを作成し、システム要件に合わせて調整します。

## 初期設定入力データ・セットをカスタマイズする

サンプル初期設定入力データ・セットの作業用コピーを作成し、システム要件に合わせて調整します。

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行する必要があります。

IBM MQ キュー・マネージャーはそれぞれ、初期定義を IBM MQ 初期設定入力データ・セットに入っている一連のコマンドから取得します。このデータ・セットは、キュー・マネージャー開始タスク・プロシージャの中で定義される DD 名 CSQINP1、CSQINP2、および CSQINPT によって参照されます。

これらのコマンドに対する応答は、DD 名 CSQOUT1、CSQOUT2、および CSQOUTT で参照される初期設定出力データ・セットに書き込まれます。

元のサンプルを保存するために、各サンプルの作業用コピーを作成します。次に、これらの作業用コピー内のコマンドを、システムの要件に合わせて調整できます。

複数の IBM MQ サブシステムを使用する場合は、初期設定入力データ・セットの名前の高位修飾子にサブシステム名を組み込むことで、各データ・セットと関連付けられている IBM MQ サブシステムをより簡単に識別できます。

サンプルについて詳しくは、以下のトピックを参照してください。

- [初期設定データ・セットの形式](#)
- [CSQINP1 サンプルの使用](#)
- [CSQINP2 サンプルの使用](#)
- [CSQINPX サンプルの使用](#)
- [CSQINPT サンプルの使用](#)

### 初期設定データ・セットの形式

初期設定入力データ・セットは、区分データ・セット (PDS) メンバーでも順次データ・セットでも構いません。初期設定入力データ・セットは、一連のデータ・セットを連結できます。レコード長 80 バイトで定義しますが、次の条件があります。

- 1 桁目から 72 桁目までだけが有効です。73 桁目から 80 桁目は無視されます。
- 1 桁目にアスタリスク (\*) が付いたレコードは注釈として解釈され、無視されます。
- ブランクのレコードは無視されます。
- 各コマンドは、新しいレコードから開始しなければなりません。
- 行末の - は、次レコードの 1 桁目に継続することを意味します。
- 行末の + は、次レコードの最初のブランク以外の桁に継続することを意味します。
- 1 つのコマンドに使用できる最大文字数は、32 762 です。

初期設定出力データ・セットは順次データ・セットです。レコード長は 125、レコード形式は VBA、ブロック・サイズは 629 です。

### CSQINP1 サンプルの使用

データ・セット thlqual.SCSQPROC は、バッファ・プール、バッファ・プール関連ページ・セット、および ALTER SECURITY コマンドの定義が含まれている 2 つのメンバーを保持しています。

メンバー CSQ4INP1 は、メッセージのクラスごとに 1 つのページ・セットを使用します。メンバー CSQ4INPR は、メッセージの主要クラス用に複数のページ・セットを使用します。

適切なサンプルを、キュー・マネージャー開始タスク・プロシージャの CSQINP1 連結に組み込みます。

注：

1. IBM MQ は、ゼロから 99 までの範囲で最大 100 個のバッファ・プールをサポートします。DEFINE BUFFPOOL コマンドは、CSQINP1 初期設定データ・セットのみから発行できます。サンプル内の定義では、4つのバッファ・プールを指定しています。
2. キュー・マネージャーで使用される各ページ・セットは、DEFINE PSID コマンドを使用して、CSQINP1 初期設定データ・セットに定義しなければなりません。ページ・セット定義は、バッファ・プール ID とページ・セットを関連付けます。バッファ・プールを指定しなかった場合、バッファ・プール・ゼロ (0) がデフォルトとして使用されます。  
ページ・セット・ゼロ (00) は、必ず定義する必要があります。このページ・セットには、すべてのオブジェクト定義が入ります。キュー・マネージャーごとに最大で 100 のページ・セットを定義できます。
3. ALTER SECURITY コマンドは、セキュリティー属性の TIMEOUT および INTERVAL を変更するのに使用できます。CSQ4INP1 では、デフォルトは TIMEOUT が 54、INTERVAL が 12 に定義されています。

バッファ・プールおよびページ・セットの編成については、[z/OS](#)での計画を参照してください。

キュー・マネージャーの実行中にバッファ・プールおよびページ・セットの定義を動的に変更する場合は、CSQINP1 定義も更新する必要があります。これらの変更は、バッファ・プール定義に REPLACE 属性が含まれていなければ、IBM MQ のコールド・スタートの場合にのみ保持されます。

## CSQINP2 サンプルの使用

次の表では、キュー・マネージャー開始タスク・プロシージャの CSQINP2 連結に組み込むことができる `thlqual.SCSQPROC` のメンバーをリストし、その機能について説明します。命名規則は CSQ4INS\* です。ご使用の構成に合わせて CSQ4INY\* を変更する必要があります。ただし、CSQINS\* メンバーは変更しないでください。次のリリースへのマイグレーション時に変更を再適用する必要があるためです。代わりに、CSQ4INY\* メンバーに DEFINE コマンドまたは ALTER コマンドを入れてください。

メンバー名	説明
CSQ4INSG	システム・オブジェクト定義。
CSQ4INSA	チャンネル認証のシステム・オブジェクトおよびデフォルト・ルール。
CSQ4INSX	システム・オブジェクト定義。
CSQ4INSS	キュー共有グループを使用する場合は、このメンバーをカスタマイズして組み込む。
CSQ4INSJ	JMS を使用してパブリッシュ/サブスクライブを行う場合は、このメンバーをカスタマイズして組み込む。
CSQ4INSM	拡張メッセージ・セキュリティーのためのシステム・オブジェクト定義。
CSQ4INSR	WebSphere Application Server を使用している場合、または IBM MQ V7 以降のキュー型パブリッシュ/サブスクライブ・デーモンによりサポートされるキュー型パブリッシュ/サブスクライブ・インターフェースを使用する場合は、このメンバーをカスタマイズして組み込む。
CSQ4DISP	オブジェクト定義を表示する CSQINP2 サンプル。
CSQ4INYC	クラスター化定義。
CSQ4INYD	分散キューイング定義。
CSQ4INYG	一般定義。
CSQ4INYR	メッセージの主要クラスに複数のページ・セットを使用するストレージ・クラス定義。
CSQ4INYS	メッセージの各クラスに 1 つのページ・セットを使用するストレージ・クラス定義。

オブジェクトの定義は1回のみ必要です。キュー・マネージャーを始動するたびに定義する必要はありません。そのため、これらの定義を毎回 CSQINP2 に含める必要はありません。それらを毎回含めると、すでに存在しているオブジェクトを定義することになり、次のようなメッセージを受け取ります。

```
CSQM095I +CSQ1 CSQMAQLC QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) ALREADY EXISTS
CSQM090E +CSQ1 CSQMAQLC FAILURE REASON CODE X'00D44003'
CSQ9023E +CSQ1 CSQMAQLC ' DEFINE QLOCAL ' ABNORMAL COMPLETION
```

オブジェクトが、このエラーによって損傷することはありません。CSQINP2 連結の中に SYSTEM 定義データ・セットを残しておきたい場合は、各オブジェクトに対して REPLACE 属性を指定することにより、このエラー・メッセージを回避できます。

## CSQINPX サンプルの使用

サンプル `thlqual.SCSQPROC(CSQ4INPX)` には、チャンネル・イニシエーターの始動のたびに実行する必要がある一連のコマンドが入っています。これらは一般的に、START LISTENER などのチャンネル関連コマンドです。これらのコマンドは、キュー・マネージャーが開始されるのではなく、チャンネル・イニシエーターが開始されるたびに毎回必要になります。また、入力データ・セット CSQINP1 あるいは CSQINP2 内では許可されません。このサンプルは使用前にカスタマイズする必要があり、カスタマイズしたあとでチャンネル・イニシエーター用の CSQINPX データ・セットに組み込むことができます。

データ・セットに含まれた IBM MQ コマンドは、チャンネル・イニシエーターの初期設定の終わりに実行され、出力は CSQOUTX DD ステートメントによって指定されたデータ・セットへ書き込まれます。出力は、IBM MQ ユーティリティ・プログラム (CSQUTIL) の COMMAND 機能によって作成された出力と同様です。詳細については、[CSQUTIL ユーティリティ](#)を参照してください。

チャンネル・コマンドのみでなく、CSQUTIL から発行できるどの IBM MQ コマンドでも指定できます。CSQINPX が処理されている間、他のソースからコマンドを入力できます。すべてのコマンドは、前のコマンドが正常に実行されたかどうかにかかわらず、順次に発行されます。

コマンド応答時間を指定するには、データ・セットの中の最初のコマンドとして疑似コマンドの COMMAND を使用できます。これは1つのオプションのキーワード RESPTIME (*nnn*) を取ります。ここで、*nnn* は、各コマンドに回答する場合の待ち時間 (秒単位) です。これは 5 ~ 999 の範囲にあります。デフォルト値は 30 です。

IBM MQ が、4つのコマンドへの応答が長すぎることを検出すると、CSQINPX の処理は停止され、それ以上コマンドは発行されません。チャンネル・イニシエーターは停止されませんが、メッセージ [CSQU052E](#) は CSQOUTX データ・セットに書き込まれ、メッセージ [CSQU013E](#) がコンソールに送信されます。

IBM MQ が CSQINPX の処理を正常に完了すると、メッセージ [CSQU012I](#) がコンソールに送られます。

## CSQINPT サンプルの使用

次の表では、キュー・マネージャー開始タスク・プロシーチャーの CSQINPT 連結に組み込むことができる `thlqual.SCSQPROC` のメンバーをリストし、その機能について説明します。

メンバー名	説明
CSQ4INST	システム・デフォルト・サブスクリプション定義。
CSQ4INYT	パブリッシュ/サブスクライブ定義。

データ・セットに含まれた IBM MQ コマンドは、パブリッシュ/サブスクライブの初期設定の完了時に実行され、出力は CSQOUTT DD ステートメントによって指定されたデータ・セットに書き込まれます。出力は、IBM MQ ユーティリティ・プログラム (CSQUTIL) の COMMAND 機能によって作成された出力と同様です。詳細については、[CSQUTIL ユーティリティ](#)を参照してください。

## 関連概念

678 ページの『ブートストラップ・データ・セットおよびログ・データ・セットを作成する』

ブートストラップ・データ・セット (BSDS) およびログ・データ・セットを準備するには、提供されたプログラム CSQJU003 を使用します。

### **z/OS** ブートストラップ・データ・セットおよびログ・データ・セットを作成する

ブートストラップ・データ・セット (BSDS) およびログ・データ・セットを準備するには、提供されたプログラム CSQJU003 を使用します。

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。

単一または重複のロギング環境を作成する CSQJU003 を実行するためのサンプル JCL およびアクセス方式サービス (AMS) 制御ステートメントが、thlqual.SCSQPROC(CSQ4BSDS) に保持されています。BSDS およびログを作成するため、およびログを事前フォーマットするためには、このジョブをカスタマイズして、実行してください。

**重要:** 最新バージョンの CSQ4BSDS を使用するか、JCL を手動で更新して RECORDS(850 60) を使用しなければなりません。

671 ページの『IBM MQ キュー・マネージャー用のプロシージャーを作成する』で説明している開始タスク・プロシージャー CSQ4MSTR は、次の形式のステートメントの BSDS を参照します。

```
//BSDS1 DD DSN=++HLQ++.BSDS01,DISP=SHR
//BSDS2 DD DSN=++HLQ++.BSDS02,DISP=SHR
```

ログ・データ・セットは、BSDS によって参照されます。

#### 注:

1. BLKSIZE を、LOGDEF ステップの SYSPRINT DD ステートメントに指定しなければなりません。この BLKSIZE は 629 でなければなりません。
2. ブートストラップ・データ・セットおよびログ・データ・セットを異なるキュー・マネージャーから識別するのに役立つために、これらのデータ・セットの高位修飾子にサブシステム名を使用してください。
3. キュー共有グループを使用する場合、SHAREOPTIONS(2 3) を指定して、ブートストラップ・データ・セットおよびログ・データ・セットを定義する必要があります。

ブートストラップとログ・データ・セット、およびそれらのサイズの計画方法について詳しくは、[z/OS](#) での計画を参照してください。

IBM MQ 8.0 では、8 バイト・ログ RB の機能強化により、[ログ相対バイト・アドレスのラガーの説明](#)に従って、キュー・マネージャーの可用性が向上します。キュー・マネージャーを最初に開始する前にキュー・マネージャーで 8 バイトのログ RBA を有効にするには、ロギング環境の作成後に以下の手順を実行します。

1. **IDCAMS ALTER** を使用して、バージョン 1 形式の BSDS (CSQJU003 プログラムを使用して作成されたもの) を ++HLQ++.V1.BSDS01 のような名前に変更します。

**注:** VSAM クラスタとともに、データおよび索引コンポーネントを必ず名前変更してください。

2. 既に定義されている属性を使用して、新規の BSDS を割り振ります。これらは、開始時にキュー・マネージャーによって使用されるバージョン 2 形式の BSDS になります。
3. BSDS 変換ユーティリティ (CSQJUCNV) を実行して、バージョン 1 形式の BSDS を新規のバージョン 2 形式の BSDS に変換します。
4. 変換が正常に完了したら、バージョン 1 形式の BSDS を削除します。

**注:** キュー・マネージャーがキュー共有グループに含まれる場合、8 バイトのログ RBA を有効にするためには、その前に次のようにしてキュー共有グループ内のすべてのキュー・マネージャーを開始しておく必要があります。

- キュー・マネージャーが IBM MQ 8.0.0 である場合は、**OPMODE(NEWFUNC,800)** を指定して開始しておく必要があります。
- キュー・マネージャーが IBM MQ 9.0.0 LTS である場合は、**OPMODE(NEWFUNC,900)** または **OPMODE(NEWFUNC,800)** を指定して開始しておく必要があります。
- キュー・マネージャーが IBM MQ 9.0.x CD、IBM MQ 9.1.0 LTS、またはそれ以降のレベルである場合は、そのレベルで開始するだけで済みます。

## 関連概念

679 ページの『ページ・セットの定義』

提供されているサンプルの 1 つを使用して、キュー・マネージャーごとにページ・セットを定義します。

## z/OS ページ・セットの定義

提供されているサンプルの 1 つを使用して、キュー・マネージャーごとにページ・セットを定義します。

- この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。

IBM MQ キュー・マネージャーごとに別々のページ・セットを定義します。thlqual.SCSQPROC(CSQ4PAGE) および thlqual.SCSQPROC(CSQ4PAGR) には、ページ・セットの定義とフォーマットを行うための JCL 制御ステートメントと AMS 制御ステートメントが入っています。メンバー CSQ4PAGE はメッセージの各クラスごとに 1 つのページ・セットを使用し、メンバー CSQ4PAGR はメッセージの主要クラスに複数のページ・セットを使用します。JCL は、提供されたユーティリティ・プログラム CSQUTIL を実行します。サンプルを検討し、必要なページ・セットの数と使用するサイズに合わせてそれらをカスタマイズしてください。ページ・セットについて、および適切なサイズの計算方法について詳しくは、[z/OS](#) での計画を参照してください。

671 ページの『IBM MQ キュー・マネージャー用のプロシージャを作成する』で説明している開始タスク・プロシージャ CSQ4MSTR は、次の形式のステートメントでページ・セットを参照します。

```
//CSQP00 nn DD DISP=OLD,DSN= xxxxxxxxxxx
```

ここで、*nn* は 00 ~ 99 までのページ・セット番号、*xxxxxxxxxxxx* は、定義するデータ・セットです。

## 注:

1. 動的ページ・セット拡張機能を使用する場合は、2 次エクステントを各ページ・セットに定義しておかなければなりません。thlqual.SCSQPROC(CSQ4PAGE) は、これを行う方法を示しています。
2. キュー・マネージャーごとにページ・セットが識別しやすくなるように、各ページ・セットへ関連付けられるデータ・セットの高位修飾子にサブシステム名を使用してください。
3. ユーティリティ・プログラム CSQUTIL の FORMAT 機能で FORCE オプションを使用する場合は、AMS DEFINE CLUSTER ステートメントに REUSE 属性を追加しなければなりません。これについては、[IBM MQ for z/OS の管理](#)で説明しています。
4. ご使用のページ・セットが 4 GB を超える場合は、ストレージ管理サブシステム (SMS) の EXTENDED ADDRESSABILITY 機能を使用する必要があります。

## 関連概念

711 ページの『IBM MQ 項目を Db2 表に追加する』

キュー共有グループを使用している場合は、CSQ5PQSG ユーティリティを実行して、キュー共有グループとキュー・マネージャーの項目を Db2 データ共有グループの IBM MQ 表に追加します。

## z/OS システム・パラメーター・モジュールの調整

IBM MQ システム・パラメーター・モジュールは、IBM MQ がその処理で使用するロギング、保存、トレース、および接続の環境を制御します。デフォルトのモジュールが提供されます。いくつかのパラメーター (データ・セット名など) は通常、サイト固有であるため、独自のシステム・パラメーター・モジュールを作成する必要があります。

- この作業は、必要に応じて IBM MQ キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。詳細については、[z/OS での IBM MQ のマイグレーション](#)を参照してください。
- 既存のキュー・マネージャーで *Advanced Message Security for z/OS* を有効にするために必要なのは、682 ページの『CSQ6SYSP の使用』で説明されているように、SPLCAP を YES に設定するだけです。このキュー・マネージャーを初めて構成する場合は、この作業全体を実行してください。

システム・パラメーター・モジュールには、次の **V9.0.3** 4 つのマクロがあります。

マクロ名	目的
CSQ6SYSP	接続およびトレースのパラメーターを指定します。682 ページの『CSQ6SYSP の使用』を参照してください。
CSQ6LOGP	ログの初期設定を制御します。690 ページの『CSQ6LOGP の使用』を参照してください。
CSQ6ARVP	アーカイブの初期設定を制御します。694 ページの『CSQ6ARVP の使用』を参照してください。
<b>V9.0.3</b> <b>V9.0.3</b> CSQ6USGP	使用率レコーディングを制御します。701 ページの『CSQ6USGP の使用』を参照してください。

IBM MQ には、デフォルトのシステム・パラメーター・モジュール CSQZPARM があります。このモジュールは、(PARM パラメーターを指定しないで) START QMGR コマンドを発行すると自動的に呼び出され、IBM MQ のインスタンスを開始します。CSQZPARM は APF 許可ライブラリー thlqual.SCSQAUTH に入っています。このライブラリーも IBM MQ で提供されています。これらのパラメーターの値は、IBM MQ の開始時に一連のメッセージとして表示されます。

このコマンドの使用法について詳しくは、[START QMGR](#) を参照してください。

## 独自のシステム・パラメーター・モジュールの作成

必要なシステム・パラメーターが CSQZPARM に入っていない場合、thlqual.SCSQPROC(CSQ4ZPRM) で提供されているサンプル JCL を使用して、独自のシステム・パラメーター・モジュールを作成することができます。

独自のシステム・パラメーター・モジュールを作成するには、次のようにします。

1. JCL サンプルの作業用コピーを作成します。
2. 必要に応じて、コピー内の各マクロに対応するパラメーターを編集します。マクロ呼び出しからいずれかのパラメーターを取り除くと、実行時にデフォルトの値が自動的に設定されます。
3. ロード・モジュールに付ける名前前でプレースホルダー ++NAME++ を置き換えます (CSQZPARM にすることができます)。
4. 使用しているアセンブラが高水準アセンブラでない場合は、アセンブラの要求に従って JCL を変更します。
5. JCL を実行することにより、調整されているバージョンのシステム・パラメーター・マクロをアセンブルおよびリンク・エディットしてロード・モジュールを生成します。これが、指定した名前を取る新しいシステム・パラメーター・モジュールになります。
6. 生成されたロード・モジュールを APF 許可ユーザー・ライブラリーに書き込みます。
7. ユーザーの READ アクセス権限を APF 許可ユーザー・ライブラリーに追加します。
8. このライブラリーを、IBM MQ キュー・マネージャーの開始タスク・プロシージャ STEPLIB に組み込みます。このライブラリー名は、STEPLIB 内のライブラリー thlqual.SCSQAUTH より前になければなりません。
9. キュー・マネージャーを起動するときに、新しいシステム・パラメーター・モジュールを呼び出します。例えば、新しいモジュールが NEWMODS という名前の場合は、次のコマンドを発行します。

```
START QMGR PARM(NEWMODS)
```

10. コマンドが正常に完了したことを確認するため、ジョブ・ログを調べます。ログに次のようなエンタリーが含まれているはずです。

```
CSQ9022I CDL1 CSQYASCP 'START QMGR' NORMAL COMPLETION
```

キュー・マネージャー開始 JCL で、パラメーター・モジュール名を指定することもできます。詳細については、[キュー・マネージャーの開始と停止](#)を参照してください。

注: モジュールの名前として CSQZPARM を選択している場合は、START QMGR コマンドに PARM パラメーターを指定する必要はありません。

## システム・パラメーター・モジュールの微調整

IBM MQ には、3つのアセンブラー・ソース・モジュールからなるセットもあります。このセットは、既存のシステム・パラメーター・モジュールを微調整するために使用します。これらのモジュールはライブラリー `thlqual.SCSQASMS` に入っています。これらのモジュールは、通常、テスト環境でシステム・パラメーター・マクロのデフォルト・パラメーターを変更するために使用します。各ソース・モジュールは、次のようにそれぞれ異なるシステム・パラメーター・マクロを呼び出します。

アセンブラー・ソース・モジュール...	呼び出されるマクロ...
CSQFSYSP	CSQ6SYSP (接続パラメーターおよびトレース・パラメーター)
CSQJLOGP	CSQ6LOGP (ログの初期設定)
CSQJARVP	CSQ6ARVP (保存の初期設定)

上記のモジュールの使用法は次のとおりです。

1. 各アセンブラー・ソース・モジュールの作業用コピーを、ユーザー・アセンブラー・ライブラリーに作成します。
2. 必要に応じてパラメーターの値を追加または変更して、コピーを編集します。
3. 編集したモジュールのコピーをアセンブルして、ユーザー・オブジェクト・ライブラリーにオブジェクト・モジュールを作成します。
4. これらのオブジェクト・コード・モジュールを既存のシステム・パラメーター・モジュールと共にリンク・エディットし、新しいシステム・パラメーター・モジュールとなるロード・モジュールを作成します。
5. 新しいシステム・パラメーター・モジュールが、ユーザー許可ライブラリーのメンバーとなるようにします。
6. このライブラリーを、キュー・マネージャーの開始タスク・プロシージャ STEPLIB に組み込みます。このライブラリーは、STEPLIB 内のライブラリー `thlqual.SCSQAUTH` より前になければなりません。
7. 前述と同様に、PARM パラメーターに新しいモジュール名を指定した START QMGR コマンドを発行して、新しいシステム・パラメーター・モジュールを呼び出します。

サンプル `usermod` は、カスタマイズされたシステム・パラメーターを SMP/E 制御の下で管理する方法を示す、SCSQPROC のメンバー `CSQ4UZPR` で提供されます。

## システム・パラメーターの変更

システム・パラメーターの中には、キュー・マネージャーの実行中に変更できるものもあります。[SET SYSTEM](#)、[SET LOG](#)、および [SET ARCHIVE](#) コマンドを参照してください。

キュー・マネージャーを始動するたびに SET コマンドが有効になるようにするため、コマンドを初期設定入力データ・セットに書き込んでください。

## 関連概念

702 ページの『[チャンネル・イニシエーター・パラメーターを調整する](#)』

ALTER QMGR を使用して、要件に適合するようにチャンネル・イニシエーターをカスタマイズします。

## CSQ6SYSP の使用

このトピックは、CSQ6SYSP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

CSQ6SYSP のデフォルト・パラメーターと、SET SYSTEM コマンドを使用して各パラメーターを変更できるかどうかについては、682 ページの表 44 に示されています。これらの値を変更する場合は、該当するパラメーターの詳細記述を参照してください。

パラメーター	説明	デフォルト値	SET コマンド
ACELIM	ACE ストレージ・プールのサイズ (1 KB ブロック単位)。	0 (制限なし)	✓
CLCACHE	使用するクラスター・キャッシュのタイプを指定します。	STATIC	-
CMDUSER	コマンド・セキュリティ検査のためのデフォルト・ユーザー ID。	CSQOPR	-
CONNSWAP	 <b>重要</b> ：IBM MQ 9.0 以降、このキーワードには効果がありません。  特定の IBM MQ API 呼び出しを発行中のジョブがスワップ可能かスワップ不能かを指定します。	YES	-
EXCLMSG	すべてのログから除外するメッセージのリストを指定します。このリストにあるメッセージは、z/OS コンソールおよびハードコピー・ログに送られません。結果として、707 ページの『 <a href="#">通知メッセージを抑止する</a> 』に記載している方式を使用するより、EXCLMSG パラメーターを使用してメッセージを除外する方が、CPU の観点から見れば効率的です。	( )	✓
EXITLIM	キュー・マネージャー出口が呼び出されるたびに実行できる時間 (秒単位)。	30	-
EXITTCB	キュー・マネージャー出口の実行に使用する開始済みサーバー・タスクの数。	8	-
LOGLOAD	1つのチェックポイントの開始から次のチェックポイントの開始までの間に IBM MQ によって書き込まれるログ・レコードの数。	500 000	✓
MULCCAPT	従量制ソフトウェア料金 (MULC) で使用されるデータを収集するためのアルゴリズムを制御する「従量制価格設定」プロパティを決定します。	<a href="#">パラメーターの説明</a> を参照してください。	-
OTMACON	OTMA 接続パラメーター。	<a href="#">パラメーターの説明</a> を参照してください。	-

表 44. CSQ6SYSP パラメーターのデフォルト値 (続き)

パラメーター	説明	デフォルト値	SET コマンド
QINDXBLD	キュー・マネージャーの再始動を、すべての索引が再作成されるまで待つか、すべての索引が再作成される前に完了するかを決定します。	WAIT	-
QMCCSID	キュー・マネージャー用のコード化文字セット ID。	0	-
QSGDATA	キュー共有グループ・パラメーター。	パラメーターの説明を参照してください。	-
RESAUDIT	RESLEVEL 監査パラメーター。	YES	-
ROUTCDE	特定のコンソールから送信請求されていないメッセージに割り当てられるメッセージ宛先コード。	1	-
SERVICE	IBM の使用のため予約済み。	0	✓
SMFACCT	キュー・マネージャーが開始されるときに SMF アカウンティング・データを収集するかどうかを指定します。 クラス 4 チャンネル・アカウンティング・データは、チャンネル・イニシエーターの開始時のみ収集される点に注意してください。	NO	-
SMFSTAT	キュー・マネージャーが開始されるときに SMF 統計を収集するかどうかを指定します。 クラス 4 チャンネル・イニシエーター統計データは、チャンネル・イニシエーターの開始時のみ収集される点に注意してください。	NO	-
SPLCAP	このキュー・マネージャーのキュー・セキュリティ・ポリシー機能を有効にするかどうかを指定します。Advanced Message Security for z/OS の場合は、このパラメーターを YES に設定します。	NO	-
STATIME	統計を収集するデフォルトの時間間隔 (分単位)。	30	✓
TRACSTR	トレースを自動的に開始するかどうかを指定します。	NO	-
TRACTBL	グローバル・トレース機能によって使用されるトレース表のサイズ (4 KB ブロック単位)。	99 (396 KB)	✓
WLMTIME	WLM 管理対象キューのキュー索引を走査する時間間隔。	30	-
WLMTIMU	WLMTIME の単位 (分または秒)	MINS	-

**ACELIM**

ACE ストレージ・プールの最大サイズを指定します (1 KB ブロック単位)。この数値は 0 から 999999 の範囲でなければなりません。デフォルト値である 0 は、システムで使用できるサイズ以外の制約は課されないことを意味します。

ACELIM の値は、適正な範囲を超えた量の ECSA ストレージを使用していると見なされているキュー・マネージャーに対してのみ設定する必要があります。ACE ストレージ・プールを制限すると、システム内の接続数が制限されるという影響があるため、キュー・マネージャーで使用される ECSA ストレージの量も制限されます。

キュー・マネージャーが制限に達すると、アプリケーションは新しい接続を取得できなくなります。新しい接続を取得できないと MQCONN の処理が失敗し、RRS によって調整されるアプリケーションは、どの IBM MQ API でも障害が発生する可能性があります。

ACE は、接続のスレッド関連制御ブロックに必要な全体の ECSA の約 12.5% を表します。したがって、例えば ACELIM=5120 を指定すると、キュー・マネージャーによって割り振られる ECSA の総量 (スレッド関連制御ブロックの場合) の上限は、約 40960K; (5120 x 8) になります。

スレッド関連制御ブロック用にキュー・マネージャーによって割り振られる ECSA の総量の上限を 5120K に設定するには、ACELIM 値を 640 に設定する必要があります。

統計 CLASS(3) のトレースで生成される SMF 115 サブタイプ 5 のレコードを使用して、「ACE/PEB」ストレージ・プールのサイズをモニターできます。これにより、ACELIM に適切な値を設定できます。

キュー・マネージャーが制御ブロックに使用する ECSA ストレージの合計量を、統計 CLASS(2) トレースによって書き込まれる SMF 115 サブタイプ 7 のレコードから取得できます (QSRSPHBT 内の最初の 2 つの要素の合計)。

ACELIM の設定は、キュー・マネージャーへのアプリケーション接続の制御手段というよりは、z/OS イメージがキュー・マネージャーの誤動作によって影響を受けないよう保護するための手段として検討する必要があります。

## CLCACHE

使用するクラスター・キャッシュのタイプを指定します。詳細については、[236 ページの『キュー・マネージャー・クラスターの構成』](#)を参照してください。

### STATIC

クラスター・キャッシュが静的であるときには、そのサイズがキュー・マネージャーの開始時に固定されて、クラスター情報の現在量に、拡張のための余分のスペースを加えたサイズになります。キュー・マネージャーが活動状態であるときには、サイズを増やすことはできません。これがデフォルトです。

### 動的

クラスター・キャッシュが動的であるときには、キュー・マネージャーの開始時に割り振られた初期サイズを、キュー・マネージャーが活動状態であるときに必要に応じて自動的に増やすことができます。

## CMDUSER

コマンド・セキュリティ検査に使用されるデフォルトのユーザー ID を指定します。このユーザー ID は、ESM(例えば、RACF)に対して、定義されている必要があります。1 から 8 文字の英数字で名前を指定してください。最初の文字は英字でなければなりません。

デフォルトは CSQOPR です。

## CONNSWAP

特定の IBM MQ API 呼び出しを発行しているバッチ・ジョブが、その IBM MQ API 要求の実行期間中にスワップ可能かスワップ不能かを指定します。次のいずれかの値を指定します。

### NO

特定の IBM MQ API 呼び出しの実行期間中、ジョブはスワップ不能です。

### YES

すべての IBM MQ API 呼び出しの実行期間中、ジョブはスワップ可能です。

デフォルト値は YES です。

このパラメーターを使用すると、低優先度のジョブ保持している IBM MQ リソースを他のジョブまたはタスクが待機しているときに、低優先度のジョブがスワップアウトされます。

IBM MQ は、WebSphere Application Server を RRSBATCH 環境の一部と見なします。CONNSWAP キーワードを使用すると、BATCH 環境または RRSBATCH 環境のすべてのアプリケーションに適用されます。CONNSWAP キーワードも TSO ユーザーには適用されますが、CICS または IMS アプリケーション

には適用されません。CONNSWAP の変更は、キュー・マネージャーのリサイクルが行われるときに実装されます。キーワードの変更が行われた後にはリサイクルが必要です。これは、CSQ6SYSP マクロが再アSEMBルされ、キュー・マネージャーがマクロによって更新されたロード・モジュールを使用して再開したためです。

または、PPT を使用することによって、WebSphere Application Server アドレス・スペースをスワップ不能にすることもできます。

## EXCLMSG

除外するエラー・メッセージのリストを指定します。

このリストは動的であり、SET SYSTEM コマンドを使用して更新されます。

デフォルト値は空のリスト ( ) です。

メッセージは、CSQ 接頭部なし、アクション・コード接尾部 (I-D-E-A) なしで指定されます。例えば、メッセージ CSQX500I を除外するには、このリストに X500 を追加します。このリストには、最大 16 のメッセージ ID を含めることができます。

リストに含める対象にできるメッセージは、MSTR または CHIN アドレス・スペースの正常始動後に発行され、かつ、先頭文字が E、H、I、J、L、M、N、P、R、T、V、W、X、Y、2、3、5、9 のいずれかの文字であるものです。

コマンド処理結果として発行されるメッセージ ID をリストに追加できますが、除外されません。例えば、DISPLAY USAGE PSID(\*) コマンドの結果としてメッセージ ID が発行されますが、このメッセージを抑止することはできません。

## EXITLIM

キュー・マネージャー出口の呼び出しごとに許可される時間 (秒単位) を指定します。(このパラメーターはチャンネル出口には効果がありません。)

5 以上 9999 以下の範囲の値を指定します。

デフォルトは 30 です。キュー・マネージャーは実行中の出口を 30 秒ごとにポーリングします。ポーリングのたびに、EXITLIM で指定されている時間を超えて実行されていた出口は強制終了されます。

## EXITTCB

キュー・マネージャー内で出口の実行に使用する開始済みサーバー・タスクの数を指定します。(このパラメーターはチャンネル出口には効果がありません。) キュー・マネージャーが実行する可能性のある (チャンネル出口以外の) 出口の最大数と同じか、またはそれよりも大きな数を指定する必要があります。そのようにしない場合、それは 6c6 異常終了を出して失敗します。

0 ~ 99 の範囲の値を指定してください。値 0 は出口を実行できないことを意味します。

デフォルトは 8 です。

## LOGLOAD

1つのチェックポイントの開始から次のチェックポイントの開始までの間に IBM MQ が書き込むログ・レコードの数を指定します。IBM MQ は、指定した数のレコードが書き込まれた後で、新しいチェックポイントを開始します。

200 から 16 000 000 の範囲の値を指定してください。

デフォルトは 500 000 です。

この値が大きければ大きいほど、IBM MQ のパフォーマンスは向上します。ただし、このパラメーターを大きな値に設定すると、再始動の時間が長くなります。

推奨設定値:

テスト・システム	10 000
実動システム	500 000

実動システムで設定されているデフォルト値を使用すると、チェックポイントの出現が頻繁になりすぎる場合があります。

LOGLOAD の値により、キュー・マネージャー・チェックポイントの頻度が決まります。値が大きすぎると、チェックポイント間のログに書き込まれるデータの量が多くなり、その結果、障害後のキュー・マネージャーの順方向回復再始動時間が長くなります。値が小さすぎると、ピーク・ロード中にチェックポイントの出現が頻繁になりすぎ、応答時間およびプロセッサの使用に悪影響が出ます。

LOGLOAD の初期値として 500 000 を推奨します。1 KB の持続メッセージが 1 秒あたり 100 メッセージの速度で送信される場合 (つまり、コミット付きの MQPUT 100 回と、コミット付きの MQGET 100 回の場合)、チェックポイント間の間隔は約 5 分です。

**注:** ここで述べたことはガイドラインにすぎません。このパラメーターの最適値は、個々のシステムの特性に依存します。

## MULCCAPT

従量制ソフトウェア料金 (MULC) で使用されるデータの収集に使用されるアルゴリズムを指定します。

### STANDARD

MULC は、IBM MQAPI MQCONN 呼び出しから IBM MQ API MQDISC 呼び出しまでの時間に基づきます。

### REFINED

MULC は、IBM MQ API 呼び出しが開始してから IBM MQ API 呼び出しが終了するまでの時間に基づきます。

デフォルトは STANDARD です。

## OTMACON

OTMA パラメーター。このキーワードは、5 つの定位置パラメーターをとります。

### OTMACON = (Group, Member, Druexit, Age, Tpipepfx)

#### グループ

IBM MQ のこの特定インスタンスが属している XCF グループの名前です。

長さは 1 ~ 8 文字です。大文字で入力する必要があります。

デフォルトはブランクです。これは、IBM MQ が XCF グループに結合しないように指示します。

#### Member

XCF グループ内における IBM MQ のこの特定インスタンスのメンバー名です。

長さは 1 ~ 16 文字です。大文字で入力する必要があります。

デフォルトは 4 文字のキュー・マネージャー名です。

#### Druexit

IMS によって実行される OTMA 宛先解決ユーザー出口の名前を指定します。

長さは 1 ~ 8 文字です。

デフォルトは DFSYDRU0 です。

このパラメーターはオプションですが、IBM MQ によって開始されなかった IMS アプリケーションから IBM MQ がメッセージを受け取る場合は必須になります。この名前は、IMS システムでコーディングされた宛先解決ユーザー出口に対応している必要があります。詳しくは、[770 ページの『IMS における OTMA 出口の使用』](#)を参照してください。

#### Age

IBM MQ からのユーザー ID が以前に IMS によって検査されたと考えられる時間の長さ (秒単位) を表します。

範囲は 0 ~ 2 147 483 647 です。

デフォルトは 2 147 483 647 です。

メインフレーム内のセキュリティー・キャッシュ設定の整合性を保守するために、このパラメーターを ALTER SECURITY コマンドの interval パラメーターと共に設定することが勧められています。

## Tpipepfx

Tpipe 名に使用される接頭部を表します。

これは、3 文字で構成されます。一番目の文字は A から Z までの範囲です。後続の文字は A から Z、または 0 から 9 です。デフォルトは CSQ です。

これは IBM MQ が Tpipe を作成するときに使用されます。名前の残りの部分は IBM MQ によって割り当てられます。IBM MQ によって作成される Tpipe の名前全体を設定することはできません。

## QINDXBLD

キュー・マネージャーの再始動を、すべてのキュー索引が再作成されるまで待つか、すべての索引が再作成される前に完了するかを決定する。

### WAIT

キュー・マネージャーの再始動を、すべてのキュー索引作成が完了するまで待ちます。これは、索引が作成されている間は、通常の IBM MQ API 処理中に遅れが発生するアプリケーションがないことを意味します。アプリケーションがキュー・マネージャーに接続できるようになる前に、すべての索引が作成されるからです。

これがデフォルトです。

### NOWAIT

キュー・マネージャーは、すべてのキュー索引作成が完了する前に再始動できます。

## QMCCSID

キュー・マネージャーで(したがって、分散キューイングで)使用されるデフォルトのコード化文字セット ID を指定します。

0 から 65535 の範囲内で値を指定します。この値は、[各国語](#)で選択した言語のネイティブ z/OS コード・ページとしてリストされている EBCDIC コード・ページを表すものでなければなりません。

デフォルト値の 0 を指定すると、現在設定されている CCSID が使用されます。設定されているものがない場合は、CCSID 500 を使用します。つまり、CCSID をゼロ以外の値に明示的に設定している場合、それを QMCCSID をゼロに設定することによってリセットできません。その後、正しいゼロ以外の CCSID を使用しなければなりません。QMCCSID がゼロの場合、コマンド DISPLAY QMGR CCSID を発行して、実際に使用されている CCSID が何であるかを調べることができます。

## QSGDATA

キュー共有グループ・データ。このキーワードは、5 つの定位置パラメーターをとります。

**QSGDATA=(Qsgname , Dsgname , Db2name , Db2serv , Db2b1ob)**

### Qsgname

これは、キュー・マネージャーが属しているキュー共有グループの名前です。

有効な文字については、[IBM MQ オブジェクトの命名規則](#)を参照してください。名前には次の条件があります。

- 長さは 1 ～ 4 文字です
- 名前の先頭は数字であってはなりません
- @ で終わってはなりません。

これは、実装上の理由により、4 文字未満の名前は内部的に @ 記号が埋め込まれるためです。

デフォルトはブランクです。これはキュー・マネージャーがキュー共有グループのメンバーではないことを示します。

### Dsgname

これは、キュー・マネージャーが接続している Db2 データ共有グループの名前です。

長さは 1 ～ 8 文字です。大文字で入力する必要があります。

デフォルトはブランクです。これはキュー共有グループを使用しないことを示します。

### Db2name

これは、キュー・マネージャーが接続している Db2 サブシステムまたはグループの名前です。

長さは 1～4 文字です。大文字で入力する必要があります。

デフォルトは空白です。これはキュー共有グループを使用しないことを示します。

**注:** Db2 サブシステム (またはグループ接続) は、Dsgname で指定された Db2 データ共有グループになければなりません。また、すべてのキュー・マネージャーは同じ Db2 データ共有グループを指定しなければなりません。

#### **Db2serv**

これは、Db2 にアクセスするために使用するサーバー・タスクの数です。

範囲は 4～10 です。

デフォルトは 4 です。

#### **Db2blob**

これは、バイナリー・ラージ・オブジェクト (BLOB) にアクセスするために使用する Db2 タスクの数です。

範囲は 4～10 です。

デフォルトは 4 です。

名前パラメーターの 1 つ (つまり、Qsgname、Dsgname、または Db2name) を指定する場合は、他の名前の値を入力する必要があります。そうしないと、IBM MQ は失敗します。

#### **RESAUDIT**

接続処理中に実行される RESLEVEL セキュリティ検査のための RACF 監査レコードが作成されるかどうか指定します。

以下のいずれかを指定します。

##### **NO**

RESLEVEL 監査は実行されません。

##### **YES**

RESLEVEL 監査は実行されます。

デフォルトは「はい」です。

#### **ROUTCDE**

メッセージに割り当てられるデフォルトの z/OS メッセージ宛先コードを指定します。このメッセージは、MQSC コマンドへの直接的な応答として送られるものではありません。

以下のいずれかを指定します。

1. 1～16 の範囲の値 (1 と 16 を含む)。
2. 値のリスト (値をコンマで区切り、括弧で囲む)。それぞれの値は 1～16 までの範囲になければなりません (1 と 16 を含む)。

デフォルトは、1 です。

z/OS 宛先コードについて詳しくは、「*z/OS MVS Routing and Descriptor Codes*」資料のいずれかのボリュームにある「*Message description*」を参照してください。

#### **SERVICE**

このフィールドは IBM の使用のために予約されています。

#### **SMFACCT**

キュー・マネージャーの開始時に IBM MQ がアカウンティング・データを SMF に自動的に送信するかどうかを指定します。

以下のいずれかを指定します。

##### **NO**

アカウンティング・データの収集を自動的に開始しません。

##### **YES**

デフォルト・クラス 1 に対応するアカウンティング・データの収集を自動的に開始します。

**integers**

アカウントリングが自動的に収集されるクラスのリスト。範囲は 1 から 4 です。

デフォルトは NO です。

**SMFSTAT**

キュー・マネージャーが開始したとき、SMF 統計を自動的に収集するかどうかを指定します。

以下のいずれかを指定します。

**NO**

統計の収集を自動的に開始しません。

**YES**

デフォルト・クラス 1 に対応する統計の収集を自動的に開始します。

**integers**

統計が自動的に収集されるクラスのリスト。範囲は 1 から 4 です。

デフォルトは NO です。

**SPLCAP**

セキュリティー・ポリシー機能は、キューに対するメッセージの書き込みおよび読み取り時にメッセージへの署名(または暗号化)を行うかどうかを制御するポリシーを介在させることにより、メッセージ・セキュリティーのレベルを高めます。

この使用については、別個にインストールされる製品 Advanced Message Security (AMS) によってライセンスが与えられます。この製品が SDRQAUTH ライブラリーに使用可能化モジュールを提供するからです。

セキュリティー・ポリシー処理は、以下のいずれかの値で SPLCAP を構成することにより、このキュー・マネージャーで使用可能になります。

**NO**

キューのためのメッセージ・セキュリティー・ポリシーを実装する機能は、キュー・マネージャーの初期化時に有効になりません。

**YES**

キュー・マネージャーの初期化時にメッセージ・セキュリティー機能が有効になります。

この制御を設定すると、キュー・マネージャーは初期化時に SDRQAUTH からライセンス使用可能化モジュールをロードして、追加のアドレス・スペース (AMSM) を開始しようとします。

AMS にライセンスが交付されていて、メッセージ・セキュリティーに必要な構成が整っていなければ、キュー・マネージャーは開始されません。

デフォルトは NO です。

**STATIME**

統計の連続収集間のデフォルトの時間間隔(分単位)を指定します。

0 から 1440 の範囲の数を指定します。

値ゼロを指定する場合、統計データとアカウントリング・データの両方が SMF データ収集ブロードキャスト時に収集されます。この設定については、[システム管理機能の使用](#)を参照してください。

デフォルトは 30 です。

**TRACSTR**

グローバル・トレースを自動的に開始するかどうかを指定します。

以下のいずれかを指定します。

**NO**

グローバル・トレースを自動的に開始しません。

**YES**

デフォルト・クラス(クラス 1)に対応するグローバル・トレースを自動的に開始します。

**integers**

グローバル・トレースが自動的に開始されるクラスのリストです。範囲は 1 ~ 4 です。

\*

すべてのクラスについて、グローバル・トレースを自動的に開始します。

マクロ内にキーワードを指定していない場合、デフォルトは NO です。

注: 提供されているデフォルトのシステム・パラメーター・ロード・モジュール (CSQZPARM) では TRACSTR=YES です (アセンブラー・モジュール CSQFSYSP で設定されています)。トレースを自動的に開始させたくない場合は、ユーザー独自のシステム・パラメーター・モジュールを作成するか、キュー・マネージャーが開始したあとで STOP TRACE コマンドを実行します。

STOP TRACE コマンドについて詳しくは、[STOP TRACE](#) を参照してください。

#### TRACTBL

グローバル・トレース機能により IBM MQ トレース・レコードが保管されるトレース表のサイズを 4 KB ブロック単位で指定します。

1～999 の範囲の値を指定してください。

デフォルトは 99 です。これは 396 KB に相当します。

注: トレース表のストレージは ECSA に割り振られます。したがって、この値の選択は慎重に行ってください。

#### WLMTIME

WLM 管理対象キューの索引を走査する時間間隔 (WLMTIMU の値によって、分単位または秒単位) を指定します。

1 から 9999 の範囲内で値を指定します。

デフォルトは 30 です。

#### WLMTIMU

WLMTIME パラメーターで使用する時間単位。

次のうちの 1 つを指定します。

#### MINS

WLMTIME は分数を表します。

#### SECS

WLMTIME は秒数を表します。

デフォルトは MINS です。

#### 関連資料

[690 ページの『CSQ6LOGP の使用』](#)

このトピックは、CSQ6LOGP を使用したロギング・オプションの指定方法の参照として使用します。

[694 ページの『CSQ6ARVP の使用』](#)

このトピックは、CSQ6ARVP を使用したアーカイブ環境の指定方法の参照として使用します。

#### CSQ6LOGP の使用

このトピックは、CSQ6LOGP を使用したロギング・オプションの指定方法の参照として使用します。

CSQ6LOGP を使用して、ロギング・オプションを設定します。

CSQ6LOGP のデフォルト・パラメーター、および [ログの設定コマンド](#) を使用して、各パラメーターを変更できるかどうかは、[CSQ6LOGP パラメーターのデフォルト値](#) に示されています。これらの値を変更する必要がある場合は、該当するパラメーターの詳細記述を参照してください。

パラメーター	説明	デフォルト値	SET コマンド
<a href="#">COMPLOG</a>	ログ圧縮を使用可能にするかどうかを制御します。	NONE	X
<a href="#">DEALLCT</a>	保存テープ装置が割り振り解除される前に未使用状態になっている時間の長さ	ゼロ	X

表 45. CSQ6LOGP パラメーターのデフォルト値 (続き)

パラメーター	説明	デフォルト値	SET コマンド
<u>INBUFF</u>	アクティブ・ログ・データ・セットおよびアーカイブ・ログ・データ・セットの入力バッファ・ストレージのサイズ	60 KB	-
<u>MAXARCH</u>	記録できるアーカイブ・ログ・ボリュームの最大数	500	X
<u>MAXCNOFF</u>	並行して実行できる CSQJOFF7 オフロード・タスクの最大数。	31	-
<u>MAXRTU</u>	読み取りアーカイブ・ログ・テープ・ボリュームに同時に割り振られる専用テープ装置の最大数	2	X
<u>OFFLOAD</u>	保存のオン/オフ	YES (ON)	-
<u>OUTBUFF</u>	アクティブ・ログ・データ・セットおよびアーカイブ・ログ・データ・セットの出力バッファ・ストレージのサイズ	4 000 KB	-
<u>TWOACTV</u>	単一または重複の活動ロギング	YES (重複)	-
<u>TWOARCH</u>	単一または重複の保存ロギング	YES (重複)	-
<u>TWOBSDS</u>	単一または重複の BS DS	YES (重複 BS DS)	-
<u>WRTHRS</u>	アクティブ・ログ・データ・セットに書き込まれる前に、満杯になる出力バッファの数	20	X
<u>ZHYWRITE</u>	zHyperWrite フィーチャーを有効にするかどうかを指定します。	No	-

#### COMPLOG

ログ圧縮を使用可能にするかどうかを指定します。

次のどちらかを指定します。

##### NONE

ログ圧縮は使用可能にされません。

##### RLE

ログ圧縮は、ラン・レングス・エンコードを使用して使用可能にされます。

##### ANY

キュー・マネージャーは、最高度のログ・レコード圧縮を提供する圧縮アルゴリズムを選択します。このオプションを指定すると、RLE 圧縮が行われます。

デフォルトは NONE です。

ログ圧縮について詳しくは、[ログ圧縮](#)を参照してください。

#### DEALLCT

保存読み取りテープ装置が割り振り解除される前に未使用状態になることができる時間の長さ (分単位) を指定します。

次の 1 つを指定します。

- 0 ~ 1440 の範囲の時間 (分単位)。
- NOLIMIT

1440 または NOLIMIT を指定すると、テープ装置が割り振り解除されないことを意味します。

デフォルト値は 0 です。

アーカイブ・ログ・データがテープから読みとられる際に、IBM MQ が複数の読み取りアプリケーション用にテープ処理を最適化できるように、十分な大きさの値を設定するようにお勧めします。

### INBUFF

回復時にアクティブ・ログおよびアーカイブ・ログを読み込むための入力バッファのサイズを K バイト単位で指定します。28～60 の範囲の 10 進数を使用してください。指定した数は、4 の倍数に切り上げられます。

デフォルトは 60 KB です。

推奨設定値:

テスト・システム            28 KB

実動システム                60 KB

最良のパフォーマンスでログ読み取りを行えるように、この値を最大に設定します。

### MAXARCH

BSDS に記録できる保存ログ・ボリュームの最大数を指定します。この数を超えると、再び BSDS の開始位置から記録が始まります。

10～1000 の範囲の 10 進数を使用してください。

デフォルトは 500 です。

推奨設定値:

テスト・システム            500 (デフォルト)

実動システム                1 000

できるだけ多くのログを BSDS に記録できるように、この値を最大に設定します。

ログおよび BSDS について詳しくは、[IBM MQ リソースの管理](#)を参照してください。

### MAXCNOFF

並行して実行できる CSQJOFF7 オフロード・タスクの数を指定します。

これにより、使用可能なテープ装置をすべて使用してしまうことがないように、キュー・マネージャー (複数可) を調整することができます。

キュー・マネージャーは、CSQJOFF7 オフロード・タスクが完了するまで待機してから、新たに保存データ・セットの割り振りを試みます。

キュー・マネージャーがテープに保存する場合は、同時になされるテープ要求の数が、使用可能なテープ装置の数以上にならないように、このパラメーターを設定してください。そのようにしないと、システムがハングする可能性があります。

なお、重複保存を使用している場合は、オフロード・タスクごとに両方の保存を実行するため、それに合わせてこのパラメーターを設定する必要があります。例えば、キュー・マネージャーがテープへの重複保存をしている場合は、値を MAXCNOFF=2 にすると、最大で 2 つのアクティブ・ログを 4 つのテープに同時に保存することができます。

複数のキュー・マネージャーがテープ装置を共有している場合は、キュー・マネージャーごとに MAXCNOFF を適宜設定する必要があります。

デフォルト値は 31 です。

1～31 の範囲の値を指定してください。

### MAXRTU

読み取りアーカイブ・ログ・テープ・ボリュームに同時に割り振られる専用テープ装置の最大数を指定します。

このパラメーターと DEALLCT パラメーターを指定すると、IBM MQ はテープ装置からのアーカイブ・ログの読み取りを最適化することができます。

1～99 の範囲の値を指定してください。

デフォルトは 2 です。

値は IBM MQ に使用可能なテープ装置の数より、少なくとも 1 つ小さく設定するようにお勧めします。そうしないと、オフロード・プロセスの遅延が起こり、システムのパフォーマンスに影響を与える可能性があります。アーカイブ・ログ処理中にスループットを最大にするには、このオプションに設定できる最大値を指定します。ただし、オフロード・プロセス用に少なくとも 1 つのテープ装置が必要であることを覚えておいてください。

#### OFFLOAD

アーカイブのオン/オフを指定します。

次のどちらかを指定します。

##### YES

保存はオン

##### NO

保存はオフ

デフォルトは「はい」です。

**注意:** テスト環境で作業している場合以外は、保存をオフに切り替えないでください。保存をオフにすると、システムまたはトランザクションの障害が発生した場合に、データを回復できないことがあります。

#### OUTBUFF

アクティブ・ログ・データ・セットおよびアーカイブ・ログ・データ・セットを書き込むための出力バッファ用に IBM MQ が使用するストレージのサイズの合計を K バイト単位で指定します。出力バッファはそれぞれ 4 KB です。

このパラメーターは 128～4000 の範囲でなければなりません。指定した数は、4 の倍数に切り上げられます。40 から 128 までの値は、互換性の理由により受け入れられ、値 128 として扱われます。

デフォルトは 4000 KB です。

推奨設定値:

テスト・システム	400 KB
実動システム	4 000 KB

ログ出力バッファが容量不足にならないように、この値を最大に設定します。

#### TWOACTV

単一または重複の活動ロギングを指定します。

次のどちらかを指定します。

##### NO

単一アクティブ・ログ

##### YES

重複アクティブ・ログ

デフォルトは「はい」です。

単一ロギングおよび重複ロギングの使用方法については、[IBM MQ リソースの管理](#)を参照してください。

#### TWOARCH

アクティブ・ログがオフロードされる場合に IBM MQ が作成するアーカイブ・ログの数を指定します。

次のどちらかを指定します。

**NO**

単一アーカイブ・ログ

**YES**

重複アーカイブ・ログ

デフォルトは「はい」です。

推奨設定値:

テスト・システム	NO
実動システム	YES (デフォルト)

単一ロギングおよび重複ロギングの使用方法については、[IBM MQ リソースの管理](#)を参照してください。

**TWOBSDS**

ブートストラップ・データ・セットの数を指定します。

次のどちらかを指定します。

**NO**

単一 BS DS

**YES**

重複 BS DS

デフォルトは「はい」です。

単一ロギングおよび重複ロギングの使用方法については、[IBM MQ リソースの管理](#)を参照してください。

**WRTHRSH**

活動ログ・データ・セットに書き込まれる前に満杯になる 4 KB 出力バッファの数を指定します。

バッファの数が多ければ、書き込みの回数が少なくなり、IBM MQ のパフォーマンスが向上します。コミット点などのような重要なイベントが発生した場合は、この数に達する前にバッファが書き込まれることがあります。

1 ~ 256 の範囲のバッファ数を指定します。

デフォルトは 20 です。

**ZHYWRITE**

zHyperWrite フィーチャーを有効にするかどうかを指定します。

値は次のいずれかです。

**NO**

zHyperWrite は無効です。



**重要:** IBM MQ 9.0 で zHyperWrite は有効ではないので、許可される値は NO のみです。

**関連資料**

682 ページの『[CSQ6SYSP の使用](#)』

このトピックは、CSQ6SYSP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

694 ページの『[CSQ6ARVP の使用](#)』

このトピックは、CSQ6ARVP を使用したアーカイブ環境の指定方法の参照として使用します。

 **z/OS** CSQ6ARVP の使用

このトピックは、CSQ6ARVP を使用したアーカイブ環境の指定方法の参照として使用します。

CSQ6ARVP を使用して、保存環境を設定します。

CSQ6ARVP のデフォルト・パラメーターと、SET ARCHIVE コマンドを使用して各パラメーターを変更できるかどうかについては、695 ページの表 46 に示されています。これらの値を変更する必要がある場合は、該当するパラメーターの詳細記述を参照してください。ストレージの計画について詳しくは、z/OS を参照してください。

表 46. CSQ6ARVP パラメーターのデフォルト値			
パラメーター	説明	デフォルト値	SET コマンド
ALCUNIT	1 次および 2 次のスペース割り振りが行われる単位	BLK (ブロック数)	X
ARCPFX1	最初のアーカイブ・ログ・データ・セット名の接頭部	CSQARC1	X
ARCPFX2	2 番目のアーカイブ・ログ・データ・セット名の接頭部	CSQARC2	X
ARCRETN	アーカイブ・ログ・データ・セットの保存期間 (日数)	9999	X
ARCWRTC	アーカイブ・ログ・データ・セットに関するオペレーターあてメッセージの宛先コードのリスト	1、3、4	X
ARCWTOR	アーカイブ・ログ・データ・セットの取り付けを試みる前に、オペレーターにメッセージを送って応答を待つかどうか	YES	X
BLKSIZE	アーカイブ・ログ・データ・セットのブロック・サイズ	28 672	X
CATALOG	アーカイブ・ログ・データ・セットを ICF にカタログ化するか	NO	X
COMPACT	アーカイブ・ログ・データ・セットを圧縮するか	NO	X
PRIQTY	DASD データ・セットの 1 次スペース割り振り	25 715	X
PROTECT	データ・セットが作成される場合、ESM プロファイルによってアーカイブ・ログ・データ・セットが保護されるかどうか	NO	X
QUIESCE	MODE(QUIESCE) を指定して ARCHIVE LOG を出す場合に静止状態が許可される最大時間 (秒単位)	5	X
SECQTY	DASD データ・セットの 2 次スペース割り振り。使用される単位については、ALCUNIT パラメーターの項を参照してください。	540	X
TSTAMP	保存データ・セット名にタイム・スタンプを設定するか	NO	X
単位	アーカイブ・ログ・データ・セットの最初のコピーが保管される装置タイプまたは装置名	TAPE	X
UNIT2	アーカイブ・ログ・データ・セットの 2 番目のコピーが保管される装置タイプまたは装置名	ブランク	X

#### ALCUNIT

1 次および 2 次のスペース割り振りが行われる単位を指定します。

以下のいずれかを指定します。

#### CYL

シリンダー

## TRK

トラック

## BLK

ブロック

BLK は装置タイプに依存しないので、BLK を使用することをお勧めします。

デフォルトは BLK です。

保存 DASD ボリュームのフリー・スペースがフラグメント化している可能性がある場合は、より小さい 1 次エクステントを指定して、2 次エクステントへ拡張できるようにすることをお勧めします。アクティブ・ログのスペース割り振りについては、[ログ・アーカイブ・ストレージの計画](#)を参照してください。

## ARCPFX1

最初の保存ログ・データ・セット名の接頭部を指定します。

データ・セット名の設定方法および ARCPFX1 の長さの制限については、TSTAMP パラメーターの項を参照してください。

このパラメーターをブランクにすることはできません。

デフォルトは CSQARC1 です。

この接頭部の付いたアーカイブ・ログを作成する許可を、IBM MQ キュー・マネージャーのアドレス・スペースに関連付けられたユーザー ID に与えることが必要な場合があります。

## ARCPFX2

2 番目の保存ログ・データ・セット名の接頭部を指定します。

データ・セット名の設定方法および ARCPFX2 の長さの制限については、TSTAMP パラメーターの項を参照してください。

このパラメーターは、TWOARCH パラメーターに NO を指定した場合でも、ブランクにすることはできません。

デフォルトは CSQARC2 です。

この接頭部の付いたアーカイブ・ログを作成する許可を、IBM MQ キュー・マネージャーのアドレス・スペースに関連付けられたユーザー ID に与えることが必要な場合があります。

## ARCRETN

アーカイブ・ログ・データ・セットが作成される場合に使用される保存期間を日数で指定します。

このパラメーターは 0 ～ 9999 の範囲でなければなりません。

デフォルトは 9999 です。

推奨設定値:

テスト・システム            3

テスト・システムでは、長期間のアーカイブ・ログは必要ありません。

実動システム                9 999 (デフォルト)

アーカイブ・ログ自動削除を効率的にオフに切り替えできるように、この値を高く設定します。

アーカイブ・ログ・データ・セットの廃棄については、[保存ログ・データ・セットの廃棄](#)を参照してください。

## ARCWRTC

アーカイブ・ログ・データ・セットに関するオペレーター向けメッセージの z/OS 宛先コードのリストを指定します。ARCWTOR が NO に設定されている場合は、このフィールドは無視されます。

それぞれ 1 以上 16 以下の範囲の値の宛先コードを、14 個まで指定できます。少なくとも 1 つのコードを指定する必要があります。コードとコードの間は、ブランクではなく、コンマで区切ります。

デフォルトは 1、3、4 という値のリストです。

z/OS 宛先コードについて詳しくは、「z/OS MVS Routing and Descriptor Codes」資料のいずれかのボリュームにある「[Message description](#)」を参照してください。

#### ARCWTOR

アーカイブ・ログ・データ・セットの取り付けを試みる前に、オペレーターにメッセージを送って応答を受信するかどうかを指定します。

その他の IBM MQ ユーザーは、データ・セットがマウントされるまで強制的に待機させられることがあります。IBM MQ がメッセージへの応答を待機している間は影響を受けません。

次のどちらかを指定します。

#### YES

装置は、保存ログ・データ・セットが取り付けられるまで、長い時間を必要とします。例えば、テープ装置がこれに該当します。

#### NO

装置は長時間の遅延を必要としません。例えば、DASD がこれに該当します。

デフォルトは「はい」です。

推奨設定値:

テスト・システム NO

実動システム YES (デフォルト)

これは操作手順に依存します。テープ・ロボットを使用する場合は、NO のほうが適切です。

#### BLKSIZE

保存ログ・データ・セットのブロック・サイズを指定します。指定するブロック・サイズは、UNIT パラメーターで指定する装置タイプと互換性がなければなりません。

このパラメーターは 4 097 から 28 672 の範囲でなければなりません。指定した値は 4 096 の倍数に切り上げられます。

デフォルトは 28 672 です。

このパラメーターは、提供されている場合、ストレージ管理サブシステム (SMS) データ・クラスのブロック・サイズによって指定変更されます。

アーカイブ・ログ・データ・セットが DASD に書き込まれる場合は、トラックごとに 2 ブロックが許容される最大ブロック・サイズを選択することをお勧めします。例えば、3390 装置の場合、24 576 のブロック・サイズを使用します。

アーカイブ・ログ・データ・セットをテープに書き込む場合、可能な最大のブロック・サイズを指定すると、アーカイブ・ログの読み取り速度が向上します。ブロック・サイズとして 28 672 を使用してください。

推奨設定値:

テスト・システム アーカイブ・ログのために使用するメディアに応じたブロック・サイズの推奨値を使用します。

つまり、ディスクの場合は 24 576、およびテープの場合は 28 672 です。

実動システム アーカイブ・ログのために使用するメディアに応じたブロック・サイズの推奨値を使用します。

つまり、ディスクの場合は 24 576、およびテープの場合は 28 672 です。

#### CATALOG

アーカイブ・ログ・データ・セットを 1 次統合カタログ機能 (ICF) カタログにカタログ化するかどうかを指定します。

次のどちらかを指定します。

#### NO

アーカイブ・ログ・データ・セットはカタログ化されません。

#### YES

アーカイブ・ログ・データ・セットはカタログ化されます。

デフォルトは NO です。

DASD に割り振られているすべてのアーカイブ・ログ・データ・セットを、カタログ化する必要があります。CATALOG パラメーターを NO に設定して DASD に保存すると、アーカイブ・ログ・データ・セットが割り振られるたびにメッセージ [CSQJ072E](#) が表示され、IBM MQ によってデータ・セットがカタログされます。

推奨設定値:

テスト・システム	YES
実動システム	YES (アーカイブが DASD に割り振られる場合)

### COMPACT

保存ログに書き込まれたデータを圧縮するかどうかを指定します。このオプションは、改良データ記録機能 (IDRC) を備えた 3480 または 3490 装置だけに適用されます。この機能がオンになっていると、テープ制御装置のハードウェアは通常よりかなり高い密度でデータを書き込むため、1つのボリュームにより多くのデータを記録することができます。3480 装置に IDRC 機能または 3490 基本モデル (ただし 3490E は除く) が装備されている場合は、NO を指定します。データを圧縮する場合は、YES を指定します。

次のどちらかを指定します。

#### NO

データ・セットを圧縮しません。

#### YES

データ・セットを圧縮します。

デフォルトは NO です。

YES を指定すると、パフォーマンスに悪影響を与えます。また、テープに圧縮されたデータは IDRC 機能をサポートする装置を使用しなければ読めないことも念頭に置いてください。保存テープをリモート回復のために別のサイトに送る必要がある場合は、これも考慮する必要があります。

推奨設定値:

テスト・システム	適用外
実動システム	NO (デフォルト)

これは 3480 および 3490 IDR 圧縮だけに適用されます。YES に設定すると、回復および再始動時のアーカイブ・ログの読み取りパフォーマンスが低下する可能性があります、テープへの書き込みには影響しません。

### PRIQTY

DASD データ・セットの 1 次スペース割り振りを ALCUNIT の単位で指定します。

値はゼロより大きくなければなりません。

デフォルトは 25 715 です。

ログ・データ・セットまたはそれに対応する BSDS のどちらか大きい方をコピーする場合は、この値で十分です。必要な値を求めるには、次の手順に従ってください。

1. 678 ページの『ブートストラップ・データ・セットおよびログ・データ・セットを作成する』で説明されているように、割り振られているアクティブ・ログ・レコードの数 (c) を判別します。
2. 次のように、各アーカイブ・ログ・ブロックの中の 4096 バイト・ブロックの数を決定します。

$$d = \text{BLKSIZE} / 4096$$

ここで、BLKSIZE は切り上げた値です。

### 3. ALCUNIT=BLK の場合:

$$\text{PRIQTY} = \text{INT}(c / d) + 1$$

ここで、INT は整数まで端数を切り捨てた値です。

### ALCUNIT=TRK の場合:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}))) + 1$$

ここで、e はトラックごとのバイト数 (3390 装置の場合は 56664) であり、INT は整数まで端数を切り捨てた値です。

### ALCUNIT=CYL の場合:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}) * f)) + 1$$

ここで、f はシリンダーごとのトラック数 (3390 装置の場合は 15) であり、INT は整数まで端数を切り捨てた値です。

ログ・データ・セットおよびアーカイブ・ログ・データ・セットの大きさについては、[678 ページの『ブートストラップ・データ・セットおよびログ・データ・セットを作成する』](#) および [679 ページの『ページ・セットの定義』](#) を参照してください。

#### 推奨設定値:

テスト・システム 1 680

活動ログ全体を保持する場合は、以下の値で十分です。

$$10 \ 080 / 6 = 1 \ 680 \text{ blocks}$$

実動システム テープに保存する場合は適用されません。

保存 DASD ボリュームのフリー・スペースがフラグメント化している可能性がある場合は、より小さい 1 次エクステントを指定して、2 次エクステントへ拡張できるようにすることをお勧めします。アクティブ・ログのスペース割り振りについては、[z/OS での計画](#) を参照してください。

## PROTECT

保存ログ・データ・セットが作成される場合、離散 ESM (外部セキュリティー管理プログラム) プロファイルによってそのデータ・セットが保護されるかどうかを指定します。

次のどちらかを指定します。

### NO

プロファイルは作成されません。

### YES

ログをオフロードする場合、離散データ・セット・プロファイルが作成されます。YES を指定する場合は、以下の条件を満たしている必要があります。

- ESM 保護は IBM MQ に対して活動状態でなければならない。
- IBM MQ キュー・マネージャー・アドレス・スペースに関連しているユーザー ID に、これらのプロファイルを作成する権限が必要である。
- テープに保存している場合、TAPEVOL クラスは活動状態でなければならない。

そうでない場合、オフロードは失敗します。

デフォルトは NO です。

### QUIESCE

MODE (QUIESCE) を指定して ARCHIVE LOG コマンドを発行するときに、静止状態が許される最大時間 (秒数) を指定します。

このパラメーターは 1 ~ 999 の範囲でなければなりません。

デフォルトは 5 です。

### SECQTY

DASD データ・セットの 2 次スペース割り振りを ALCUNIT の単位で指定します。2 次エクステントは、15 回まで割り振ることができます。詳細は、「z/OS MVS JCL 解説書」および「z/OS MVS JCL ユーザーズ・ガイド」を参照してください。

パラメーターは 0 より大きくなければなりません。

デフォルトは 540 です。

### TSTAMP

保存ログ・データ・セット名にタイム・スタンプを含めるかどうかを指定します。

次のどちらかを指定します。

#### NO

名前にタイム・スタンプは入りません。保存ログ・データ・セットの名前は、次のように設定されます。

```
arcpxi.A nnnnnnn
```

この場合、*arcpxi* は、ARCPFX1 または ARCPFX2 で指定されているデータ・セット名接頭部です。*arcpxi* は、最大 35 文字を保持することができます。

#### YES

名前にタイム・スタンプを入れます。保存ログ・データ・セットの名前は、次のように設定されます。

```
arcpxi.cydd.T hhmsst.A nnnnnnn
```

この場合、*c* は、1999 年までは「D」、2000 年以降は「E」になります。*arcpxi* は、ARCPFX1 または ARCPFX2 で指定されているデータ・セット名接頭部です。*arcpxi* は、最大 19 文字を保持することができます。

#### EXT

名前にタイム・スタンプを入れます。保存ログ・データ・セットの名前は、次のように設定されます。

```
arcpxi.D yyyydd.T hhmsst.A nnnnnnn
```

この場合、*arcpxi* は、ARCPFX1 または ARCPFX2 で指定されているデータ・セット名接頭部です。*arcpxi* は、最大 17 文字を保持することができます。

デフォルトは NO です。

## 単位

保存ログ・データ・セットの最初のコピーの保管に使用する装置の装置タイプまたは装置名を指定します。

装置タイプまたは装置名は 1 ～ 8 文字の英数字で指定します。最初の文字は英字でなければなりません。

このパラメーターをブランクにすることはできません。

デフォルトは TAPE です。

DASD にアーカイブする場合は、総称装置タイプと限定するボリューム範囲を指定することができます。例: UNIT=3390。

DASD に保存する場合は、次の点に注意してください。

- 1 次スペース割り振りは、アクティブ・ログ・データ・セットのすべてのデータを入れるのに十分かどうか。
- アーカイブ・ログ・データ・セットのカatalog・オプション (CATALOG) が YES に設定されているかどうか。
- BLKSIZE に対して適切な値が使用されているかどうか。

テープに保存する場合、IBM MQ は最大 20 ボリュームまで拡張できます。

推奨設定値:

テスト・システム	DASD
実動システム	TAPE

アーカイブ・ログのロケーションの選択について詳しくは、[z/OS での計画](#)を参照してください。

## UNIT2

アーカイブ・ログ・データ・セットの 2 番目のコピーの保管に使用する装置の装置タイプまたは装置名を指定します。

装置タイプまたは装置名は 1 ～ 8 文字の英数字で指定します。最初の文字は英字でなければなりません。このパラメーターがブランクの場合は、UNIT パラメーターの値が使用されます。

デフォルトはブランクです。

## 関連資料

682 ページの『[CSQ6SYSP の使用](#)』

このトピックは、CSQ6SYSP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

690 ページの『[CSQ6LOGP の使用](#)』

このトピックは、CSQ6LOGP を使用したロギング・オプションの指定方法の参照として使用します。

 [CSQ6USGP の使用](#)

このトピックは、CSQ6USGP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

CSQ6USGP を使用して、製品の使用記録を制御します。

CSQ6USGP のデフォルトのパラメーターは、[701 ページの表 47](#) に示すとおりです。これらの値を変更する必要がある場合は、該当するパラメーターの詳細記述を参照してください。



**重要:** SET SYSTEM コマンドを使用してこれらのパラメーターを変更することはできません。

パラメーター	説明	デフォルト値
QMGRPROD	キュー・マネージャーの使用が記録される製品	ブランク

表 47. CSQ6USGP パラメーターのデフォルト値 (続き)		
パラメーター	説明	デフォルト値
AMSPROD	Advanced Message Security (AMS) の使用が記録される製品	ブランク

### QMGRPROD

キュー・マネージャーの使用が記録される製品を指定します。

以下のいずれかを指定します。

#### MQ

キュー・マネージャーの使用がスタンドアロン IBM MQ for z/OS 製品として、製品 ID 5655-MQ9 で記録されます。

#### VUE

キュー・マネージャーの使用がスタンドアロン IBM MQ for z/OS Value Unit Edition (VUE) 製品として、製品 ID 5655-VU9 で記録されます。

#### ADVANCEDVUE

キュー・マネージャーの使用が IBM MQ Advanced for z/OS, Value Unit Edition 製品の一部として、製品 ID 5655-AV1 で記録されます。

### AMSPROD

AMS の使用が記録される製品を指定します (使用される場合)。

以下のいずれかを指定します。

#### アームズ

AMS の使用がスタンドアロン Advanced Message Security for z/OS 製品として、製品 ID 5655-AM9 で記録されます。

#### ADVANCED

AMS の使用が IBM MQ Advanced for z/OS 製品の一部として、製品 ID 5655-AV9 で記録されます。

#### ADVANCEDVUE

AMS の使用が IBM MQ Advanced for z/OS, Value Unit Edition 製品の一部として、製品 ID 5655-AV1 で記録されます。

製品使用記録の詳細については、[製品情報のレポート](#)を参照してください。

### 関連資料

#### 682 ページの『CSQ6SYSP の使用』

このトピックは、CSQ6SYSP を使用したシステム・パラメーターの設定方法のリファレンスとして使用してください。

#### 690 ページの『CSQ6LOGP の使用』

このトピックは、CSQ6LOGP を使用したロギング・オプションの指定方法の参照として使用します。

## チャンネル・イニシエーター・パラメーターを調整する

ALTER QMGR を使用して、要件に適合するようにチャンネル・イニシエーターをカスタマイズします。

- この作業は、必要に応じて IBM MQ キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行する必要があります。

多数のキュー・マネージャー属性によって、分散キューイングの操作方法が制御されます。MQSC コマンド ALTER QMGR を使用して、これらの属性を設定してください。初期設定データ・セットのサンプル thlqual.SCSQPROC(CSQ4INYG) には、カスタマイズできる設定が含まれています。詳しくは、[ALTER QMGR](#) を参照してください。

これらのパラメーターの値は、チャンネル・イニシエーターを開始するたびに一連のメッセージとして表示されます。

## アダプター、ディスパッチャー、およびチャンネルの最大数の間の関係

ALTER QMGR パラメーターの CHIADAPS および CHDISPS は、チャンネル・イニシエーターによって使用されるタスク制御ブロック (TCB) の数を定義します。CHIADAPS (アダプター) TCB は、キュー・マネージャーに対して IBM MQ API 呼び出しを行うために使用されます。CHDISPS (ディスパッチャー) TCB は、通信ネットワークに対して呼び出しを行うために使用されます。

ALTER QMGR パラメーターの MAXCHL は、ディスパッチャー TCB へのチャンネルの配分に影響を与えます。

### CHDISPS

チャンネルが少数の場合には、デフォルト値を使用してください。

各プロセッサにつきタスクを 1 つにすると、システム・パフォーマンスが最適化されます。ディスパッチャー・タスクでは CPU の使用量が高いため、原則として、ビジーになるタスクをできるだけ少ない状態にします。そうすることで、スレッドを見つけて開始するまでにかかる時間を短縮します。

チャンネル数が 100 を超えるシステムでは、CHDISPS(20) が適しています。必要以上のディスパッチャー TCB がある場合に CHDISPS(20) を使用しても、特に問題が生じることはありません。

指針として、1000 を超える数のチャンネルがある場合、50 の現行チャンネルごとに 1 つのディスパッチャーを配分してください。例えば、2000 までのアクティブ・チャンネルを処理するには、CHDISPS(40) を指定します。

TCP/IP を使用している場合、CHDISPS に 100 よりも大きい値を指定した場合でも、TCP/IP チャンネルに使用されるディスパッチャーの最大数は 100 となります。

### CHIADAPS

キュー・マネージャーへの各 IBM MQ API 呼び出しは、他のいずれとも独立していて、任意のアダプター TCB に対して行うことができます。持続メッセージを使用する呼び出しは、ログ入出力のために非持続メッセージの場合よりも時間がかかることがあります。したがって、多くのチャンネルで多数の持続メッセージを処理するチャンネル・イニシエーターは、最適なパフォーマンスを得るために、デフォルトの 8 個のアダプター TCB を必要とする場合があります。特にそうなるのは、作成されたバッチ・サイズが小さい場合 (各バッチ処理の最後にもログ I/O が必要となるため)、およびシン・クライアント・チャンネルが使用される場合です。

実稼働環境のための推奨値は、CHIADAPS(30) です。これより大きな値を指定しても特に利点はなく、必要以上のアダプター TCB がある場合に CHIADAPS(30) を指定しても特に問題が生じることはありません。

### MAXCHL

各チャンネルはチャンネル開始時に特定のディスパッチャー TCB に関連付けられて、そのチャンネルが停止するまでその TCB に関連付けられたままとなります。多数のチャンネルで、それぞれの TCB を共有できます。MAXCHL を使用すると、使用可能なディスパッチャー TCB にチャンネルを展開します。開始する最初の ( $\text{MIN}(\text{MAXCHL} / \text{CHDISPS}), 10$ ) チャンネルは最初のディスパッチャー TCB に関連付けられて、すべてのディスパッチャー TCB が使用されるまで同様に関連付けられます。

このため、チャンネルが少数で、MAXCHL が大きい値の場合には、チャンネルはディスパッチャーに対して均等に配分されません。例えば、CHDISPS(10) を設定して MAXCHL がデフォルト値の 200 のままの場合に、チャンネル数が 50 だけであれば、5 つのディスパッチャーはそれぞれ 10 のチャンネルと関連付けられて、5 つのディスパッチャーは使用されません。実際に使用するチャンネルの数が小さな固定数のときには、MAXCHL をその数に設定することをお勧めします。

このキュー・マネージャー・プロパティを変更する場合は、確実に値が互換であるようにするため、ACTCHL、LU62CHL、および TCPCHL キュー・マネージャー・プロパティも検討しなければなりません。これらのプロパティの完全な説明とその関係については、[キュー・マネージャーのパラメーター](#)を参照してください。

## チャンネル・イニシエーター用の z/OS UNIX システム・サービス環境のセットアップ

チャンネル・イニシエーター (CHINIT) は OMVS スレッドを使用します。OMVS 構成パラメーターを検討してから、新規 CHINIT を作成するか、ディスパッチャーまたは SSLTASKS の数を変更します。

各 CHINIT は、3 + CHIDISP + SSLTASKS OMVS スレッドを使用します。これらは、LPAR で使用される OMVS スレッドの総数、CHINIT 開始タスク・ユーザー ID により使用されるスレッドの数に入ります。

**D OMVS,L** を使用して、現行使用量、最高使用量、MAXPROCSYS のシステムしきい値 (システムが許容するプロセス最大数) を検討できます。

新規 CHINIT を追加するか CHIDISPS または SSLTASKS の値を増やした場合、スレッドの増加を計算して、MAXPROCSYS 値への影響を検討しなければなりません。SETOMVS コマンドを使用して、直接、MAXPROCSYS の変更、または BPXPRCxx parmlib 値の更新 (またはその両方) を行えます。

OMVS パラメーター MAXPROCUSER は、単一 OMVS ユーザー、つまり同じ UID が所有できる OMVS スレッドの数です。スレッドは、この値に数えられます。それで、同じ開始タスク・ユーザー ID の 2 つの CHINITs に、それぞれ 10 のディスパッチャーと 3 つの SSLTASKS がある場合、その OMVS uid には  $2 * (3 + 10 + 3) = 32$  スレッドがあることになります。

デフォルト MAXPROCUSER は **D OMVS,O** コマンドを発行して表示できますし、SETOMVS コマンドを使用して、直接、MAXPROCUSER の変更、または BPXPRCxx parmlib 値の更新 (またはその両方) を行えます。

この値は、RACF コマンド **ALTUSER userid OMVS(PROCUSERMAX(nnnn))** または同等のコマンドを使用して、ユーザーごとにオーバーライドできます。

チャンネル・イニシエーターを開始するには、次のコマンドを実行します。

```
START CHINIT
```

チャンネル・イニシエーターが正常に開始したことを確認するために、xxxxCHIN(ssidCHIN) ジョブ・ログに ICH408I エラーがないことを確認します。

## 関連概念

704 ページの『[バッチ、TSO、および RRS アダプターをセットアップする](#)』

ライブラリーを該当する STEPLIB 連結に追加して、アダプターをアプリケーションで使用できるようにします。アダプターによって発行されるスナップ・ダンプに対応するには、CSQSNAP DD 名を割り振ります。CSQBDEFV を使用してアプリケーション・プログラムの移植性を向上させることを検討してください。

## 関連情報

[チャンネル・イニシエーター統計データ・レコード](#)

## **バッチ、TSO、および RRS アダプターをセットアップする**

ライブラリーを該当する STEPLIB 連結に追加して、アダプターをアプリケーションで使用できるようにします。アダプターによって発行されるスナップ・ダンプに対応するには、CSQSNAP DD 名を割り振ります。CSQBDEFV を使用してアプリケーション・プログラムの移植性を向上させることを検討してください。

- この作業は、必要に応じて IBM MQ キュー・マネージャーごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

バッチ接続を使用するバッチおよびその他のアプリケーションでアダプターを使用できるようにするには、バッチ・アプリケーションの STEPLIB 連結に、次の IBM MQ ライブラリーを追加します。

- thlqual.SCSQANLx
- thlqual.SCSQAUTH

この場合、x は各国語の言語を示す文字です。(ライブラリーが LPA またはリンク・リストにある場合は、これを行う必要はありません。)

TSO アプリケーションの場合は、TSO ログオン・プロシージャ内でライブラリーを STEPLIB 連結に追加するか、TSO コマンドの TSOLIB を使用してライブラリーをアクティブ化します。

アダプターは、予期しない IBM MQ エラーを検出すると、DD 名 CSQSNAP に z/OS SNAP ダンプを発行し、アプリケーションに理由コード MQRC\_UNEXPECTED\_ERROR を発行します。CSQSNAP DD ステートメントがアプリケーション JCL にないか、または CSQSNAP が TSO の下のデータ・セットに割り振られていない場合は、ダンプはとられません。この場合、アプリケーション JCL に CSQSNAP DD ステートメントを含め

るか、または CSQSNAP を TSO の下のデータ・セットに割り振ってから、アプリケーションを再実行することができます。しかし、いくつかの問題が断続的に発生することがあるため、CSQSNAP ステートメントをアプリケーション JCL に含めるか、または CSQSNAP を TSO ログオン・プロシージャでデータ・セットに割り振って、問題の発生時に障害の理由を見極めることをお勧めします。

提供されているプログラム CSQBDEFV により、使用しているアプリケーション・プログラムの移植性が向上します。CSQBDEFV では、接続先のキュー・マネージャーまたはキュー共有グループの名前を指定でき、その名前をアプリケーション・プログラムの MQCONN または MQCONNX 呼び出しの中で指定せずに済みます。キュー・マネージャーまたはキュー共有グループごとに、新しいバージョンの CSQBDEFV を作成できます。そのためには、次の手順を実行してください。

1. IBM MQ アセンブラー・プログラム CSQBDEFV を、thlqual.SCSQASMS からユーザー・ライブラリーへコピーします。
2. 提供されたプログラムには、デフォルトのサブシステム名 CSQ1 が設定されます。テストおよびインストール検査では、この名前をそのまま使用できます。実動サブシステムの場合は、NAME=CSQ1 を 1～4 文字の独自のサブシステム名に変更することも、CSQ1 をそのまま使用することもできます。

キュー共有グループを使用する場合、CSQ1 ではなくキュー共有グループ名を指定することができます。これを行う場合、プログラムはそのグループ内の活動状態のキュー・マネージャーに接続要求を出します。

3. プログラムのアセンブルおよびリンク・エディットを実行し、CSQBDEFV ロード・モジュールを生成します。アセンブルするときには、ライブラリー thlqual.SCSQMACS を SYSLIB 連結に組み込みます。その場合は、リンク・エディット・パラメーター RENT,AMODE=31,RMODE=ANY を使用してください。これは thlqual.SCSQPROC(CSQ4DEFV) のサンプル JCL に示されています。次に、ロード・ライブラリーを thlqual.SCSQAUTH より前に z/OS バッチ、TSO STEPLIB に組み込みます。

## 関連概念

705 ページの『[操作パネルおよび制御パネルをセットアップする](#)』

操作パネルおよび制御パネルをセットアップするには、まず、必要なパネル、EXEC、メッセージ、および表が入ったライブラリーをセットアップする必要があります。そのためには、パネルに使用する各国語機能を考慮する必要があります。これを行った後、オプションで IBM MQ 操作パネルおよび制御パネル用のメイン ISPF メニューを更新し、ファンクション・キーの設定を変更することができます。

## **z/OS** 操作パネルおよび制御パネルをセットアップする

操作パネルおよび制御パネルをセットアップするには、まず、必要なパネル、EXEC、メッセージ、および表が入ったライブラリーをセットアップする必要があります。そのためには、パネルに使用する各国語機能を考慮する必要があります。これを行った後、オプションで IBM MQ 操作パネルおよび制御パネル用のメイン ISPF メニューを更新し、ファンクション・キーの設定を変更することができます。

- この作業は、IBM MQ を実行するそれぞれの z/OS システムごとに実行する必要があります。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

## ライブラリーのセットアップ

以下のステップを実行して、IBM MQ の操作パネルと制御パネルをセットアップします。

1. 連結に含まれるすべてのライブラリーは、同じ形式 (F、FB、V、VB) であり、かつ同じブロック・サイズであるか、降順ブロック・サイズの順番のどちらかにしてください。そのようにしないと、これらのパネルを使用しようとしたときに問題が発生することがあります。
2. ライブラリー thlqual.SCSQEXEC を SYSEXEC 連結または SYSPROC 連結に組み込むか、あるいは、TSO ALTLIB コマンドを使用してライブラリーをアクティブ化します。このライブラリーは、インストール時に固定ブロック 80 のレコード形式で割り振られており、必要な EXEC が設定されています。

ライブラリーは SYSEXEC 連結に入れることをお勧めします。ただし、ライブラリーを SYSPROC 連結に入れる場合は、ライブラリーは 80 バイトのレコード長でなければなりません。

3. thlqual.SCSQAUTH がリンク・リストまたは LPA に入っていない場合は、thlqual.SCSQAUTH および thlqual.SCSQANLx を TSO ログオン・プロシージャー STEPLIB に追加するか、TSO TSOLIB コマンドを使ってアクティブ化します。
4. IBM MQ パネル・ライブラリーを永続的に ISPF ライブラリー・セットアップに追加することも、パネルの使用時に動的にセットアップされるようにすることもできます。永続的にインストールするには、次の手順を実行する必要があります。
  - a. 操作パネルと制御パネルの定義が入っているライブラリーを ISPLLIB 連結に組み込みます。この名前は thlqual.SCSQPNLx です。この場合、x は各国語の言語を示す文字です。
  - b. 必要な表が入っているライブラリーを ISPTLIB 連結に組み込みます。この名前は thlqual.SCSQTBx です。この場合、x は各国語の言語を示す文字です。
  - c. 必要なメッセージが入っているライブラリーを ISPLMLIB 連結に組み込みます。この名前は thlqual.SCSQMSGx です。この場合、x は各国語の言語を示す文字です。
  - d. 必要なロード・モジュールが入っているライブラリーを ISPLLIB 連結に組み込みます。このライブラリーの名前は thlqual.SCSQAUTH です。
5. TSO コマンド処理プログラムのパネルから IBM MQ パネルへのアクセスが可能かどうかを検査します。これは通常、ISPF/PDF 基本オプション・メニューのオプション 6 です。実行する EXEC の名前は CSQOREXX です。ステップ 4 で IBM MQ ライブラリーを永続的に ISPF セットアップに入れた場合は、指定するパラメーターはありません。永続的にインストールしていない場合は、次のパラメーターを指定します。

```
CSQOREXX thlqual langletter
```

この場合、langletter は、使用する各国語を示す文字です。

- C** 中国語 (簡体字)
- E** U.S. 英語(大/小文字混合)
- F** フランス語
- K** 日本語
- U** U.S. 英語(大文字)

## ISPF メニューの更新

ISPF メインメニューを更新すると、ISPF から IBM MQ 操作パネルおよび制御パネルにアクセスすることができます。&ZSEL に必要な設定は、次のとおりです。

```
CMD(%CSQOREXX thlqual langletter)
```

thlqual および langletter についての詳細は、ステップ 706 ページの『5』を参照してください。

詳細については、「z/OS: ISPF ダイアログ開発者 ガイドとリファレンス」を参照してください。

## ファンクション・キーおよびコマンド設定値の更新

通常の ISPF プロシージャーを使用して、パネルで使用されるファンクション・キーおよびコマンド設定値を変更することができます。アプリケーション ID は CSQO です。

ただし、作成した変更内容を反映するようにヘルプ情報が更新されるわけではないので、この変更はお勧めしません。

## 関連概念

707 ページの『[IBM MQ ダンプ・フォーマット・メンバーを組み込む](#)』

対話式問題管理システム (IPCS) を使用して IBM MQ ダンプをフォーマットできるようにするには、いくつかのシステム・ライブラリーを更新する必要があります。

### **z/OS** IBM MQ ダンプ・フォーマット・メンバーを組み込む

対話式問題管理システム (IPCS) を使用して IBM MQ ダンプをフォーマットできるようにするには、いくつかのシステム・ライブラリーを更新する必要があります。

- この作業は、IBM MQ を実行するそれぞれの z/OS システムごとに実行する必要があります。
- 旧バージョンから移行する際に、この作業を実行する必要があります。

対話式問題管理システム (IPCS) を使用して IBM MQ ダンプをフォーマットできるようにするには、データ・セット `thlqual.SCSQPROC(CSQ7IPCS)` を `SYS1.PARMLIB` にコピーします。このデータ・セットを編集する必要はありません。

IPCS 用の TSO プロシージャをカスタマイズした場合は、`thlqual.SCSQPROC(CSQ7IPCS)` を IPCSPARM 定義内の任意のライブラリーにコピーすることができます。IPCSPARM の詳細については、「[z/OS MVS IPCS カスタマイズ](#)」を参照してください。

ライブラリー `thlqual.SCSQPNLA` も `ISPLLIB` 連結に取り込む必要があります。

ダンプ・フォーマット・プログラムを TSO セッションまたは IPCS ジョブで使用できるようにするには、ライブラリー `thlqual.SCSQAUTH` も `STEPLIB` 連結に組み込むか、`TSO TSOLIB` コマンドを使用して `thlqual.SCSQAUTH` をアクティブ化する必要があります (リンク・リストまたは LPA に既に存在する場合)。

## 関連概念

707 ページの『[通知メッセージを抑止する](#)』

IBM MQ システムでは、多数の通知メッセージが生成されることがあります。選択したメッセージがコンソールまたはハードコピー・ログに送信されないようにすることができます。

### **z/OS** 通知メッセージを抑止する

IBM MQ システムでは、多数の通知メッセージが生成されることがあります。選択したメッセージがコンソールまたはハードコピー・ログに送信されないようにすることができます。

- この作業は、IBM MQ を実行するそれぞれの z/OS システムごとに実行する必要があります。
- 旧バージョンから移行する際は、この作業を実行する必要はありません。

IBM MQ システムが大量に使用されている場合、多数のチャネルが停止して開始している場合は、多数の情報メッセージが z/OS コンソールおよびハードコピー・ログに送信されます。IBM MQ-IMS ブリッジとバッファ管理プログラムも、大量の通知メッセージを生成します。

必要な場合は、`SYS1.PARMLIB` の `MPFLSTxx` メンバーで指定されている z/OS メッセージ処理機能リストを使用して、これらのコンソール・メッセージの一部を抑止することができます。指定したメッセージは、ハードコピー・ログには書き込まれますが、コンソールには表示されません。

サンプル `thlqual.SCSQPROC(CSQ4MPFL)` に、`MPFLSTxx` の推奨設定値が示されています。MPFLSTxx について詳しくは、「[z/OS MVS 初期設定およびチューニング解説書](#)」を参照してください。

ハードコピー・ログの選択した通知メッセージを抑止する場合は、z/OS インストール・システム出口 `IEAVMXIT` を使用することができます。必要なメッセージに対して次のビット・スイッチを ON に設定できます。

## CTXTRDTM

メッセージを削除します。

このメッセージはコンソールに表示されたり、ハードコピーに記録されることがありません。

## CTXTESJL

ジョブ・ログから抑止します。

このメッセージは JES ジョブ・ログに入れられません。

## CTXTNWTP

WTP 処理を実行しません。

メッセージは TSO 端末またはバッチ・ジョブのシステム・メッセージ・データ・セットに送られません。

注:

1. その他のパラメーターの詳細については、[MVS インストール・システム出口](#)の資料を参照してください。
2. 提案されている抑止リスト CSQ4MPFL にあるもの以外のメッセージを抑止することは勧められていません。

さらに、パラメーターを追加指定することもできます。

## EXCLMSG

すべてのログから除外するメッセージのリストを指定します。

このリストにあるメッセージは、z/OS コンソールおよびハードコピー・ログに送られません。詳しくは、682 ページの『[CSQ6SYSP の使用](#)』の EXCLMSG を参照してください。

## 関連タスク

719 ページの『[z/OS でのキュー・マネージャーのテスト](#)』

キュー・マネージャーのカスタマイズまたは移行を済ませた後、IBM MQ for z/OS に添付されているインストール検査プログラムおよびサンプル・アプリケーションを実行することによって、キュー・マネージャーをテストできます。

## キュー共有グループの構成

高可用性のために共有キューを使用する場合は、キュー共有グループを構成するためのステップバイステップ・ガイドとしてこれらのトピックをご利用ください。

IBM MQ for z/OS システムのセットアップ処理のこの部分のステップを完了した後、679 ページの『[システム・パラメーター・モジュールの調整](#)』を行い、キュー共有グループ・データを追加する必要があります。CSQ6SYSP を変更して QSGDATA パラメーターを指定する必要があります。

## Db2 環境のセットアップ

キュー共有グループを使用する場合は、いくつかのサンプル・ジョブをカスタマイズして実行することで、必要な Db2 オブジェクトを作成する必要があります。

## Db2 環境のセットアップ

いくつかのサンプル・ジョブをカスタマイズして実行することで、必要な Db2 オブジェクトを作成し、バインドする必要があります。

- この作業は、Db2 データ共有グループごとに繰り返してください。
- 旧バージョンから移行する際に、bind および grant のステップを実行する必要があります。
- キュー共有グループを使用しない場合は、この作業を省略してください。

後にキュー共有グループを使用する必要性が生じた場合は、その時点でこのタスクを実行してください。

 IBM MQ には、同等のジョブ・セットが 2 つあります。接頭部が CSQ45 のジョブは、旧バージョンの IBM MQ との互換性を保持し、Db2 バージョン 11 以前で使用するために提供されています。Db2 V12 以降で新しいデータ共有グループをセットアップする場合は、接頭部が CSQ4X のジョブを使用することをお勧めします。これらのジョブは、動的なサイズ変更やユニバーサル表スペースに、より新しい Db2 機能を活用するためです。

IBM MQ がキュー共有グループのために使用する Db2 計画にアクセスし、実行できる環境を決定する必要があります。

次の手順は、新規 Db2 データ共有グループごとに実行する必要があります。すべてのサンプル JCL は thlqual.SCSQPROC にあります。

1. サンプル JCL CSQ45CSG **V 9.0.4** (または CSQ4XCSG) を、IBM MQ データベース、表スペース、および表に使用されるストレージ・グループを作成するようにカスタマイズして、それを実行します。
2. サンプル JCL CSQ45CDB **V 9.0.4** (または CSQ4XCDB) を、この Db2 データ共有グループに接続するすべてのキュー・マネージャーによって使用されるデータベースを作成するようにカスタマイズして、それを実行します。
3. サンプル JCL CSQ45CTS **V 9.0.4** (または CSQ4XCTS) を、(手順 1 で作成される) キュー共有グループ用に使用されるキュー・マネージャーおよびチャンネル・イニシエーターを含む表スペースを作成するようにカスタマイズして、それを実行します。
4. サンプル JCL CSQ45CTB **V 9.0.4** (または CSQ4XCTB) を、12 個の Db2 表と関連索引を作成するようにカスタマイズして、それを実行します。行名または属性は変更しないでください。
5. サンプル JCL CSQ45BPL を、キュー・マネージャー、ユーティリティ、およびチャンネル・イニシエーター用の Db2 計画をバインドするようにカスタマイズして、それを実行します。
6. サンプル JCL CSQ45GEX を、キュー・マネージャー、ユーティリティ、およびチャンネル・イニシエーターによって使用されるユーザー ID 用の計画に実行権限を認可するようにカスタマイズして、それを実行します。キュー・マネージャーおよびチャンネル・イニシエーター用のユーザー ID は、開始された作業プロシージャが実行しているユーザー ID です。ユーティリティのユーザー ID は、バッチ・ジョブが実行依頼されるユーザー ID です。

該当する計画の名前を以下の表に示します。

- **LTS** LTS 列に Long Term Support ・バージョン。
- **CD** CD 列の Continuous Delivery バージョン。n は CD リリースを表します。

リリースごとに n は 1 ずつ増加します。例えば、IBM MQ 9.0.3 では、CSQ5A90n は CSQ5A903 です。

ユーザー	計画 (LTS)	計画 (CD)
キュー・マネージャー	CSQ5A 900, CSQ5C 900, CSQ5D 900, CSQ5K 900, CSQ5L 900, CSQ5M 900, CSQ5P 900, CSQ5R 900, CSQ5S 900, CSQ5T 900, CSQ5U 900, CSQ5W 900	CSQ5A 90n, CSQ5C 90n, CSQ5D 90n, CSQ5K 90n, CSQ5L 90n, CSQ5M 90n, CSQ5P 90n, CSQ5R 90n, CSQ5S 90n, CSQ5T 90n, CSQ5U 90n, CSQ5W 90n
CSQUTIL バッチ・ユーティリティ用の SDFS 機能	CSQ52 900	CSQ52 90n
CSQ5PQSG および CSQJUCNV バッチ・ユーティリティ	CSQ5B 900	CSQ5B 90n
CSQUZAP サービス・ユーティリティ	CSQ5Z 900	CSQ5Z 90n

Db2 のセットアップ中に障害が発生する場合、以下のジョブをカスタマイズして、実行することができます。

- CSQ45DTB。表および索引を除去します。
- CSQ45DTS **V 9.0.4** (または CSQ4XDTS)。表スペースを除去します。
- CSQ45DDB **V 9.0.4** (または CSQ4XDDB)。データベースを除去します。
- CSQ45DSG **V 9.0.4** (または CSQ4XDSEG)。ストレージ・グループを除去します。

注: Db2 のロックの問題によってこれらのジョブが失敗する場合は、Db2 リソースの競合が原因である可能性があります。特に、システムの負荷が大きい場合は、その可能性が高くなります。ジョブを後から再実行依頼してください。システムの負荷を軽減するか、またはシステムが静止状態になったときに、これらのジョブを実行することをお勧めします。

Db2 のセットアップについて詳しくは、*Db2 For z/OS 11.0.0* の「[Db2 Administration](#)」を参照してください。

**V 9.0.4** Db2 のセットアップについて詳しくは、*Db2 For z/OS 12.0.0* の「[Db2 Administration](#)」を参照してください。

Db2 表サイズに関する情報については、[z/OS での計画](#) を参照してください。

## 関連概念

710 ページの『[カップリング・ファシリティのセットアップ](#)』

キュー共有グループを使用する場合、キュー共有グループ (QSG) 内のキュー・マネージャーが使用するカップリング・ファシリティ構造を、IXCMIAPU を使用して、カップリング・ファシリティ・リソース管理 (CFRM) ポリシー・データ・セットに定義します。

## **z/OS** **カップリング・ファシリティのセットアップ**

キュー共有グループを使用する場合、キュー共有グループ (QSG) 内のキュー・マネージャーが使用するカップリング・ファシリティ構造を、IXCMIAPU を使用して、カップリング・ファシリティ・リソース管理 (CFRM) ポリシー・データ・セットに定義します。

- この作業は、キュー共有グループごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。
- キュー共有グループを使用しない場合は、この作業を省略してください。

後にキュー共有グループを使用する必要がある場合は、その時点でこのタスクを実行してください。

キュー共有グループのすべての構造は、キュー共有グループの名前で始まります。以下の構造を定義します。

- *qsg-name*CSQ\_ADMIN と呼ばれる管理構造。この構造は IBM MQ そのものに使用され、ユーザー・データは含まれません。
- *qsg-name* CSQSYSAPPL と呼ばれるシステム・アプリケーション構造。この構造は IBM MQ システム・キューが状態情報を保管するために使用されます。
- 共有キューのメッセージを保持するために使用する 1 つ以上の構造。これには、長さが 16 文字までの任意の名前を付けることができます。
  - 最初の 4 文字はキュー共有グループ名でなければなりません。(キュー共有グループ名が 4 文字未満の場合は、4 文字になるまで @ 記号を埋め込む必要があります。)
  - 5 番目の文字は英字で、その後の文字は英字または数字でなければなりません。名前のこの部分 (キュー共有グループ名以外) は、共有キューまたは CF 構造オブジェクトを定義する際に CFSTRUCT 名に指定するものです。

共有キューのメッセージを保持するために使用する構造名には英字および数字だけを使用できます。その他の文字 (例えば、\_ 文字。これは管理構造の名前で使用されます) は使用できません。

IXCMIAPU 用のサンプル制御ステートメントがデータ・セット `thlqual.SCSQPROC(CSQ4CFRM)` に入っています。これらをカスタマイズしてから、カップリング・ファシリティ用の IXCMIAPU ジョブに追加し、それを実行してください。

構造の定義が正常に終了したら、使用する予定の CFRM ポリシーをアクティブ化します。そのためには、次の z/OS コマンドを発行します。

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME= policy-name
```

CF 構造とそれらのサイズの計画方法については、[カップリング・ファシリティ・リソースの定義](#)を参照してください。

### 関連概念

673 ページの『[ESM セキュリティ管理を実施する](#)』

キュー・マネージャーとチャンネル・イニシエーターのセキュリティ管理を実装します。

### SMDS 環境をセットアップする

SMDS を使用して共有キューのメッセージをオフロードする場合は、SMDS オフロード・ストレージ環境をセットアップします。

- SMDS にデータをオフロードするように構成するキュー共有グループのキュー・マネージャーおよび構造ごとにこの作業を実行します。
- SMDS にデータをオフロードするための追加の構造を後で構成する場合は、その時点で、再度この作業を実行できます。
- キュー共有グループを使用しない場合は、この作業を省略してください。

後にキュー共有グループを使用する必要がある場合は、その時点でこのタスクを実行してください。

### SMDS 環境をセットアップする

1. 構造とデータ・セット・スペース所要量を見積もります。[共有メッセージ・データ・セット容量の考慮事項](#)を参照してください。
2. データ・セットを割り振り、事前フォーマットします。[共有メッセージ・データ・セットの作成](#)を参照してください。
3. IBM MQ に CF 構造を定義する場合、必ず CFLEVEL(5) および OFFLOAD(SMDS) を使用して CFSTRUCT を定義します。

### 関連概念

710 ページの『[カップリング・ファシリティのセットアップ](#)』

キュー共有グループを使用する場合、キュー共有グループ (QSG) 内のキュー・マネージャーが使用するカップリング・ファシリティ構造を、IXCMIAPU を使用して、カップリング・ファシリティ・リソース管理 (CFRM) ポリシー・データ・セットに定義します。

### IBM MQ 項目を Db2 表に追加する

キュー共有グループを使用している場合は、CSQ5PQSG ユーティリティを実行して、キュー共有グループとキュー・マネージャーの項目を Db2 データ共有グループの IBM MQ 表に追加します。

- この作業は、IBM MQ キュー共有グループごとに繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。
- キュー共有グループを使用しない場合は、この作業を省略してください。

後にキュー共有グループを使用する必要がある場合は、その時点でこのタスクを実行してください。

キュー共有グループごとに、およびキュー共有グループのメンバーになるキュー・マネージャーごとに CSQ5PQSG を実行してください。(CSQ5PQSG については、[IBM MQ for z/OS の管理](#)で説明しています。)

以下のアクションを指定された順序で実行します。

1. CSQ5PQSG プログラムの ADD QSG 機能を使用して、キュー共有グループの項目を IBM MQ Db2 表に追加します。thlqual.SCSQPROC(CSQ45AQS) にサンプルが用意されています。

Db2 データ共有グループに定義されているそれぞれのキュー共有グループごとに、この機能を 1 回実行します。キュー共有グループの項目は、そのキュー共有グループを参照するキュー・マネージャーの項目を追加する前に、存在していなければなりません。

2. CSQ5PQSG プログラムの ADD QMGR 機能を使用して、キュー・マネージャーの項目を IBM MQ Db2 表に追加します。thlqual.SCSQPROC(CSQ45AQM) にサンプルが用意されています。

キュー共有グループのメンバーになるそれぞれのキュー・マネージャーごとに、この機能を実行します。

注:

- a. 1つのキュー・マネージャーに対して、それをメンバーにできるのは1つのキュー共有グループだけです。
- b. キュー共有グループを使用できるようにするには、RRS が実行されていなければなりません。

## 関連概念

679 ページの『システム・パラメーター・モジュールの調整』

IBM MQ システム・パラメーター・モジュールは、IBM MQ がその処理で使用するロギング、保存、トレース、および接続の環境を制御します。デフォルトのモジュールが提供されます。いくつかのパラメーター(データ・セット名など)は通常、サイト固有であるため、独自のシステム・パラメーター・モジュールを作成する必要があります。

z/OS

## キュー共有グループの ESM セキュリティー管理の実装

キュー共有グループ内のすべてのキュー・マネージャーが Db2 とカップリング・ファシリティ・リスト構造にアクセスするためのセキュリティ管理を実装します。

- キュー共有グループ内の IBM MQ キュー・マネージャーごとにこの作業を繰り返してください。
- 旧バージョンから移行する際に、この作業を実行しなければならない場合もあります。

キュー・マネージャー、チャンネル・イニシエーター、およびユーティリティーに関連したユーザー ID が、接続を確立する各 Db2 サブシステムに対して RRSF 接続を確立する権限を持っていることを確認してください。キュー・マネージャーおよびチャンネル・イニシエーター用のユーザー ID は、開始された作業プロシージャーが実行しているユーザー ID です。

ユーティリティーのユーザー ID は、バッチ・ジョブが実行依頼されるユーザー ID です。ユーザー ID が READ アクセスを必要とする RACF プロファイルは、DSNR 資源クラスの Db2ssid.RRSF です。

キュー共有グループ内の各キュー・マネージャーに関連付けられているユーザー ID に、カップリング・ファシリティ・リスト構造に対する適切なアクセス・レベルを付与する必要があります。RACF クラスは FACILITY です。

以下のユーザー ID には ALTER アクセス権が必要です。

- IXLSTR.structure-name プロファイルに対するキュー・マネージャー ID
- CSQ5PQSG を実行するユーザー ID

## 関連概念

673 ページの『ESM セキュリティー管理を実施する』

キュー・マネージャーとチャンネル・イニシエーターのセキュリティ管理を実装します。

z/OS

## Advanced Message Security for z/OS の構成

これらのトピックを、Advanced Message Security を構成するためのステップバイステップ・ガイドとしてご利用ください。

z/OS

## Advanced Message Security 用のプロシージャーを作成する

Advanced Message Security を使用するように構成される各 IBM MQ サブシステムには、AMS アドレス・スペースを開始するカタログ式プロシージャーが必要です。独自のプロシージャー・ライブラリーを作成するか、あるいは IBM 提供のプロシージャー・ライブラリーを使用することができます。

Advanced Message Security を使用するように構成される IBM MQ サブシステムごとに、サンプル・プロシージャー CSQ4AMSM のコピーを調整します。これを行うには、以下のステップを実行します。

1. サンプルの開始タスク・プロシージャー thlqual.SCSQPROC(CSQ4AMSM) を SYS1.PROCLIB にコピーするか、SYS1.PROCLIB を使用していない場合はプロシージャー・ライブラリーにコピーします。このプ

ロシージャーに xxxxAMSM という名前を付けます。ここで、xxxx は IBM MQ サブシステムの名前です。例えば、CSQ1AMSM は、キュー・マネージャー CSQ1 の AMS 開始タスク・プロシージャーに相当します。

2. 使用する IBM MQ サブシステムごとに 1 つずつコピーを作成します。
3. サンプル・プロシージャー CSQ4AMSM 内の命令を使用して、要件に合うようにプロシージャーを調整します。さらに、JCL で記号パラメーターを使用して、プロシージャーを開始時に変更できます。
4. Language Environment® の \_CEE\_ENVFILE ファイルを使用して AMS タスクに渡されるパラメーターを検討し、任意に変更します。サンプル thlqual.SCSQPROC(CSQ40ENV) は、サポートされるパラメーターをリストしています。

注：この作業は、IBM MQ キュー・マネージャーごとに繰り返してください。

## **z/OS** 開始タスク・ユーザー *Advanced Message Security* をセットアップする

Advanced Message Security タスクには、それを UNIX システム・サービス・プロセスとして認識させることができるユーザー ID が必要です。

さらに、このタスクに割り当てられるユーザーは、UNIX システム・サービス・ユーザーとして認識されるように、UNIX UID (ユーザー ID) および GID (グループ ID) が適切に定義されていなければなりません。UNIX システム・サービスの UID および GID の定義については、「[z/OS: Security Server RACF セキュリティー管理者のガイド](#)」を参照してください。

「[z/OS: UNIX システム・サービス計画](#)」では、従来の UNIX セキュリティーを z/OS セキュリティーと比較しています。従来の UNIX セキュリティーと z/OS セキュリティーとの主な違いは、カーネル・サービスで UNIX レベルと z/OS UNIX レベルという 2 つのレベルの適切な特権がサポートされる点にあります。

インストール済み環境のセキュリティ・ポリシーに応じて、Advanced Message Security タスクは、スーパーユーザー権限 (uid(0)) で実行するか、あるいは、RACF FACILITY クラスの BPX.DAEMON および BPX.SERVER プロファイルに対するアクセス権がある RACF ID で実行する必要があります (このタスクに、そのユーザーの RACF ID を付与できなければならないため)。

後者の方式が使用される場合、あるいは BPX.DAEMON または BPX.SERVER プロファイルを既にアクティブ化している場合には、Advanced Message Security タスク・プログラム (thlqual.SCSQAUTH(CSQ0DSRV)) が、RACF プログラム管理ライブラリーに配置されていなければなりません。

「[z/OS: UNIX システム・サービス計画](#)」を参照して、従来の UNIX セキュリティーと z/OS UNIX セキュリティーとのセキュリティ上の違いを把握するようにしてください。これにより、UNIX System Services の特権プロセスをデプロイして実行するために、インストール済み環境のセキュリティ・ポリシーに従って Advanced Message Security タスクを管理することが可能になります。

この作業を検討する上で、以下の資料が参照用として役に立ちます。

- [z/OS: UNIX システム・サービス計画](#)。
- [z/OS: Security Server RACF セキュリティー管理者のガイド](#)。

注：このタスクのユーザー ID に関連付けられた鍵リングに Advanced Message Security の受信側証明書がロードされるため、このユーザー ID は注意深く選択してください。この考慮事項については、[z/OS](#) での証明書の使用に説明されています。

以下の手順では、Advanced Message Security 開始タスク・ユーザーのセットアップ方法を示します。この手順では、RACF コマンドを例として使用します。別のセキュリティ・マネージャーを使用する場合には、同等のコマンドを使用する必要があります。

注：このセクションの例は、RACF の STARTED、FACILITY、および SURROGAT クラスの総称プロファイル・コマンド処理と総称プロファイル検査を既にアクティブ化していることを前提としています。RACF で総称プロファイルを処理する方法については、「[z/OS: Security Server RACF コマンド言語解説書](#)」を参照してください。

1. 最初に、Advanced Message Security 開始タスク・ユーザーの RACF ユーザー・プロファイルを定義します。これらは、同じユーザーでも構いません。

```
ADDUSER WMQAMSM NAME(' Advanced Message Security user') OMVS (UID(0)) DFLTGRP(group)
```

インストール済み環境の基準に応じて適切にデフォルトのグループを選択します。

**注:** USS スーパーユーザー権限 (UID(0)) を付与しない場合は、BPX.DAEMON および BPX.SERVER 機能クラス・プロファイルに対するアクセス権を Advanced Message Security ユーザー ID に付与する必要があります。

```
PERMIT BPX.DAEMON CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
```

また、Advanced Message Security タスク・プログラム (*thlqual.SCSQAUTH(CSQ0DSRV)*) が RACF プログラム管理ライブラリーに配置されていなければなりません。

SCSQAUTH ライブラリーをプログラム管理下に置くには、以下のコマンドを使用します。

```
RALTER PROGRAM * ADDMEM('thlqual.SCSQAUTH'//NOPADCHK) -or-  
RALTER PROGRAM ** ADDMEM('thlqual.SCSQAUTH'//NOPADCHK)  
SETROPTS WHEN(PROGRAM) REFRESH
```

また、Advanced Message Security タスクによって使用される各国語ライブラリー (*thlqual.SCSQANLx*) のプログラム管理も有効にする必要があります。

2. RACF STARTED クラスがアクティブであるかどうかを判別します。アクティブでない場合は、RACF STARTED クラスをアクティブ化します。

```
SETROPTS CLASSACT(STARTED)
```

3. ステップ 1 で選択または作成したユーザー ID を指定して、Advanced Message Security タスクの開始クラス・プロファイルを定義します。

```
RDEFINE STARTED qmgr AMSM.* STDATA(USER(WMQAMSM))
```

ここで、*qmgr* は、開始タスク名の接頭部の名前です。例えば、開始タスクの名前を CSQ1AMSM とします。この場合は、*qmgr* AMSM.\* を CSQ1AMSM.\* に置き換えます。

開始タスク名の名前は *qmgr* AMSM.\* でなければなりません。

4. SETROPTS RACF コマンドを使用して、ストレージ内 RACLIST 開始クラス・プロファイルをリフレッシュします。

```
SETROPTS RACLIST(STARTED) REFRESH
```

5. Advanced Message Security タスクには、IBM MQ メッセージの保護処理中に、クライアント要求側のホスト・ユーザー ID の ID が一時的に付与されます。このため、要求を行うことができる各ユーザー ID の SURROGAT クラスにプロファイルを定義する必要があります。

RACF SURROGAT クラスがアクティブである場合には、これを単一の総称プロファイルを使用して行うことができます。SURROGAT クラスがアクティブでない場合には、検査が無視されます。必要な SURROGAT プロファイルについては、「z/OS: UNIX システム・サービス計画」に説明されています。

SURROGAT クラスにプロファイルを定義するには、以下を行います。

- a. RACF SETROPTS コマンドを使用して RACF SURROGAT クラスをアクティブ化します。

```
SETROPTS CLASSACT(SURROGAT)
```

- b. RACFSURROGAT クラスの総称プロファイル処理を活動化します:

```
SETROPTS GENERIC(SURROGAT)
```

- c. RACF SURROGAT クラスの総称プロファイル・コマンド処理をアクティブ化します。

```
SETROPTS GENCMD(SURROGAT)
```

- d. サロゲート・クラス総称プロファイルを定義します。

```
RDEFINE SURROGAT BPX.SRV.* UACC(NONE)
```

- e. 総称 SURROGAT クラス・プロファイルに対するアクセス権を Advanced Message Security ユーザー ID に付与します。

```
PERMIT BPX.SRV.* CLASS(SURROGAT) ID(WMQASMS) ACCESS(UPDATE)
```

注：「z/OS: UNIX システム・サービス計画」に説明されているように、特定のユーザーが Advanced Message Security タスクによって処理されるように制限する必要がある場合、より具体的なプロファイルを定義できます。

- f. Advanced Message Security ユーザー ID を BPX.SERVER 機能に対して許可します(証明書および鍵リングの作成でまだ実行されていない場合):

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(WMQASMS) ACCESS(READ)
```

6. Advanced Message Security タスクでは、z/OS System SSL サービスによって提供される機能を使用して、SAF 管理の鍵リングを開きます。鍵リングの内容にアクセスする、基礎となるシステム許可機能 (SAF) は、RACF か、または同等のセキュリティー・マネージャーによって制御されます。

このサービスは IRRSDL00 (R\_datalib) 呼び出し可能サービスです。この呼び出し可能サービスは、RACF FACILITY クラスに定義される、RACF RACDCERT コマンドの保護に使用するのと同じプロファイルで保護されます。したがって、Advanced Message Security ユーザー ID には、それらのプロファイルへのアクセス権を、以下のコマンドを使用して付与する必要があります。

- a. まだ行っていない場合は、RACDCERT コマンドおよび IRRSDL00 呼び出し可能サービスを保護する RACF FACILITY クラスに、RACF 総称プロファイルを定義します。

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)  
SETROPTS RACLIST(FACILITY) REFRESH
```

- b. RACF 総称プロファイルに対する権限を、開始タスク・ユーザー ID に付与します。

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(WMQASMS) ACC(READ)
```

あるいは、以下のように、RDATALIB クラスのデータ・サービス・タスク・ユーザーの鍵リングに対する READ アクセス権を付与することができます。

```
PERMIT WMQASMD.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(WMQASMS) ACC(READ)
```

## AMS のリソース・セキュリティー

開始タスク・ユーザーは、SYSTEM.PROTECTION.POLICY.QUEUE に対する読み取り権限を必要とします。

開始タスク・ユーザーは、BATCH アプリケーションとしてキュー・マネージャーに接続するための権限を必要とします。詳しくは、[バッチ接続のための接続セキュリティー・プロファイル](#)を参照してください。

## Advanced Message Security のセキュリティー管理者に RACDCERT 権限を付与する

Advanced Message Security のセキュリティー管理者には、デジタル証明書を作成して管理するために RACDCERT コマンドを使用する権限が必要です。

この役割のために適切なユーザー ID を特定し、RACDCERT コマンドを使用する権限を付与します。以下に例を示します。

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(admin) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

ここで、admin は、Advanced Message Security セキュリティー管理者のユーザー ID です。

## **Advanced Message Security のためのリソース・アクセス権をユーザーに付与する**

Advanced Message Security のユーザーには、適切なリソース・アクセス権が必要です。

Advanced Message Security のユーザー、つまり Advanced Message Security によって保護されたメッセージの書き込みと取得を行うユーザーは、以下を必要とします。

- ユーザー ID に関連付けられた OMVS セグメント
- IRR.DIGTCERT.LISTRING または RDATA LIB に対するアクセス権
- ICSF クラスの CSFSERV および CSFKEYS プロファイルに対するアクセス権

Advanced Message Security タスクは、クライアント ID を一時的に想定します。つまり、Advanced Message Security によって、保護されているキューへの IBM MQ メッセージの処理中に、タスクは Advanced Message Security ユーザーの z/OS ユーザー ID の代理として機能することになります。

このタスクにユーザーの z/OS ID が付与されるようにするには、クライアントの z/OS ユーザー ID に、ユーザー・プロファイルと関連付けられた OMVS セグメントが定義されていなければなりません。

管理の助けとして、RACF は、RACF ユーザーおよびグループのプロファイルに関連付けできる、デフォルト OMVS セグメントを定義する機能を備えています。このデフォルトは、z/OS ユーザー ID またはグループのプロファイルに OMVS セグメントが明示的に定義されていない場合に使用されます。多数のユーザーが Advanced Message Security を使用する予定であるなら、ユーザーごとに明示的に OMVS セグメントを定義する代わりに、このデフォルトの使用を選択することもできます。

「z/OS: Security Server RACF セキュリティー管理者のガイド」には、デフォルト OMVS セグメントを定義する手順が詳しく説明されています。この資料に概説されている手順を参照して、RACF ユーザーおよびグループのプロファイルにデフォルト OMVS セグメントを定義することが、ご使用のインストール済み環境にとって適切かどうか見極めてください。

IRR.DIGTCERT.LISTRING クラス機能に対する READ 権限を、すべての Advanced Message Security ユーザーに付与するには、次のコマンドを発行します。

```
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(READ)
```

または、ユーザーごとに READ 権限を付与するには、次のコマンドを発行します。

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
```

ここで、userid は Advanced Message Security ユーザーの名前です。

あるいは、RDATA LIB クラスを使用して、特定の鍵リングに対するアクセス権を付与することもできます (RDATA LIB 権限は、IRR.DIGTCERT.LISTRING 権限よりも優先されます)。以下に例を示します。

```
PERMIT user.DRQ.AMS.KEYRING.LST CLASS(RDATA LIB) ID(user) ACC(READ)
```

ICSF 管理の証明書と秘密鍵を使用している場合には、Advanced Message Security ユーザーは、特定のクラス CSFSERV および CSFKEYS のプロファイルに対するアクセス権を必要とします。このアクセス権について、以下の表に示します。

表 48. クラス CSFSERV および CSFKEYS のプロファイルに対する必須のユーザー・アクセス権

Class	プロファイル	アクセス権
CSFSERV	CSFDSG	READ
CSFSERV	CSFPKE	READ
CSFSERV	CSFPKD	READ
CSFSERV	CSFDSV	READ
CSFKEYS	ICSF PKDS ラベル	READ

## V 9.0.1 z/OS mqweb サーバーの構成

これらのトピックを、mqweb サーバーを構成するためのステップバイステップ・ガイドとしてご利用ください。

### 関連タスク

605 ページの『IBM MQ Console および REST API の構成』

IBM MQ Console と REST API をホストする mqweb サーバーには、デフォルトの構成が準備されています。どちらのコンポーネントを使用する場合も、ユーザーにログインを許可するために、セキュリティーを構成するなどの複数の構成タスクを実行する必要があります。このトピックでは、使用可能なすべての構成オプションについて説明します。

## V 9.0.1 z/OS Liberty サーバー定義の作成

IBM MQ for z/OS Unix System Services Web Components をインストールして MQ Console、または REST API を使用する場合は、Liberty サーバー定義を作成してカスタマイズする必要があります。

### 始める前に

Liberty サーバーを使用するには、SYSTEM.REST.REPLY.QUEUE を作成する必要があります。これを行うには、675 ページの『初期設定入力データ・セットをカスタマイズする』にある最新の **CSQ4INSG** サンプルを使用してください。

### このタスクについて

- この作業は、MQ Console または REST API を実行するそれぞれの z/OS システムごとに実行する必要があります。
- 実行中の IBM MQ のバージョンごとに Liberty サーバーが必要です。
- 前のバージョンから移行する際に、サーバー構成をリフレッシュまたは変更しなければならない場合があります。

IBM MQ for z/OS Unix System Services Web コンポーネントでは、mqweb という単一の Liberty サーバーを作成する必要があります。

このサーバーの構成ファイルとログ・ファイルはすべて、Liberty ユーザー・ディレクトリー内に格納されます。

以下の手順を実行して mqweb サーバー定義を作成します。

### 手順

1. Liberty ユーザー・ディレクトリーとして適切な場所を選択します。  
mqweb サーバーを実行するユーザー ID には、このユーザー・ディレクトリーとそのコンテンツへの読み取り権限と書き込み権限が必要です。このユーザー・ディレクトリーにはログ・ファイルやサーバー構成が格納されるため、このディレクトリーは別のファイル・システムに作成する必要があります。
2. 現行ディレクトリーが **crtmqweb.sh** スクリプトのロケーションである PathPrefix/web/bin であることを確認します。

PathPrefix は IBM MQ UNIX System Services Components のインストール・パスです。

3. **crtmqweb.sh** スクリプトを実行して、テンプレートの mqweb サーバー定義が含まれた Liberty ユーザー・ディレクトリーを作成します。

注: **crtmqweb.sh** スクリプトは、オプション・パラメーターを 1 つ (Liberty ユーザー・ディレクトリーの名前) 受け入れます。

Liberty ユーザー・ディレクトリーの名前を指定しないと、デフォルト値の /var/mqm/web/installation1 が使用されます。

4. 以下のコマンドを使用して、Liberty ユーザー・ディレクトリー内のディレクトリーとファイルの所有権を、mqweb サーバーを実行するユーザー ID とグループに属するように変更します。

```
chown -R userid:group path
```

このパスに対する書き込み権限をグループに付与するには、次のコマンドを実行します。

```
chmod -R 770 path
```

## 次のタスク

718 ページの『[Liberty サーバー用のプロシーチャーの作成](#)』

### 関連タスク

605 ページの『[IBM MQ Console および REST API の構成](#)』

IBM MQ Console と REST API をホストする mqweb サーバーには、デフォルトの構成が準備されています。どちらのコンポーネントを使用する場合も、ユーザーにログインを許可するために、セキュリティーを構成するなどの複数の構成タスクを実行する必要があります。このトピックでは、使用可能なすべての構成オプションについて説明します。

V 9.0.1

z/OS

## Liberty サーバー用のプロシーチャーの作成

IBM MQ for z/OS Unix System Services Web Components をインストールして MQ Console、または REST API を使用する場合は、Liberty mqweb サーバーを開始するためのカタログ式プロシーチャーを作成する必要があります。

- この作業は、IBM MQ を実行するそれぞれの z/OS システムごとに実行する必要があります。
- 実行中の IBM MQ のバージョンごとに Liberty サーバー・インスタンスが必要です。例えば、IBM MQ 9.0.1 のキュー・マネージャーの MQWB0901 という開始タスク、IBM MQ 9.0.2 のキュー・マネージャーの MQWB0902 という開始タスクなどがあります。

キュー・マネージャーが 1 つしかない場合は、単一の Liberty サーバーの開始タスクを実行し、それがキュー・マネージャーの移行時に使用するライブラリーを変更できます。

- 前のバージョンから移行する際に、カタログ式プロシーチャーを変更しなければならない場合があります。

以下の手順を実行して、カタログ式プロシーチャーを作成します。

1. サンプルの開始タスク・プロシーチャー th1qua1.SCSQPROC (CSQ4WEBS) をプロシーチャー・ライブラリーにコピーします。

企業の規約に従ってプロシーチャーに名前を付けます。

例えば、MQWB0901 は、これが Liberty for IBM MQ 9.0.1 のカタログ式プロシーチャーであることを示します。

2. サンプル・プロシーチャー CSQ4WEBS 内の命令を使用して、要件に合うようにプロシーチャーを調整します。

Liberty ユーザー・ディレクトリーは、mqweb サーバー定義を作成するために **crtmqweb.sh** スクリプトを実行したときに指定したディレクトリーです。

詳細については、[717 ページの『Liberty サーバー定義の作成』](#)を参照してください。

3. 外部セキュリティー管理プログラムの下で実行できるようにプロシーチャーに許可を与えます。

#### 4. S **procname** コマンドを使用してプロシージャを開始します。

「+CWWKE0001I: サーバー mqweb が起動しました。(The server mqweb has been launched.)」というメッセージが生成されます。

サーバーが正常に開始しない場合は、メッセージを確認してください。

プロシージャが開始すると、出力が **USERDIR** パラメーターの下にあるファイルに保管されます。例えば、ユーザー・ディレクトリーが /u/mq/mqweb である場合は、/u/mq/mqweb/servers/mqweb/logs を確認してください。

このファイルは ASCII で書き込まれるため、一般的なシステム・ツールを使用して表示することができません。

#### 5. IBM ワークロード・マネージャー (WLM) を使用して、このアドレス・スペースを分類します。

Liberty サーバーは IBM MQ アプリケーションで、ユーザーはこのアプリケーションと対話します。アプリケーションは WLM での重要度を高くする必要はなく、サービス・クラス **STCUSER** が適切です。

#### 6. P **procname** コマンドを使用してプロシージャを停止します。

注:

- ファイルのデータが小文字になるように、メンバーの編集時には必ず「**大文字オフ (Caps off)**」を指定してください。
- Web サーバーの開始やシャットダウンにかなりの時間がかかる (場合によっては 1 分を超える) こともあります。

## 次のタスク

[ユーザーおよび役割の構成](#)

### 関連タスク

605 ページの『[IBM MQ Console および REST API の構成](#)』

IBM MQ Console と REST API をホストする mqweb サーバーには、デフォルトの構成が準備されています。どちらのコンポーネントを使用する場合も、ユーザーにログインを許可するために、セキュリティーを構成するなどの複数の構成タスクを実行する必要があります。このトピックでは、使用可能なすべての構成オプションについて説明します。

## **ReportingService (旧称 BluemixRegistration) スタンザの構成**

このタスクは、IBM Cloud (formerly Bluemix) 上の IBM Cloud Product Insights サービスに登録データと使用データを公開する作業の 1 つに含まれていました。IBM Cloud Product Insights サービスは使用可能でなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## **z/OS でのキュー・マネージャーのテスト**

キュー・マネージャーのカスタマイズまたは移行を済ませた後、IBM MQ for z/OS に添付されているインストール検査プログラムおよびサンプル・アプリケーションを実行することによって、キュー・マネージャーをテストできます。

### このタスクについて

IBM MQ for z/OS をインストールおよびカスタマイズした後、提供されているインストール検査プログラム CSQ4IVP1 を使用して、IBM MQ for z/OS が操作可能であることを確認できます。

基本的なインストール検査プログラム CSQ4IVP1 は、非共有キューをテストし、C、COBOL、CICS サンプルを使用せずに基本 IBM MQ を検査します。

基本インストール検査を実行した後、別のキューで CSQ4IVP1 を使用して共有キューをテストし、Db2 とカップリング・ファシリティーが正しくセットアップされていることをテストすることもできます。分散

キューイングが操作可能であることを確認するには、提供されているインストール検査プログラム CSQ4IVPX を使用できます。

CSQ4IVP1 はロード・モジュールとして提供され、Message Queue Interface (MQI) の典型的な使用方法を示すソース・モジュールとして一連のプロシージャ型サンプル・アプリケーションを提供します。これらのソース・モジュールを使用して、さまざまなプログラミング言語環境をテストすることができます。提供されたサンプル JCL を使用して、インストール済み環境に適した他のサンプルのうちどれでもコンパイルおよびリンク・エディットできます。

## 手順

- z/OS でキュー・マネージャーをテストする方法の詳細については、以下のサブトピックを参照してください。
  - [720 ページの『基本インストール検査プログラムの実行』](#)
  - [724 ページの『キュー共有グループのテスト』](#)
  - [725 ページの『分散キューイングのテスト』](#)
  - [728 ページの『IBM MQ for z/OS を使用した C、C++、COBOL、PL/I、および CICS プログラムのテスト』](#)

## 関連タスク

[657 ページの『z/OS でのキュー・マネージャーの構成』](#)

IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

## 関連情報

[IBM MQ for z/OS の概念](#)

[z/OS での IBM MQ 環境の計画](#)

[管理 IBM MQ for z/OS](#)

## **基本インストール検査プログラムの実行**

IBM MQ をインストールおよびカスタマイズした後、提供されているインストール検査プログラム CSQ4IVP1 を使用して、IBM MQ が操作可能であることを確認できます。

基本的なインストール検査プログラムは、C、COBOL、CICS サンプルを使用せずに基本 IBM MQ を検査するバッチ・アセンブラー IVP です。

バッチ・アセンブラー IVP は SMP/E でリンク・エディットされ、そのロード・モジュールはライブラリー thlqual.SCSQLOAD に入って出荷されます。

SMP/E APPLY ステップおよびカスタマイズ・ステップの両方を完了した後、バッチ・アセンブラー IVP を実行してください。

詳細については、以下のセクションを参照してください。

- [CSQ4IVP1 アプリケーションの概要](#)
- [CSQ4IVP1 の実行準備](#)
- [CSQ4IVP1 の実行](#)
- [CSQ4IVP1 の実行結果の検査](#)

## CSQ4IVP1 アプリケーションの概要

CSQ4IVP1 は、使用している IBM MQ サブシステムに接続して、次の基本機能を実行するためのバッチ・アプリケーションです。

- IBM MQ 呼び出しを発行する。
- コマンド・サーバーと通信する。
- トリガーが活動状態であることを検証する。

- 動的キューを生成および削除する。
- メッセージの有効期限処理を検査する。
- メッセージのコミット処理を検査する。

## CSQ4IVP1 の実行準備

CSQ4IVP1 を実行する前に、次のことを行ってください。

1. IVP 項目が、キュー・マネージャー始動プログラムの中の CSQINP2 データ・セット連結に入っているかどうか検査します。この IVP 項目は、メンバー `thlqual.SCSQPROC(CSQ4IVPQ)` に入って提供されず。入っていない場合には、`thlqual.SCSQPROC(CSQ4IVPQ)` で提供された定義を、CSQINP2 連結に追加してください。キュー・マネージャーが現在実行されている場合、これらの定義を有効にするために、キュー・マネージャーを再始動する必要があります。
2. インストール検査プログラムを実行するために必要なサンプル JCL の CSQ4IVPR は、ライブラリー `thlqual.SCSQPROC` に入っています。

CSQ4IVPR JCL をカスタマイズするため、IBM MQ ライブラリーの高位修飾子、使用する各国語、4 文字の IBM MQ キュー・マネージャー名、およびジョブ出力の宛先を指定します。

3. RACF を更新して、IBM MQ セキュリティーが活動状態の場合に CSQ4IVP1 が RACF の資源にアクセスできるようにします。

IBM MQ のセキュリティーが使用可能になったときに CSQ4IVP1 を実行するには、オブジェクトへのアクセス権を持っている RACF ユーザー ID が必要です。RACF への資源の定義の詳細は、[z/OS でのセキュリティーのセットアップ](#)を参照してください。IVP を実行するユーザー ID には、次のようなアクセス権が必要です。

Authority	プロファイル	Class
READ	ssid.DISPLAY.PROCESS	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.COMMAND.REPLY.MODEL	MQQUEUE
UPDATE	ssid.CSQ4IVP1.**	MQQUEUE
READ	ssid.BATCH	MQCONN

上記の要件は、すべての IBM MQ セキュリティーが活動状態であることを想定しています。IBM MQ のセキュリティーを活動化するための RACF コマンドは、[721 ページの図 100](#) に示してあります。この例では、キュー・マネージャー名が CSQ1 であること、およびサンプル CSQ4IVP1 を実行するユーザーのユーザー ID が TS101 であることを前提としています。

```
RDEFINE MQCMDS CSQ1.DISPLAY.PROCESS
PERMIT CSQ1.DISPLAY.PROCESS CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.REPLY.MODEL
PERMIT CSQ1.SYSTEM.COMMAND.REPLY.MODEL CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.CSQ4IVP1.**
PERMIT CSQ1.CSQ4IVP1.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)
```

図 100. CSQ4IVP1 のためのコマンド RACF

## CSQ4IVP1 の実行

これらのステップを完了したら、キュー・マネージャーを開始してください。キュー・マネージャーが既に実行中で、かつ CSQINP2 を変更した場合は、キュー・マネージャーを停止して再始動する必要があります。

IVP はバッチ・ジョブとして実行されます。ジョブ・カードを、インストール先のジョブ実行依頼の要件に合うようにカスタマイズしてください。

## CSQ4IVP1 の実行結果の検査

IVP は、10 のステージに分かれており、各ステージが完了して完了コード・ゼロを出さないと、次のステージを実行することはできません。IVP は、次のものをリストしたレポートを生成します。

- 接続されているキュー・マネージャーの名前
- 各ステージから戻された完了コードおよび理由コードを示す 1 行のメッセージ
- 1 行の通知メッセージ (該当する場合)

サンプル・レポートは、[724 ページの図 101](#)にあります。

**z/OS** 完了コードと理由コードの解説については、[IBM MQ for z/OS のメッセージ、完了コード、および理由コード](#)を参照してください。

ステージによっては複数の IBM MQ 呼び出しがあるため、障害があった場合は、障害を戻した特定の IBM MQ 呼び出しを示すメッセージが発行されます。また、あるステージでは、IVP が説明および診断のための情報をコメント欄に書き込みます。

IVP ジョブは、特定のキュー・マネージャー・オブジェクトの排他制御を要求するので、システム全体を通して単一スレッドで実行されます。ただし、キュー・マネージャーに対して実行できる IVP の回数には、制限がありません。

各ステージで実行される機能は、次のとおりです。

### ステージ 1

MQCONN API 呼び出しを発行することにより、キュー・マネージャーに接続します。

### ステージ 2

要求メッセージを取り出すために、コマンド・サーバーによって使用されるシステム・コマンド入力キューの名前を判別します。このキューは、ステージ 5 からの表示要求を受け取ります。

これを行うための呼び出しの順序は、次のとおりです。

1. キュー・マネージャー・オブジェクトをオープンするために、キュー・マネージャー名を指定して MQOPEN 呼び出しを発行します。
2. システム・コマンド入力キューの名前を知るために、MQINQ 呼び出しを発行します。
3. キュー・マネージャーの各種イベント・スイッチを知るために、MQINQ 呼び出しを発行します。
4. キュー・マネージャー・オブジェクトを閉じるために、MQCLOSE 呼び出しを発行します。

このステージが正常に完了すると、システム・コマンド入力キューの名前がコメント欄に表示されます。

### ステージ 3

MQOPEN 呼び出しを使用して、開始キューをオープンします。

このキューは、ステージ 5 からのリクエストにコマンドサーバーが返答した結果として、到着するトリガー・メッセージを予期して、このステージでオープンされます。トリガー基準を満たすために、入力用にキューをオープンする必要があります。

### ステージ 4

モデルとして CSQ4IVP1.MODEL キューを使用して、永続動的キューを作成します。動的キューは、それが作成されたモデルと同じ属性を持ちます。これは、ステージ 5 でのコマンド・サーバー要求からの応答がこのキューに書き込まれると、ステージ 3 でオープンされた開始キューにトリガー・メッセージが書き込まれることを意味します。

このステージが正常に完了すると、永続動的キューの名前がコメント欄に表示されます。

#### ステージ 5

コマンド・サーバーのコマンド・キューに対して、MQPUT1 要求を発行します。

タイプが MQMT\_REQUEST のメッセージが、プロセス CSQ4IVP1 の表示を要求して、システム・コマンド入力キューに書き込まれます。このメッセージのメッセージ記述子は、コマンド・サーバーの応答のための応答先キューとしてステージ 4 で作成された永続動的キューを指定しています。

#### ステージ 6

開始キューから **MQGET** 要求を発行します。この段階では、ステージ 3 でオープンされた開始キューに対して、1 分間隔で GET WAIT が発行されます。戻されるメッセージは、応答先キューに書き込まれるコマンド・サーバーの応答メッセージによって、生成されるトリガー・メッセージとなることが期待されます。

#### ステージ 7

ステージ 4 で作成した永続動的キューを削除します。キューには依然としてメッセージが含まれているため、MQCO\_PURGE\_DELETE オプションが使用されます。

#### ステージ 8

1. 動的キューをオープンします。
2. 有効期限が設定されたメッセージを MQPUT します。
3. メッセージの期限が切れるのを待ちます。
4. 期限切れのメッセージに対して MQGET を試行します。
5. キューを MQCLOSE します。

#### ステージ 9

1. 動的キューをオープンします。
2. メッセージを MQPUT します。
3. MQCMIT を発行して、現在の作業単位をコミットします。
4. メッセージを MQGET します。
5. MQBACK を発行して、メッセージをバックアウトします。
6. 同じメッセージを MQGET し、バックアウト・カウントが 1 に設定されていることを確認します。
7. MQCLOSE を発行して、キューを閉じます。

#### ステージ 10

**MQDISC** を使用して、キュー・マネージャーから切断します。

IVP を実行した後、必要のないオブジェクトはすべて削除できます。

IVP を正常に実行できない場合は、手動でステップを 1 つ 1 つ試行し、どの機能に障害があるかを判別してください。

```

DATE : 2005.035                IBM MQ for z/OS - V6                PAGE : 0001
INSTALLATION VERIFICATION PROGRAM
PARAMETERS ACCEPTED. PROGRAM WILL CONNECT TO : CSQ1
,OBJECT QUALIFER : CSQ4IVP1
INSTALLATION VERIFICATION BEGINS :
STAGE 01 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR BRIDGE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS EXCP FOR CHANNEL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR SSL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR INHIBITED EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR LOCAL EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR PERFORMANCE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR REMOTE EVENTS
STAGE 02 INFO: QMGR EVENT SWITCH IS OFF FOR START/STOP EVENTS
STAGE 02 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 SYSTEM.COMMAND.INPUT
STAGE 03 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 04 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810EFEAC8980
STAGE 05 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 06 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 07 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
STAGE 08 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9810F0070E645
STAGE 09 COMPLETE. COMPCODE : 0000 REASON CODE : 0000 CSQ4IVP1.BAB9812BA8706803
STAGE 10 COMPLETE. COMPCODE : 0000 REASON CODE : 0000
>>>>>>>>>> END OF REPORT <<<<<<<<<<<<

```

図 101. CSQ4IVP1 からのサンプル・レポート

## z/OS キュー共有グループのテスト

基本インストール検査プログラム CSQ4IVP1 は非共有キューをテストします。

CSQ4IVP1 は、キュー・マネージャーがキュー共有グループのメンバーであるかどうかにかかわらず、使用することができます。基本 IVP を実行した後、別のキューで CSQ4IVP1 インストール検査プログラムを使用して、共有キューのテストを行うことができます。また、Db2 とカップリング・ファシリティが正しく設定されているかどうかをテストします。

### キュー共有グループ用の CSQ4IVP1 の実行準備

CSQ4IVP1 を実行する前に、次のことを行ってください。

- 710 ページの『[カップリング・ファシリティのセットアップ](#)』に説明されているとおり、IVP が使用するカップリング・ファシリティ構造を CFRM ポリシー・データ・セットに追加します。提供されているサンプルでは、APPLICATION1 という構造を使用しますが、必要に応じて変更することもできます。
- IVP 項目が、キュー・マネージャー始動プログラムの中の CSQINP2 データ・セット連結に入っているかどうか検査します。この IVP 項目は、メンバー thlqual.SCSQPROC(CSQ4IVPG) に入って提供されます。入っていない場合には、thlqual.SCSQPROC(CSQ4IVPG) で提供された定義を、CSQINP2 連結に追加してください。キュー・マネージャーが現在実行されている場合、これらの定義を有効にするために、キュー・マネージャーを再始動する必要があります。
- 必要であれば、thlqual.SCSQPROC(CSQ4IVPG) で使用されているカップリング・ファシリティ構造の名前を変更します。
- キュー共有グループ用のインストール検査プログラムを実行するために必要なサンプル JCL の CSQ4IVPS は、ライブラリー thlqual.SCSQPROC に入っています。  
CSQ4IVPS JCL をカスタマイズするため、IBM MQ ライブラリーの高位修飾子、使用する各国語、4 文字の IBM MQ キュー・マネージャー名、およびジョブ出力の宛先を指定します。
- RACF を更新して、IBM MQ セキュリティが活動状態の場合に CSQ4IVP1 が RACF の資源にアクセスできるようにします。

IBM MQ のセキュリティが使用可能になったときに CSQ4IVP1 を実行するには、オブジェクトへのアクセス権を持っている RACF ユーザー ID が必要です。RACF への資源の定義の詳細は、[z/OS でのセキュリティのセットアップ](#)を参照してください。IVP を実行するユーザー ID には、基本 IVP を実行するのに必要な権限に加えて、次のようなアクセス権が必要です。

Authority	プロファイル	Class
UPDATE	ssid.CSQ4IVPG.**	MQQUEUE

上記の要件は、すべての IBM MQ セキュリティーが活動状態であることを想定しています。IBM MQ のセキュリティを活動化するための RACF コマンドは、725 ページの図 102 に示してあります。この例では、キュー・マネージャー名が CSQ1 であること、およびサンプル CSQ4IVP1 を実行するユーザーのユーザー ID が TS101 であることを前提としています。

```
RDEFINE MQQUEUE CSQ1.CSQ4IVPG.**
PERMIT CSQ1.CSQ4IVPG.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)
```

図 102. キュー共有グループ用の CSQ4IVP1 のための RACF コマンド

## キュー共有グループ用の CSQ4IVP1 の実行

これらのステップを完了したら、キュー・マネージャーを開始してください。キュー・マネージャーが既に実行中で、かつ CSQINP2 を変更した場合は、キュー・マネージャーを停止して再始動する必要があります。

IVP はバッチ・ジョブとして実行されます。ジョブ・カードを、インストール先のジョブ実行依頼の要件に合うようにカスタマイズしてください。

## キュー共有グループ用の CSQ4IVP1 の実行結果の検査

キュー共有グループ用の IVP は、作成されるキューが CSQIVPG と呼ばれることを除いて、基本 IVP と同じように動作します。xx. 722 ページの『CSQ4IVP1 の実行結果の検査』にある指示に従って、キュー共有グループ用の IVP の実行結果を検査してください。

### 分散キューイングのテスト

提供されたインストール検査プログラム CSQ4IVPX を使用すると、分散キューイングが操作可能になっているかどうかを確認できます。

## CSQ4IVPX ジョブの概要

CSQ4IVPX は、チャンネル・イニシエーターを開始し、IBM MQ DISPLAY CHINIT コマンドを発行するバッチ・ジョブです。これは、チャンネルやネットワークの定義を設定しなくても、分散キューイングの主要なすべての部分が操作可能になっているかどうかを検査します。

## CSQ4IVPX の実行準備

CSQ4IVPX を実行する前に、次のことを行ってください。

1. インストール検査プログラムを実行するために必要なサンプル JCL の CSQ4IVPX は、ライブラリー `thlqual.SCSQPROC` に入っています。

CSQ4IVPX JCL をカスタマイズするため、IBM MQ ライブラリーの高位修飾子、使用する各国語、4 文字のキュー・マネージャー名、およびジョブ出力の宛先を指定します。

2. RACF を更新して、IBM MQ セキュリティーが活動状態の場合に CSQ4IVPX が RACF の資源にアクセスできるようにします。IBM MQ のセキュリティが使用可能になったときに CSQ4IVPX を実行するには、オブジェクトへのアクセス権限を持っている RACF ユーザー ID が必要です。RACF への資源の定義の詳細は、[z/OS でのセキュリティのセットアップ](#)を参照してください。IVP を実行するユーザー ID には、次のようなアクセス権が必要です。

Authority	プロファイル	Class
CONTROL	ssid.START.CHINIT および ssid.STOP.CHINIT	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CSQUUTIL.*	MQQUEUE
READ	ssid.BATCH	MQCONN
READ	ssid.DISPLAY.CHINIT	MQCMDS

上記の要件では、接続セキュリティ・プロファイル `ssid.CHIN` が定義されていて (チャンネル・イニシエーターのための接続セキュリティ・プロファイルを参照)、しかもすべての IBM MQ セキュリティが活動状態になっているものと想定しています。この作業を行うための RACF コマンドを、[727 ページの図 103](#) に示します。この例では、次のことを前提としています。

- キュー・マネージャー名が CSQ1 である
  - サンプルの CSQ4IVPX を実行しているユーザーのユーザー ID が TS101 である
  - チャンネル・イニシエーター・アドレス・スペースは、ユーザー ID CSQ1MSTR の下で実行されている
3. RACF を更新して、チャンネル・イニシエーター・アドレス・スペースに以下のアクセス権限を許可します。

Authority	プロファイル	Class
READ	ssid.CHIN	MQCONN
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.INITQ	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.SYNCQ	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.COMMAND.QUEUE	MQQUEUE
UPDATE	ssid.SYSTEM.CLUSTER.TRANSMIT.QUEUE	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.REPOSITORY.QUEUE	MQQUEUE
CONTROL	ssid.CONTEXT.**	MQADMIN

この作業を行うための RACF コマンドも、[727 ページの図 103](#) に示します。

```

RDEFINE MQCMDS CSQ1.DISPLAY.DQM
PERMIT CSQ1.DISPLAY.DQM CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQCMDS CSQ1.START.CHINIT
PERMIT CSQ1.START.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQCMDS CSQ1.STOP.CHINIT
PERMIT CSQ1.STOP.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101,CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CSQUTIL.*
PERMIT CSQ1.SYSTEM.CSQUTIL.* CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)

RDEFINE MQCONN CSQ1.CHIN
PERMIT CSQ1.CHIN CLASS(MQCONN) ID(CSQ1MSTR) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.SYNCQ
PERMIT CSQ1.SYSTEM.CHANNEL.SYNCQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.INITQ
PERMIT CSQ1.SYSTEM.CHANNEL.INITQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQADMIN CSQ1.CONTEXT.**
PERMIT CSQ1.CONTEXT.** CLASS(MQADMIN) ID(CSQ1MSTR) ACCESS(CONTROL)

```

図 103. CSQ4IVPX のための RACF コマンド

## CSQ4IVPX の実行

これらのステップを完了したら、キュー・マネージャーを開始してください。

IVP はバッチ・ジョブとして実行されます。ジョブ・カードを、インストール先のジョブ実行依頼の要件に合うようにカスタマイズしてください。

## CSQ4IVPX の実行結果の検査

CSQ4IVPX は CSQUTIL IBM MQ ユーティリティを実行して、3つの MQSC コマンドを発行します。SYSPRINT 出力データ・セットは、728 ページの図 104 のようにしてください。ただし、使用するキュー・マネージャーの属性によって詳細は異なります。

- コマンド **(1)** が表示され、それぞれにいくつかのメッセージが続きます。
- 各コマンドの最後のメッセージは次のようになります"CSQ9022I... NORMAL COMPLETION" **(2)**。
- 全体としてのジョブは、戻りコード **0 (3)** で完了します。

```

CSQU000I CSQUTIL IBM MQ for z/OS - V6
CSQU001I CSQUTIL Queue Manager Utility - 2005-05-09 09:06:48
COMMAND
CSQU127I CSQUTIL Executing COMMAND using input from CSQUCMD data set
CSQU120I CSQUTIL Connecting to queue manager CSQ1
CSQU121I CSQUTIL Connected to queue manager CSQ1
CSQU055I CSQUTIL Target queue manager is CSQ1
START CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM138I +CSQ1 CSQMSCHI CHANNEL INITIATOR STARTING
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' START CHINIT' NORMAL COMPLETION
(2)
DISPLAY CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMDDQM DISPLAY CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 12, RETURN=00000000, REASON=00000000
CSQX830I +CSQ1 CSQXRQDM Channel initiator active
CSQX002I +CSQ1 CSQXRQDM Queue sharing group is QSG1
CSQX831I +CSQ1 CSQXRQDM 8 adapter subtasks started, 8 requested
CSQX832I +CSQ1 CSQXRQDM 5 dispatchers started, 5 requested
CSQX833I +CSQ1 CSQXRQDM 0 SSL server subtasks started, 0 requested
CSQX840I +CSQ1 CSQXRQDM 0 channel connections current, maximum 200
CSQX841I +CSQ1 CSQXRQDM 0 channel connections active, maximum 200,
including 0 paused
CSQX842I +CSQ1 CSQXRQDM 0 channel connections starting,
0 stopped, 0 retrying
CSQX836I +CSQ1 Maximum channels - TCP/IP 200, LU 6.2 200
CSQX845I +CSQ1 CSQXRQDM TCP/IP system name is TCPIP
CSQX848I +CSQ1 CSQXRQDM TCP/IP listener INDISP=QMGR not started
CSQX848I +CSQ1 CSQXRQDM TCP/IP listener INDISP=GROUP not started
CSQX849I +CSQ1 CSQXRQDM LU 6.2 listener INDISP=QMGR not started
CSQX849I +CSQ1 CSQXRQDM LU 6.2 listener INDISP=GROUP not started
CSQ9022I +CSQ1 CSQXCRPS ' DISPLAY CHINIT' NORMAL COMPLETION
(2)
STOP CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMTCHI STOP CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' STOP CHINIT' NORMAL COMPLETION
(2)
CSQU057I CSQUCMDS 3 commands read
CSQU058I CSQUCMDS 3 commands issued and responses received, 0 failed
CSQU143I CSQUTIL 1 COMMAND statements attempted
CSQU144I CSQUTIL 1 COMMAND statements executed successfully
CSQU148I CSQUTIL Utility completed, return code=0
(3)

```

図 104. CSQ4IVPX からの出力の例

## z/OS IBM MQ for z/OS を使用した C、C++、COBOL、PL/I、および CICS プログラムのテスト

IBM MQ で提供されているサンプル・アプリケーションを使用して、C、C++、COBOL、PL/I、または CICS のいずれかをテストすることができます。

IVP (CSQ4IVP1) はロード・モジュールとして提供されますが、ソース・モジュールとしてサンプルを提供します。これらのソース・モジュールを使用して、さまざまなプログラミング言語環境をテストすることができます。

サンプル・アプリケーションについて詳しくは、[IBM MQ for z/OS のサンプル・プログラム](#)を参照してください。

## z/OS 別のキュー・マネージャーとの通信のセットアップ

このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

分散キューイングの要件定義では、以下の項目の定義が必要です。

- チャンネル・イニシエーターのプロシージャーとデータ・セットの定義
- チャンネル定義の定義
- キューおよびその他のオブジェクトの定義
- アクセス・セキュリティの定義

分散キューイングを使用可能にするには、以下に示す3つの作業を実行する必要があります。

- 分散キューイング機能をカスタマイズし、IBM MQ の必須オブジェクトを定義する ([システム・オブジェクトの定義](#)および [657 ページの『z/OSでのキュー・マネージャーのカスタマイズ準備』](#)を参照)。
- アクセス・セキュリティを定義する ([z/OS で使用するチャンネル・イニシエーターのセキュリティに関する考慮事項](#)を参照)。
- [749 ページの『z/OSにおける通信のセットアップ』](#)の説明に従って、通信をセットアップします。

キュー共用グループを使用している場合は、[分散キューイングとキュー共用グループ](#)を参照してください。

IBM MQ for z/OS で分散キューイングを使用する場合の追加の考慮事項については、以下のセクションを参照してください。

## オペレーター・メッセージ

チャンネル・イニシエーターは、非同期的に稼働するディスパッチャーをいくつかを使用するため、ログに表示されるオペレーター・メッセージは発生順ではないことがあります。

## チャンネル操作コマンド

チャンネル操作コマンドには、一般に2つのステージが関係します。コマンド構文を検査し、チャンネルの存在が確認されたら、要求はチャンネル・イニシエーターに送信されます。メッセージ CSQM134I または CSQM137I が コマンド発行者に送信され、最初のステージが完了したことが示されます。チャンネル・イニシエーターがコマンドを処理したら、その処理の成否を示す別のメッセージが、メッセージ CSQ9022I または CSQ9023I と共に、コマンド発行者に送信されます。エラー・メッセージが生成される場合には、z/OS コンソールにも送信されます。

ただし、DISPLAY CLUSQMGR 以外のクラスター・コマンドはすべて、非同期に機能します。オブジェクト属性を変更するコマンドは、オブジェクトを更新し、要求をチャンネル・イニシエーターに送信します。クラスターを操作するコマンドは、構文が検査され、要求がチャンネル・イニシエーターに送信されます。どちらのケースでも、要求が送信されたことを示すメッセージ CSQM130I がコマンド発行者に送信されます。このメッセージには、メッセージ CSQ9022I が続きます。これは、要求が送信されたという点でコマンドが正常に完了したことを示しています。これは、クラスター要求が正常に完了したことを示すものではありません。チャンネル・イニシエーターに送信された要求は、クラスターの他のメンバーから受信したクラスター要求と共に、非同期で処理されます。場合によっては、この要求をクラスター全体に送信して、成功したかどうかを判断する必要があります。エラーがある場合は、チャンネル・イニシエーターが稼働しているシステムの z/OS に報告されます。コマンド発行者には送信されません。

## 未配布メッセージ・キュー

IBM MQ for z/OS には送達不能ハンドラーが備わっています。詳しくは、[送達不能キュー・ハンドラー・ユーティリティ \(CSQUDLQH\)](#) を参照してください。

## 使用中のキュー

受信側チャンネルの MCA は、メッセージが伝送されていないときでも宛先キューをオープンしておくことができます。この動作により、キューが「使用中」として見えるようになります。

## セキュリティの変更

ユーザー ID に対するセキュリティ・アクセスを変更しても、それがすぐに有効になるわけではありません。(詳しくは、[z/OS で使用するチャンネル・イニシエーターのセキュリティに関する考慮事項](#)、[キュー・](#)

セキュリティーのためのプロファイル、または 673 ページの『ESM セキュリティー管理を実施する』のいずれかを参照してください。

## 通信の停止 - TCP

何らかの理由で TCP が停止されてから再始動される場合は、TCP ポート上で待機中の IBM MQ for z/OS TCP リスナーは停止されます。

自動チャンネル再接続では、チャンネル・イニシエーターが TCP/IP が使用不可であることを検知し、TCP/IP の戻り時に TCP/IP リスナーを自動的に再始動することができます。この自動再始動によって、TCP/IP の問題を検知してリスナーを手動で再始動するという操作スタッフの責務が軽減されます。リスナーが動作していない間に、チャンネル・イニシエーターを使用して、LSTRTMR で指定された間隔でリスナーを再試行することもできます。TCP/IP が戻ってリスナーが正常に再始動するまで、この試みを続けることができます。LSTRTMR については、[ALTER QMGR](#) および [分散キューイング・メッセージ \(CSQX ...\)](#) を参照してください。

## 通信の停止 - LU6.2

APPC が停止した場合は、リスナーも停止します。この場合も、LSTRTMR 間隔でリスナーの自動再試行が行われるので、APPC が再始動するとリスナーも再始動することができます。

Db2 に障害が起きた場合、既に実行中であった共有チャンネルの実行は継続されますが、以後のチャンネル開始要求はすべて失敗します。Db2 が復元されたら、新規の要求を完了することができます。

## z/OS 自動再始動管理 (ARM)

自動再始動管理 (ARM) は、z/OS のリカバリー機能の 1 つです。この機能を利用すると、特定のバッチ・ジョブまたは開始済みタスク (例えば、サブシステム) の可用性を向上させることができます。これによって、生産性の高い作業をより速やかに再開することができます。

ARM を使用するには、キュー・マネージャーとチャンネル・イニシエーターが自動的に再始動されるように設定する必要があります。詳しくは、[z/OS 自動再始動マネージャー \(ARM\)](#) の使用を参照してください。

### 関連概念

[661 ページの『IBM MQ for z/OS のセットアップ』](#)

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

[732 ページの『z/OS におけるチャンネルのモニターと制御』](#)

リモート・キュー・マネージャーにつながるチャンネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。

[749 ページの『z/OS における通信のセットアップ』](#)

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。このセクションでは、接続を定義する方法を説明します。

[753 ページの『キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備』](#)

このセクションの説明を使用して、IBM MQ for z/OS で、キュー共有グループを使用した分散キューイングを構成します。

[758 ページの『キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ』](#)

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。この試行を正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

### 関連タスク

[140 ページの『分散キューイングの構成』](#)

このセクションでは、キュー定義、チャンネル定義、トリガー操作および同期点プロシーチャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

IBM MQ コマンド入力方式の 1 つを利用して、IBM MQ オブジェクトを定義します。これらのオブジェクトの定義についての詳細は、このトピック内の情報を参照してください。

オブジェクトの定義については、732 ページの『z/OS におけるチャンネルのモニターと制御』を参照してください。

## 伝送キューおよびチャンネルのトリガー操作

以下のものを定義します。

- それぞれの送信側メッセージ・チャンネルごとに、用途として XMITQ を指定したローカル・キューを 1 つずつ。
- リモート・キュー定義。

リモート・キュー・オブジェクトは、名前および内容の指定方法に応じて、3 つの個別の用途が与えられます。

- リモート・キュー定義
- キュー・マネージャー別名定義
- 応答先キュー別名定義

これらの 3 つの方法は、[リモート・キュー定義オブジェクトの 3 つの使用方法](#)に示されています。

伝送キューの TRIGDATA フィールドを使用して、指定したチャンネルをトリガー操作します。以下に例を示します。

```
DEFINE QLOCAL(MYXMITQ) USAGE(XMITQ) TRIGGER +  
INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(MYCHANNEL)  
DEFINE CHL(MYCHANNEL) CHLTYPE(SDR) TRPTYPE(TCP) +  
XMITQ(MYXMITQ) CONNAME('9.20.9.30(1555)')
```

システムから提供されるサンプル CSQ4INXD は、必要な定義について別の例を提供しています。

共有チャンネルの同期キューが定義されている CF 構造への接続が失われるか、同種の問題が発生すると、一時的にチャンネルが開始できなくなる可能性があります。問題が解決した後、トリガー・タイプ FIRST を使用している場合にそのトリガー時にチャンネルを開始できない場合、チャンネルを手動で開始する必要があります。問題が解決した後に、トリガーされたチャンネルが自動的に開始されるようにする場合、キュー・マネージャーの TRIGINT 属性をデフォルト以外の値に設定することを検討してください。TRIGINT 属性をデフォルト以外の値に設定すると、伝送キューにメッセージがある間に、チャンネル・イニシエーターがチャンネルの開始を定期的に再試行するようになります。

## 同期キュー

DQM には、シーケンス番号および作業論理単位 ID (LUWID) で使用するためのキューが必要です。SYSTEM.CHANNEL.SYNCQ という名前のキューを利用できるようにする必要があります (z/OS での計画を参照してください)。このキューが使用可能になっていないと、チャンネル・イニシエーターを開始できません。

このキューは、必ず INDXTYPE(MSGID) を使用して定義してください。この属性により、このキューに対するアクセス速度が向上します。

## チャンネル・コマンド・キュー

使用するシステムには必ず SYSTEM.CHANNEL.INITQ という名前のチャンネル・コマンド・キューを用意する必要があります。

チャンネル・イニシエーターが SYSTEM.CHANNEL.INITQ に問題があることを検出すると、その問題が訂正されるまで、正常に継続することができません。問題は、次のいずれかが原因だと考えられます。

- キューが満杯である。

- キューは書き込みができない。
- キューが入っているページ・セットが満杯である。
- 当該チャンネル・イニシエーターがキューに対して正しいセキュリティー権限をもっていない。

チャンネル・イニシエーターの実行中に、キューの定義が GET(DISABLED) に変更されると、チャンネル・イニシエーターはそのキューからメッセージを読み取ることができなくなり、終了します。

## チャンネル・イニシエーターの開始

トリガー操作は、チャンネル・イニシエーターによって実行されます。IBM MQ for z/OS では、イニシエーターは MQSC コマンドの START CHINIT で開始されます。

## チャンネル・イニシエーターの停止

チャンネル・イニシエーターは、キュー・マネージャーを停止すると自動的に停止されます。ただし、チャンネル・イニシエーターを停止しなければならないが、キュー・マネージャーは停止しなくてもよい場合は、MQSC コマンド STOP CHINIT を使用できます。

## **z/OS** z/OS におけるチャンネルのモニターと制御

リモート・キュー・マネージャーにつながるチャンネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。

各 z/OS キュー・マネージャーには、本来の z/OS 機能を使用してリモート・キュー・マネージャーとの相互接続を制御するための DQM プログラム (チャンネル・イニシエーター) が備わっています。

z/OS におけるこれらのパネルとコマンドは、操作および制御パネルと MQSC コマンドに組み込まれて実装されます。これらのパネルとコマンドの 2 つのセットには、編成上の違いはありません。

プログラマブル・コマンド・フォーマット (PCF) コマンドを使用してコマンドを入力することもできます。これらのコマンドの使用については、[管理タスクの自動化](#)を参照してください。

このセクションの情報は、分散キューイングでチャンネル・イニシエーターが使用されるすべてのケースに適用されます。キュー共有グループを使用するかグループ内キューイングを使用するかにかかわらず適用されます。

## DQM チャンネル制御機能

分散キューの管理モデルの概要については、[162 ページの『メッセージの送受信』](#)を参照してください。

チャンネル制御機能は、パネル、コマンド、プログラム、同期キュー、チャンネル・コマンド・キュー、およびチャンネル定義で構成されます。このトピックでは、チャンネル制御機能のコンポーネントについて簡単に説明します。

- チャンネル定義は、z/OS の他の IBM MQ オブジェクトと同様に、ページ・セット 0 または Db2 にオブジェクトとして保存されます。
- 操作および制御パネル、MQSC コマンド、または PCF コマンドを使用する目的は以下のとおりです。
  - チャンネル定義の作成、コピー、表示、変更、および削除
  - チャンネル・イニシエーターおよびリスナーの開始および停止
  - リンクを再確立できない場合の、チャンネルの開始、停止、ping、チャンネル・シーケンス番号のリセット、および未確定メッセージの解決
  - チャンネルに関する状況情報の表示
  - DQM に関する情報の表示

特に CSQINPX 初期設定の入力データ・セットを使用して、MQSC コマンドを出すことができます。このセットは、チャンネル・イニシエーターを開始するたびに処理することができます。詳しくは、[初期化コマンド](#)を参照してください。

- チャンネルの再同期に使われる 2 つのキュー (SYSTEM.CHANNEL.SYNCQ および SYSTEM.QSG.CHANNEL.SYNCQ) があります。パフォーマンス上の理由から、INDXTYPE(MSGID) を使用してこれらのキューを定義してください。
- チャンネル・イニシエーター、チャンネル、およびリスナーに対するコマンドを保持するために、チャンネル・コマンド・キュー (SYSTEM.CHANNEL.INITQ) が使用されます。
- チャンネル制御機能プログラムは、キュー・マネージャーとは異なる独自のアドレス・スペースで実行されます。チャンネル制御機能プログラムは、チャンネル・イニシエーター、リスナー、MCA、トリガー・モニターおよびコマンド・ハンドラーから構成されます。
- キュー共用グループおよび共用チャンネルの場合は、共用キューとキュー共用グループを参照してください。
- グループ内キューイングについては、グループ内キューイングを参照してください

## z/OS におけるチャンネルの管理

チャンネル、チャンネル・イニシエーター、およびリスナーを管理する方法についての詳細は、以下の表のリンクを使用します。

表 49. チャンネル操作	
実行される操作	MQSC コマンド
<a href="#">チャンネルの定義</a>	DEFINE CHANNEL
<a href="#">チャンネル定義の変更</a>	ALTER CHANNEL
<a href="#">チャンネル定義の表示</a>	DISPLAY CHANNEL
<a href="#">チャンネル定義の削除</a>	DELETE CHANNEL
<a href="#">チャンネル・イニシエーターの開始</a>	START CHINIT
<a href="#">チャンネル・イニシエーターの停止</a>	STOP CHINIT
<a href="#">チャンネル・イニシエーター情報の表示</a>	DISPLAY CHINIT
<a href="#">チャンネル・リスナーの開始</a>	START LISTENER
<a href="#">チャンネル・リスナーの停止</a>	STOP LISTENER
<a href="#">チャンネルの開始</a>	START CHANNEL
<a href="#">チャンネルのテスト</a>	PING CHANNEL
<a href="#">チャンネルのメッセージ・シーケンス番号のリセット</a>	RESET CHANNEL
<a href="#">チャンネルの未確定メッセージの解決</a>	RESOLVE CHANNEL
<a href="#">チャンネルの停止</a>	STOP CHANNEL
<a href="#">チャンネル状況の表示</a>	DISPLAY CHSTATUS
<a href="#">クラスター・チャンネルの表示</a>	DISPLAY CLUSQMGR

### 関連概念

734 ページの『[パネルおよびコマンドの使用法](#)』

MQSC コマンド、PCF コマンド、または操作および制御パネルを使用して DQM を管理できます。

728 ページの『[他のキュー・マネージャーとの通信のセットアップ](#)』

このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

661 ページの『[IBM MQ for z/OS のセットアップ](#)』

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

#### 749 ページの『z/OS における通信のセットアップ』

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。このセクションでは、接続を定義する方法を説明します。

#### 753 ページの『キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備』

このセクションの説明を使用して、IBM MQ for z/OS で、キュー共有グループを使用した分散キューイングを構成します。

#### 758 ページの『キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ』

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。この試行を正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

### 関連タスク

#### 140 ページの『分散キューイングの構成』

このセクションでは、キュー定義、チャンネル定義、トリガー操作および同期点プロシージャを含む、IBM MQ インストール間の相互通信の詳細について説明します。

## z/OS パネルおよびコマンドの使用法

MQSC コマンド、PCF コマンド、または操作および制御パネルを使用して DQM を管理できます。

MQSC コマンドの構文の詳細は、[スクリプト \(MQSC\) コマンド](#)を参照してください。PCF コマンドについては、[プログラマブル・コマンド・フォーマットの概要](#)を参照してください。

### 初期パネルの使用

操作および制御パネルの呼び出し、ファンクション・キーの使用、およびヘルプの表示を行う方法の概要については、[IBM MQ for z/OS の管理](#)を参照してください。

**注：**操作および制御パネルを使用するには、正しいセキュリティ許可が必要です。詳しくは、[IBM MQ for z/OS の管理](#)およびサブピックを参照してください。734 ページの図 105 は、パネル・セッションを開始したときに表示されるパネルを示しています。パネルの後のテキストでは、このパネルで実行するアクションを説明します。

```
IBM MQ for z/OS - Main Menu
Complete fields. Then press Enter.

Action . . . . . 1 0. List with filter 4. Manage
1. List or Display 5. Perform
2. Define like 6. Start
3. Alter 7. Stop
8. Command
Object type . . . . . CHANNEL +
Name . . . . . *
Disposition . . . . . A Q=Qmgr, C=Copy, P=Private, G=Group,
S=Shared, A=All

Connect name . . . . . MQ25 - local queue manager or group
Target queue manager . . . MQ25
- connected or remote queue manager for command input
Action queue manager . . . MQ25 - command scope in group
Response wait time . . . . 10 5 - 999 seconds

(C) Copyright IBM Corporation 1993, 2023. All rights reserved.

Command ==> _____
F1=Help F2=Split F3=Exit F4=Prompt F9=SwapNext F10=Messages
F12=Cancel
```

図 105. 操作および制御初期パネル

このパネルから以下のことを行うことができます。

- 「**Action (アクション)**」フィールドに適切な番号を入力することによって、実行したいアクションを選択します。

- 作業を進めたいオブジェクト・タイプを指定します。オブジェクト・タイプが分からない場合は、F4 キーを押してオブジェクト・タイプのリストを表示します。
- 指定されたタイプのオブジェクトのリストを表示します。「Name (名前)」フィールドにアスタリスク (\*) を入力し、Enter キーを押して、このサブシステムですでに定義されているオブジェクト (指定したタイプ) のリストを表示します。次に、1つまたは複数のオブジェクトを選択して、それらを順次に処理できます。735 ページの図 106 は、この方法で生成されたチャンネルのリストを示しています。
- 処理したいオブジェクトのキュー共有グループにおける属性指定を「Disposition (属性指定)」フィールドに指定します。オブジェクトの保管先とオブジェクトの動作は、属性指定によって決まります。
- 接続先にしたいローカル・キュー・マネージャーまたはキュー共有グループを「Connect name (接続名)」フィールドで選びます。リモート・キュー・マネージャーでコマンドを出したければ、そのリモート・キュー・マネージャーがキュー共有グループのメンバーであるかどうかに応じて、「Target queue manager (宛先 QMGR)」フィールドまたは「Action queue manager (アクション QMGR)」フィールドを選びます。リモート・キュー・マネージャーがキュー共有グループのメンバーでなければ、「Target queue manager (宛先 QMGR)」フィールドを選びます。リモート・キュー・マネージャーがキュー共有グループのメンバーであれば、「Action queue manager (アクション QMGR)」フィールドを選びます。
- 応答を受信するまでの待ち時間を「Response wait time (応答待ち時間)」フィールドで選びます。

List Channels - MQ25

Row 1 of 8

Type action codes, then press Enter. Press F11 to display connection status.

1=Display 2=Define like 3=Alter 4=Manage 5=Perform  
6=Start 7=Stop

```
Name          Type      Disposition Status
<> *          CHANNEL  ALL      MQ25
- SYSTEM.DEF.CLNTCONN CLNTCONN  QMGR  MQ25
- SYSTEM.DEF.CLUSRCVR CLUSRCVR  QMGR  MQ25 INACTIVE
- SYSTEM.DEF.CLUSSDR  CLUSSDR   QMGR  MQ25 INACTIVE
- SYSTEM.DEF.RECEIVER RECEIVER   QMGR  MQ25 INACTIVE
- SYSTEM.DEF.REQUESTER REQUESTER QMGR  MQ25 INACTIVE
- SYSTEM.DEF.SENDER   SENDER    QMGR  MQ25 INACTIVE
- SYSTEM.DEF.SERVER   SERVER     QMGR  MQ25 INACTIVE
- SYSTEM.DEF.SVRCONN  SVRCONN   QMGR  MQ25 INACTIVE
***** End of list *****
```

Command ==>

F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd  
F8=Fwd F9=SwapNext F10=Messages F11=Status F12=Cancel

図 106. チャンネルのリスト

## z/OS z/OS でのチャンネルの定義

z/OS では、MQSC コマンドを使用して、または操作パネルと制御パネルを使用して、チャンネルを定義できます。

MQSC コマンドを使用してチャンネルを定義するには、DEFINE CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	2 (類似定義)
オブジェクト・タイプ	チャンネル・タイプ (SENDER など) または CHANNEL プ
名前	
後処理	新規オブジェクトのロケーション

いくつかのパネルが表示されるので、定義するチャンネルに指定したい名前と属性に関する情報を入力します。それらは、デフォルトの属性値で初期化されています。任意の箇所を変更してから、Enter キーを押します。

**注: object type** フィールドに CHANNEL を入力した場合、最初に「Select a Valid Channel Type (有効なチャンネル・タイプの選択)」パネルが表示されます。

既存のチャンネルと同じ属性を持つチャンネルを定義する場合は、コピーするチャンネルの名前を初期パネルの「Name」フィールドに入力します。パネルは、既存オブジェクトの属性で初期化されます。

チャンネル属性の詳細は、[チャンネル属性](#)を参照してください。

**注:**

1. ネットワーク内のすべてのチャンネルに固有の名前を付けてください。このネーミングのためには、すべてのチャンネルを示すネットワーク・ダイアグラムに示すように、ソースおよびターゲットのキュー・マネージャーの名前をチャンネル名に取り入れるのはよい方法です。

チャンネルを定義した後、チャンネルを保護する必要があります。[737 ページの『チャンネルの保護』](#)を参照してください

### **チャンネル定義の変更**

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネル定義を変更できます。

MQSC コマンドを使用してチャンネル定義を変更するには、ALTER CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	3 (変更)
オブジェクト・タイプ	チャンネル・タイプ (SENDER など) または CHANNEL プ
名称	CHANNEL.TO.ALTER
後処理	保管されているオブジェクトのロケーション

チャンネルの現行の属性に関する情報が入ったパネルがいくつか表示されます。必要に応じて、無保護フィールドを新しい値で指定変更し、Enter キーを押して、チャンネル定義を変更してください。

チャンネル属性の詳細は、[チャンネル属性](#)を参照してください。

### **チャンネル定義の表示**

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネル定義を表示できます。

MQSC コマンドを使用してチャンネル定義を表示するには、DISPLAY CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	1 (リストまたは表示)
オブジェクト・タイプ	チャンネル・タイプ (SENDER など) または CHANNEL
名称	CHANNEL.TO.DISPLAY
後処理	オブジェクトのロケーション

チャンネルの現行の属性に関する情報を示すパネルがいくつか表示されます。

チャンネル属性の詳細は、[チャンネル属性](#)を参照してください。

## チャンネル定義の削除

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネル定義を削除できます。

MQSC コマンドを使用してチャンネル定義を削除するには、DELETE CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	4 (管理)
オブジェクト・タイプ	チャンネル・タイプ (SENDER など) または CHANNEL
名称	CHANNEL.TO.DELETE
後処理	オブジェクトのロケーション

別のパネルが表示されます。このパネルで機能タイプ 1 を選択します。

Enter キーを押してチャンネル定義を削除しようとする、チャンネル定義を削除したいことを確認するためにもう一度 Enter キーを押すよう指示されます。

注：チャンネル定義を削除できるようにするためには、チャンネル・イニシエーターを実行させておく必要があります (クライアント接続チャンネルを除く)。

## チャンネル・イニシエーターに関する情報の表示

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネル・イニシエーターに関する情報を表示できます。

MQSC コマンドを使用して、チャンネル・イニシエーターに関する情報を表示するには、DISPLAY CHINIT を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	1 (表示)
オブジェクト・タイプ	SYSTEM
名前	ブランク

別のパネルが表示されます。このパネルで機能タイプ 1 を選択します。

注：

1. 多数のチャンネルがあると、分散キューイング情報の表示に時間がかかることがあります。
2. 分散キューイングの情報を表示するには、先にチャンネル・イニシエーターを実行しておく必要があります。

## チャンネルの保護

MQSC コマンドを使用して、または操作パネルと制御パネルを使用して、チャンネルを保護できます。

MQSC コマンドを使用してチャンネルを保護するには、[SET CHLAUTH](#) を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	8

MQSC コマンドを提供できるエディターが表示されます。この場合は、CHLAUTH コマンドを使用してください。738 ページの図 107 を参照してください。コマンド入力の最後には正符号 (+) が必要です。PF3 と入力してこのエディターを終了し、コマンドをコマンド・サーバーに実行依頼します。

```
***** Top of Data *****
000001 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000002 TYPE(SSLPEERMAP) +
000003 SSLPEER('CN="John Smith"') +
000004 MCAUSER('PUBLIC')
***** Bottom of Data *****

Command ==>          Scroll ==> PAGE
F1=Help  F3=Exit  F4=LineEdit F12=Cancel
```

### 図 107. コマンド入力

次に、コマンドの出力が表示されます。738 ページの図 108 を参照してください

```
***** Top of Data *****
000001 CSQU000I CSQUTIL IBM MQ for z/OS 7.1.0
000002 CSQU001I CSQUTIL Queue Manager Utility - 2011-04-20 14:42:58
000003 COMMAND TGTQMGR(MQ23) RESPTIME(30)
000004 CSQU127I Executing COMMAND using input from CSQUCMD data set
000005 CSQU120I Connecting to MQ23
000006 CSQU121I Connected to queue manager MQ23
000007 CSQU055I Target queue manager is MQ23
000008 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000009 TYPE(SSLPEERMAP) +
000010 SSLPEER('CN="John Smith"') +
000011 MCAUSER('PUBLIC')
000012 CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
000013 CSQ9022I !MQ23 CSQMCA ' SET CHLAUTH' NORMAL COMPLETION
000014 CSQU057I 1 commands read
000015 CSQU058I 1 commands issued and responses received, 0 failed
000016 CSQU143I 1 COMMAND statements attempted
000017 CSQU144I 1 COMMAND statements executed successfully
000018 CSQU148I CSQUTIL Utility completed, return code=0
Command ==>          Scroll ==> PAGE
F1=Help  F3=Exit  F5=Rfind  F6=Rchange  F9=SwapNext F12=Cancel
```

### 図 108. コマンド出力

## チャネル・イニシエーターの開始

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャネル・イニシエーターを開始できます。

MQSC コマンドを使用してチャネル・イニシエーターを開始するには、START CHINIT を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	6 (開始)
オブジェクト・タイプ	SYSTEM
名称	ブランク

「Start a System Function) (システム機能の開始)」パネルが表示されます。以下のパネルの後のテキストでは、実行するアクションについて説明します。

## Start a System Function

Select function type, complete fields, then press Enter to start system function.

```
機能タイプ . . . . . _ 1. チャンネル・イニシエーター  
2. Channel listener  
アクション・キュー・マネージャー . . . : MQ25
```

```
チャンネル・イニシエーター  
JCL 置換 . . . -----  
-----
```

```
Channel listener  
インバウンド後処理。 Q G = グループ、 Q=キュー・マネージャー  
トランスポート・タイプ . . . . . _ L=LU6.2、 T=TCP/IP  
LU 名 (LU6.2) . . . . . - _____  
ポート番号 (TCP/IP) 。 1414 年  
IP アドレス (TCP/IP) 。 -----
```

```
コマンド ==> _____  
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel
```

図 109. システム機能の開始

「Function type (機能タイプ)」の「1」(チャンネル・イニシエーター)を選択して、Enter キーを押します。

### チャンネル・イニシエーターの停止

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネル・イニシエーターを停止できます。

MQSC コマンドを使用してチャンネル・イニシエーターを停止するには、STOP CHINIT を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	7 (停止)
オブジェクト・タイプ	SYSTEM
名称	ブランク

「Stop a System Function (システム機能の停止)」パネルが表示されます。パネルの後のテキストでは、このパネルの使用方法を説明します。

## Stop a System Function

Select function type, complete fields, then press Enter to stop system function.

```
機能タイプ . . . . . _ 1. チャンネル・イニシエーター  
2. Channel listener  
アクション・キュー・マネージャー . . . : MQ25
```

```
チャンネル・イニシエーター  
Restart shared channels Y Y=Yes, N=No
```

```
Channel listener  
インバウンド後処理。 Q G = グループ、 Q=キュー・マネージャー  
トランスポート・タイプ . . . . . _ L=LU6.2、 T=TCP/IP
```

```
ポート番号 (TCP/IP) 。 ----- -  
IP アドレス (TCP/IP) 。 -----
```

```
コマンド ==> _____  
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel
```

図 110. 機能制御の停止

「Function type (機能タイプ)」の「1」(チャンネル・イニシエーター)を選択して、Enter キーを押します。

チャンネル・イニシエーターは、実行中のすべてのチャンネルが静止モードで停止するのを待ってから、停止します。

**注:** チャンネルのなかに実行されているがアクティブになっていない受信側チャンネルまたは要求側チャンネルがある場合、受信側または送信側のチャンネル・イニシエーターに出された停止要求によって、そのチャンネルは即座に停止されます。

ただし、次のようなメッセージが表示される場合には、チャンネル・イニシエーターは現行のバッチ・メッセージが完了するのを待ってから、そのチャンネルを停止します。

### チャンネル・リスナーの開始

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネル・リスナーを開始できます。

MQSC コマンドを使用してチャンネル・リスナーを開始するには、START LISTENER を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	6 (開始)
オブジェクト・タイプ	SYSTEM
名称	ブランク

システム機能の開始のパネルが表示されます(739 ページの図 109 を参照)。

機能タイプ 2 (チャンネル・リスナー) を選択します。Inbound disposition (到着属性指定) を選択します。Transport type (トランスポート・タイプ) を選択します。トランスポート・タイプが L の場合、LU 名を選択します。トランスポート・タイプが T の場合、ポート番号と (オプションで) IP アドレスを選択します。Enter キーを押します。

**注:** TCP/IP リスナーの場合、ポートと IP アドレスの複数の組み合わせを開始することができます。

## チャンネル・リスナーの停止

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネル・リスナーを停止できます。

MQSC コマンドを使用してチャンネル・リスナーを停止するには、STOP LISTENER を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	7 (停止)
オブジェクト・タイプ	SYSTEM
名前	ブランク

「システム機能の停止」パネルが表示されます(740 ページの図 110 を参照)。

機能タイプ 2 (チャンネル・リスナー) を選択します。Inbound disposition (到着属性指定) を選択します。Transport type (トランスポート・タイプ) を選択します。トランスポート・タイプが T の場合、ポート番号と (オプションで) IP アドレスを選択します。Enter キーを押します。

注: TCP/IP リスナーの場合、ポートと IP アドレスの組み合わせを個別に停止したりすべての組み合わせを停止したりすることができます。

## チャンネルの開始

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネルを開始できます。

MQSC コマンドを使用してチャンネルを開始するには、START CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	6 (開始)
オブジェクト・タイプ	チャンネル・タイプ (SENDER など) または CHANNEL
名	CHANNEL.TO.USE
後処理	オブジェクトの属性指定

「Start a Channel (チャンネルの開始)」パネルが表示されます。パネルの後のテキストでは、パネルの使用方法を説明します。

Start a Channel

Select disposition, then press Enter to start channel.

チャンネル名 . . . . : CHANNEL.TO.USE  
チャンネル・タイプ . . . . : 送信者  
説明 . . . . : CHANNEL.TO.USE の説明

後処理 . . . . P P=Private on MQ25  
S=Shared on MQ25  
A=Shared on any queue manager

コマンド ==> \_\_\_\_\_  
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel

## 図 111. チャンネルの開始

チャンネル・インスタンスの属性指定と、どのキュー・マネージャーでそれを開始するかを選択します。  
Enter キーを押してチャンネルを開始させます。

### 共有チャンネルの開始

共有チャンネルを開始し、指定されたチャンネル・イニシエーター上でそれを保持するには、属性 = S を使用します (START CHANNEL コマンドで、CHLDISP(FIXSHARED) を指定します)。

一度に実行される共有チャンネルのインスタンスは 1 つのみです。2 つ目のチャンネル・インスタンスを開始しようとしても失敗します。

この方法でチャンネルを開始するときには、以下の規則がそのチャンネルに適用されます。

- キュー共有グループ内のどのキュー・マネージャーからでもチャンネルを停止することができます。チャンネルの停止要求の発行時点に、開始元のチャンネル・イニシエーターが実行されていなかった場合でも停止することができます。チャンネルが停止したとき、同じチャンネル・イニシエーターまたは別のチャンネル・イニシエーターで属性 = S (CHLDISP(FIXSHARED)) を指定してチャンネルを再始動することができます。属性 = A (CHLDISP(SHARED)) を指定してチャンネルを開始することもできます。
- STARTING 状態または RETRY 状態のチャンネルは、同じチャンネル・イニシエーターまたは別のチャンネル・イニシエーターで属性 = S (CHLDISP(FIXSHARED)) を指定して再始動することができます。属性 = A (CHLDISP(SHARED)) を指定してチャンネルを開始することもできます。
- チャンネルが非アクティブ状態になるとき、チャンネルがトリガーを開始することは適切です。トリガーを開始する共有チャンネルは、常に、共有属性 (CHLDISP(SHARED)) を持っています。
- チャンネルが非アクティブ状態になるとき、チャンネル・イニシエーターで CHLDISP(FIXSHARED) を指定してチャンネルを開始することは適切です。属性 = A (CHLDISP(SHARED)) を指定してチャンネルを開始することもできます。
- チャンネルが開始されたチャンネル・イニシエーターが SHARED(RESTART) で停止されたとき、またはチャンネル・イニシエーターが異常終了したときに、チャンネルがキュー共有グループ内の他のアクティブなチャンネル・イニシエーターによってリカバリーされない。チャンネルがリカバリーされるのは、チャンネルが開始されたチャンネル・イニシエーターが次に再始動されたときだけです。これにより、キュー共有グループ内の他のチャンネル・イニシエーターには、失敗したチャンネルをリカバリーしようとすることによって生じる作業負荷がかかりません。

### チャンネルのテスト

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネルをテストできます。

MQSC コマンドを使用してチャンネルをテストするには、PING CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	5 (実行)
オブジェクト・タイプ	SENDER、SERVER、または CHANNEL
名称	CHANNEL.TO.USE
後処理	チャンネル・オブジェクトの属性指定

「Perform a Channel Function (チャンネル機能の実行)」パネルが表示されます。パネルの後のテキストでは、パネルの使用方法を説明します。

```
Perform a Channel Function
Select function type, complete fields, then press Enter.

機能タイプ . . . . . 1. リセット 3. コミットによる解決
2. ping 4. バックアウトで解決

チャンネル名 . . . . . : CHANNEL.TO.USE
チャンネル・タイプ . . . . . : 送信者
説明 . . . . . : CHANNEL.TO.USE の説明

後処理 . . . . . P   P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager

Sequence number for reset . . . 1      1 - 999999999
Data length for ping . . . . 16     16 - 32768

コマンド ==> _____
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel
```

図 112. チャンネルのテスト

機能タイプ 2 (Ping) を選択します。

テストの対象のチャンネルの属性指定と、どのキュー・マネージャーでそれをテストするかを選択します。データ長は最初は 16 に設定されます。必要があればその長さを変更してから Enter キーを押します。

### チャンネルのメッセージ・シーケンス番号のリセット

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネルのメッセージ・シーケンス番号をリセットできます。

MQSC コマンドを使用してチャンネル・シーケンス番号をリセットするには、RESET CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	5 (実行)

フィールド	値
オブジェクト・タイプ	チャンネル・タイプ (SENDER など) または CHANNEL
名前	CHANNEL.TO.USE
後処理	チャンネル・オブジェクトの属性指定

「チャンネル機能の実行」パネルが表示されます(743 ページの図 112 を参照)。

機能タイプ 2 (リセット) を選択します。

リセットの対象のチャンネルの属性指定と、どのキュー・マネージャーでそれをリセットするかを選択します。

「sequence number (シーケンス番号)」フィールドは初めは 1 に設定されています。必要があればその値を変更してから Enter キーを押します。

### チャンネルの未確定メッセージの解決

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネルの未確定メッセージを解決できます。

MQSC コマンドを使用するチャンネルの未確定メッセージを解決するには、RESOLVE CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	5 (実行)
オブジェクト・タイプ	SENDER、SERVER、または CHANNEL
名称	CHANNEL.TO.USE
後処理	オブジェクトの属性指定

「チャンネル機能の実行」パネルが表示されます(743 ページの図 112 を参照)。

「Function type (機能タイプ)」の「3」または「4」(コミットまたはバックアウトで解決)を選択します。(詳しくは、180 ページの『未確定チャンネル』を参照してください。)

解決対象のチャンネルの属性指定と、どのキュー・マネージャーでそれを解決するかを選択します。Enter キーを押します。

### チャンネルの停止

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネルを停止できます。

MQSC コマンドを使用してチャンネルを停止するには、STOP CHANNEL を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	7 (停止)
オブジェクト・タイプ	チャンネル・タイプ (SENDER など) または CHANNEL
名称	CHANNEL.TO.USE
後処理	オブジェクトの属性指定

「Stop a Channel (チャンネルの停止)」パネルが表示されます。パネルの後のテキストでは、パネルの使用方を説明します。

```
Stop a Channel

Complete fields, then press Enter to stop channel.

チャンネル名 . . . . . : CHANNEL.TO.USE
チャンネル・タイプ . . . . . : 送信者
説明 . . . . . : CHANNEL.TO.USE の説明

後処理 . . . . . P    P=Private on MQ25
A=Shared on any queue manager

停止モード . . . . . 1 個です。 静止 2。 強制
停止状況 . . . . . 1 個です。 2を停止しました。 非アクティブ

キュー・マネージャー . . . . . -----
接続名 . . . . . -----

コマンド ==> -----
F1=Help  F2=Split  F3=Exit  F9=SwapNext F10=Messages F12=Cancel
```

### 図 113. チャンネルの停止

停止の対象のチャンネルの属性指定と、どのキュー・マネージャーでそれを停止するかを選択します。  
必要な停止モードを選択します。

#### 静止

バッチ・サイズの値がまだ上限に達しておらず、伝送キューでは既にメッセージが待ち状態になっていても、現行メッセージが完了し、次にバッチが終了すると、チャンネルが停止します。新しいバッチは開始されません。このモードはデフォルトです。

#### 強制

チャンネルは即時に停止します。メッセージのバッチ化が進行中である場合は、「未確定」状態の結果が生じることがあります。

停止するチャンネルのキュー・マネージャーと接続名を選択します。

必要な状況を選択します。

#### 停止

チャンネルは自動的に再開しないので、手動で再開する必要があります。このモードは、キュー・マネージャーまたは接続名を指定しない場合のデフォルトです。名前を指定する場合、このモードは許可されません。

#### 非アクティブ

チャンネルは必要なときに自動的に再開されます。このモードは、キュー・マネージャーまたは接続名を指定する場合のデフォルトです。

Enter キーを押してチャンネルを停止させます。

詳しくは、『178 ページの『チャンネルの停止および静止』』を参照してください。停止したチャンネルの再始動に関する情報については、『180 ページの『停止したチャンネルの再始動』』を参照してください。

注：共有チャンネルが再試行状態になっていて、開始元のチャンネル・イニシエーターが実行されていない場合は、コマンドの入力元のキュー・マネージャーでそのチャンネルの停止要求を出します。

## チャンネル状況の表示

MQSC コマンドを使用して、または操作および制御パネルを使用して、チャンネル状況を表示できます。

MQSC コマンドを使用してチャネル状況または一連のチャネルを表示するには、DISPLAY CHSTATUS を使用します。

注: 多数のチャネルがある場合は、チャネル状況情報の表示にはある程度時間がかかります。

「リスト・チャネル」パネル(735 ページの図 106 を参照)で操作および制御パネルを使用すると、チャネル状況の要約がチャネルごとに次のように表示されます:

INACTIVE	アクティブの接続がありません。
状況	アクティブの接続が 1 つあります。
nnn status	複数の現行接続があり、すべてが同じ状況です。
nnn CURRENT	複数の現行接続がありますが、すべてが同じ状況であるわけではありません。
ブランク	IBM MQ は、アクティブの接続数を判断することができません (チャネル・イニシエーターが実行中でないなどの理由)。

注: GROUP の属性指定を指定されたチャネル・オブジェクトには、状況は示されません。

ここで、nnn はアクティブの接続数を表し、status は次のいずれかです。

INIT	INITIALIZING
BIND	BINDING
START	STARTING
RUN	実行中
STOP	STOPPING または STOPPED
RETRY	RETRYING
REQST	REQUESTING

チャネル状況に関する詳細情報を表示するには、「リスト・チャネル」または「表示」、または「チャネルの変更」パネルで「状況」キー(F11)を押して、「リスト・チャネル-現在の状況」パネルを表示します(746 ページの図 114 を参照)。

```
List Channels - Current Status - MQ25      Row 1 of 16
アクション・コードを入力してから Enter を押します。 F11 を押して、保管状況を表示します
1=Display current status

Channel name      Connection name      State
Start time      Messages Last message time Type Disposition
<> *
CHANNEL ALL MQ25

- RMA0.CIRCUIT.ACL.F RMA1
- 2005-03-21 10.22.36 557735 2005-03-24 09.51.11 SENDER PRIVATE MQ25
- RMA0.CIRCUIT.ACL.N RMA1
- 2005-03-21 10.23.09 378675 2005-03-24 09.51.10 SENDER PRIVATE MQ25
- RMA0.CIRCUIT.CL.F RMA2
- 2005-03-24 01.12.51 45544 2005-03-24 09.51.08 SENDER PRIVATE MQ25
- RMA0.CIRCUIT.CL.N RMA2
- 2005-03-24 01.13.55 45560 2005-03-24 09.51.11 SENDER PRIVATE MQ25
- RMA1.CIRCUIT.CL.F RMA1
- 2005-03-21 10.24.12 360757 2005-03-24 09.51.11 RECEIVER PRIVATE MQ25
- RMA1.CIRCUIT.CL.N RMA1
- 2005-03-21 10.23.40 302870 2005-03-24 09.51.09 RECEIVER PRIVATE MQ25
***** End of list *****
コマンド ==>
F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd
F8=Fwd F9=SwapNext F10=Messages F11=Saved F12=Cancel
```

図 114. チャネル接続のリスト

状況の値は次のとおりです。

INIT	INITIALIZING
BIND	BINDING
START	STARTING
RUN	実行中
STOP	STOPPING または STOPPED
RETRY	RETRYING
REQST	REQUESTING
DOUBT	STOPPED および INDOUBT(YES)

詳しくは、『171 ページの『チャンネルの状態』』を参照してください。

F11 キーを押して、同様のチャンネル接続リストと保管された状況を表示することができます。F11 キーを再度押すと、現行リストに戻ります。保存された状況は、メッセージの少なくとも 1 つのバッチがチャンネルに伝送されるまでは適用されません。

アクション・コード「1」またはスラッシュ (/) を使用して接続を選択し、Enter キーを押します。「Display Channel Connection Current Status (チャンネル接続の現在の状況の表示)」パネルが表示されます。

### クラスタ・チャンネルの表示

MQSC コマンドを使用して、または操作および制御パネルを使用して、クラスタ・チャンネルを表示できます。

明示的に定義されているかまたは自動定義を使用して定義されているすべてのクラスタ・チャンネルを表示するには、MQSC コマンド DISPLAY CLUSQMGR を使用します。

初期パネルから開始し、操作および制御パネルを使用して、以下のフィールドを完成させたあと、Enter キーを押します。

フィールド	値
アクション	1 (リストまたは表示)
オブジェクト・タイプ	CLUSCHL
名称	*

図 748 ページの図 115 のようなパネルが表示されます。このパネルには、各クラスタ・チャンネルの情報が三行を占め、チャンネル、クラスタ、およびキュー・マネージャー名が含まれています。クラスタ送信側チャンネルについては、全体的な状態が表示されます。

List Cluster queue manager Channels - MQ25 Row 1 of 9

アクション・コードを入力してから Enter を押します。 F11 を押して接続状況を表示します。  
1=Display 5=Perform 6=Start 7=Stop

```
Channel name      Connection name      State
Type      Cluster name      Suspended
Cluster queue manager name      Disposition
<> *      -      MQ25
- TO.MQ90.T      HURSLEY.MACH90.COM(1590)
- CLUSRCVR      VJH01T      N
- MQ90      -      MQ25
- TO.MQ95.T      HURSLEY.MACH95.COM(1595)      RUN
- CLUSSDRA      VJH01T      N
- MQ95      -      MQ25
- TO.MQ96.T      HURSLEY.MACH96.COM(1596)      RUN
- CLUSSDRB      VJH01T      N
- MQ96      -      MQ25
***** End of list *****
```

コマンド ==> \_\_\_\_\_  
F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd  
F8=Fwd F9=SwapNext F10=Messages F11=Status F12=Cancel

図 115. クラスター・チャネルのリスト

1 つまたは複数のチャネルに関する完全な情報を表示するには、チャネル名の前にアクション・コード「1」を入力して、Enter キーを押します。機能 (ping、解決、リセットなど) の実行、クラスター・チャネルの開始または停止を行うには、アクション・コード「5」、「6」、または「7」を使用します。

チャネル状況の詳細を表示するには、状況キー (F11) を押します。

## zEnterprise Data Compression Express 機能を使用するための IBM MQ for z/OS の準備

zEnterprise® Data Compression (zEDC) Express 機能は、最小 z/OS レベル z/OS 2.1 を使用して、IBM zEC12 GA2 以降の IBM Z® マシンの特定のモデルで使用可能です。

詳しくは、「[zEnterprise Data Compression \(zEDC\)](#)」を参照してください。

### 前提条件

IBM z15 以降では、zEnterprise Data Compression (zEDC) Express 機能は、Integrated Accelerator for zEDC としてオンチップとなる、ハードウェア・システムの PCIe I/O ドロワーのオプション・フィーチャーから移動されました。この変更により、構成の前提条件が更新され、ご使用のハードウェア・システムに依存するようになります。

#### IBM z15 以降

ご使用の z/OS のレベルに応じて、以下のいずれかの PTF を適用します。

- z/OS 2.5: UJ00639
- z/OS 2.4: UJ00636
- z/OS 2.3: UJ00635
- z/OS 2.2: UJ00638

z15 以降のシステムでは、ハードウェア要件はありません。これらのシステムにおける Integrated Accelerator for zEDC ソリューションは組み込みデータ・アクセラレーションを提供するため、別のアダプターは必要ありません。

#### IBM zEC12 GA2 から IBM z14

ご使用のシステムに以下の要件が必要です。

- zEDC Express<sup>®</sup> アダプター。ハードウェア・システムの PCIe I/O ドロワーに取り付けられます。
- zEDC ソフトウェア機能 (オプションの有料機能) は、IFAPRDxx parmlib メンバーで使用可能にする必要があります。

## 手順

### IBM zEC12 GA2 から IBM z14

チャンネル・イニシエーターのユーザー ID に、RACF FACILITY CLASS の FPZ.ACCELERATOR.COMPRESSION プロファイルに対する READ 権限か、または企業が使用している外部セキュリティ・マネージャー (ESM) におけるそれと同等のものに対する READ 権限が付与されていることを確認します。



**重要** : IBM z15 以降では必要ありません。

### IBM zEnterprise zEC12 GA2 以降

送信側および受信側の両方で、COMPMSG(ZLIBFAST) を指定してチャンネルを構成します。構成が済むと、zlib 圧縮を使用して、チャンネル全体を流れるメッセージの圧縮および圧縮解除が行われます。

圧縮するデータのサイズが最小しきい値を超えると、zEDC で圧縮が実行されます。しきい値は、使用されている IBM z ハードウェアによって異なります。

- IBM zEC12 GA2 から IBM z14 への最小しきい値 4KB
- IBM z15 以降の最小しきい値は 1KB

しきい値サイズを下回るメッセージの場合、圧縮またはインフレーションはソフトウェアで実行されません。

## z/OS における通信のセットアップ

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。このセクションでは、接続を定義する方法を説明します。

DQM は IBM MQ 用のリモート・キューイング機能です。これはキュー・マネージャーにチャンネル制御プログラムを提供し、このプログラムによって、通信リンクに対するインターフェースが形成されます。これらのリンクは、システム・オペレーターによって制御可能です。分散キューイング管理によって維持されるチャンネル定義は、これらの接続を使用します。

z/OS で使用できる次の 2 つの形式の通信プロトコルから、いずれかを選択します。

- 750 ページの『[z/OS での TCP 接続の定義](#)』
- 752 ページの『[APPC/MVS を使用した z/OS 用の LU6.2 接続の定義](#)』

どのチャンネル定義も伝送プロトコル (トランスポート・タイプ) 属性を 1 つだけ指定する必要があります。1 つのキュー・マネージャーで、複数のプロトコルを通信のために使用できます。

構成例- [IBM MQ for z/OS](#) を参照すると役立つ場合もあります。キュー共有グループを使用している場合は、758 ページの『[キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ](#)』を参照してください。

### 関連概念

734 ページの『[パネルおよびコマンドの使用法](#)』

MQSC コマンド、PCF コマンド、または操作および制御パネルを使用して DQM を管理できます。

728 ページの『[他のキュー・マネージャーとの通信のセットアップ](#)』

このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

661 ページの『[IBM MQ for z/OS のセットアップ](#)』

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

#### 732 ページの『z/OS におけるチャネルのモニターと制御』

リモート・キュー・マネージャーにつながるチャネルの作成、モニターおよび制御を行うためには、DQM コマンドおよびパネルを使用します。

#### 753 ページの『キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備』

このセクションの説明を使用して、IBM MQ for z/OS で、キュー共有グループを使用した分散キューイングを構成します。

#### 758 ページの『キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ』

分散キューイング管理チャネルは、開始時に、チャネル定義で指定された接続の使用を試みます。この試行を正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

### 関連タスク

#### 140 ページの『分散キューイングの構成』

このセクションでは、キュー定義、チャネル定義、トリガー操作および同期点プロシージャーを含む、IBM MQ インストール間の相互通信の詳細について説明します。

## z/OS での TCP 接続の定義

TCP 接続を定義するために、構成するいくつかの設定があります。

TCP アドレス・スペースの名前は、TCP システム・パラメーター・データ・セット `tcPIP.TCPIP.DATA` 内に指定する必要があります。データ・セットには、"`TCPIPJOBNAME TCPIP_proc`" ステートメントを組み込む必要があります。

ファイアウォールを使用している場合、チャネル・イニシエーターからチャネルのアドレス、およびリモート接続からキュー・マネージャーへの `allow` 接続を構成する必要があります。

通常、ファイアウォールの定義により、送信側の IP アドレスとポートから宛先の IP アドレスとポートが構成されます。

- z/OS イメージには複数のホスト名を含めることができるので、ファイアウォールのソース・アドレスとして複数のホストのアドレスを構成する必要が生じることもあります。

NETSTAT HOME コマンドを使用して、これらの名前とアドレスを表示できます。

- チャネル・イニシエーターでは異なるポートで複数のリスナーを指定できるため、それらのポートを構成する必要があります。
- キュー共有グループに対して共用ポートを使用する場合、共用ポートの構成も必要です。

チャネル・イニシエーターのアドレス・スペースには、データ・セットに対する読み取り権限が必要です。使用している TCP/IP 製品とインターフェースに応じて、次の方法で `TCPIP.DATA` にアクセスすることができます。

- 環境変数 `RESOLVER_CONFIG`
- HFS ファイル `/etc/resolv.conf`
- `//SYSTCPD DD` ステートメント
- `//SYSTCPDD DD` ステートメント
- `jobname/userid.TCPIP.DATA`
- `SYS1.TCPPARMS(TCPDATA)`
- `zapname.TCPIP.DATA`

また、TCP/IP に正しい高位修飾子を指定する場合にも注意が必要です。

ドメイン・ネーム・システム (DNS) サーバーを適切に構成し、名前から IP アドレスへの変換と IP アドレスから名前への変換の両方を実行できるようにする必要があります。

**注:** リゾルバーの構成に対する変更のいくつかは、例えば IBM MQ など、それを使用するアプリケーションのリサイクルを必要とします。

詳細については、次の資料を参照してください。

- [基本 TCP/IP システム](#)
- [z/OS UNIX System Services](#)。

TCP チャンネルはいずれも開始されると、TCP リソースを使用します。PROFILE.TCPIP 構成データ・セットの次のパラメータを調整することが必要な場合もあります。

#### **ACBPOOLSIZE**

開始された TCP チャンネルごとに 1 を加え、さらに 1 を加える。

#### **CCBPOOLSIZE**

開始した TCP チャンネルごとに 1、DQM ディスパッチャーごとに 1 を加え、さらに 1 を加える。

#### **DATABUFFERPOOLSIZE**

開始した TCP チャンネルごとに 2 を加え、さらに 1 を加える。

#### **MAXFILEPROC**

チャンネル・イニシエーター内のそれぞれのディスパッチャーが処理できるチャンネルの数を制御する。

このパラメータは SYSL.PARMLIB の BPXPRMxx メンバーで指定します。必ず要件を満たす値を指定してください。

チャンネル・イニシエーターは、デフォルトでは、TCPNAME キュー・マネージャー属性に指定されているスタックに関連する IP アドレスへのバインドのみを実行できます。チャンネル・イニシエーターがシステム上の追加 TCP/IP スタックを使用して通信するようにするには、TCPSTACK キュー・マネージャー属性を MULTIPLE に変更する必要があります。

#### **関連概念**

##### [751 ページの『送信側』](#)

TCP/IP 接続の送信側には、構成するいくつかの設定があります。

##### [751 ページの『TCP での受信』](#)

TCP/IP 接続の受信側には、構成するいくつかの設定があります。

##### [752 ページの『TCP リスナー・バックログ・オプションの使用』](#)

TCP/IP で受信するときに、未解決の接続要求の最大数を設定します。これらの未解決の要求は、リスナーが要求を受け入れるのを TCP/IP ポート上で待っている要求のバックログと考えることができます。

#### **z/OS 送信側**

TCP/IP 接続の送信側には、構成するいくつかの設定があります。

チャンネル定義内の接続名 (CONNAME) フィールドには、ホスト名 (例: MVSHUR1) か、またはターゲットの TCP ネットワーク・アドレスを設定する必要があります。TCP ネットワーク・アドレスは、IPv4 小数点付き 10 進数形式 (例えば、127.0.0.1) または IPv6 16 進数形式 (例えば、2001:DB8:0:0:0:0:0) になります。接続名がホスト名である場合は、ホスト名を TCP ホスト・アドレスに変換するために TCP ネーム・サーバーが必要です。(この要件は、IBM MQ の機能でなく、TCP の機能です。)

接続の開始側 (送信側、要求側、およびサーバーのチャンネル・タイプ) では、例えば、次のようにオプションの接続用ポート番号を指定できます。

#### **接続名**

192.0.2.0(1555)

この場合、接続の開始側では、ポート 1555 で listen 中の受信プログラムに接続しようとしています。

**注:** オプションのポート番号が指定されない場合、デフォルトのポート番号である 1414 が使用されます。

チャンネル・イニシエーターはアクティブかつ使用可能な任意の TCP/IP スタックを使用できます。チャンネル・イニシエーターは、デフォルトでは、アウトバウンド・チャンネルを TCPNAME キュー・マネージャー属性に指定されている TCP/IP スタックのデフォルト IP アドレスにバインドします。異なるスタックを使用して接続するには、チャンネルの LOCLADDR 属性に、スタックのホスト名または IP アドレスを指定する必要があります。

#### **z/OS TCP での受信**

TCP/IP 接続の受信側には、構成するいくつかの設定があります。

受信側のチャンネル・プログラムは、送信側チャンネルからの開始要求に応じて開始されます。そのためには、リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャンネルを開始する必要があります。このリスナー・プログラムは **START LISTENER** コマンドか操作および制御パネルを使用して開始してください。

デフォルトでは、次のようになります。

- TCP リスナー・プログラムはポート 1414 を使用し、TCP スタックで使用できるすべてのアドレスで listen します。
- TCP/IP リスナーは、TCPNAME キュー・マネージャー属性に指定されている TCP/IP スタックに関連する IP アドレスへのバインドのみを実行できます。

他のアドレス、または使用可能なすべての TCP スタックのリスナーを開始するには、TCPSTACK キュー・マネージャー属性を「MULTIPLE」に設定します。

START LISTENER コマンドで IPADDR を指定すれば、特定のアドレスまたはホスト名でのみ listen するように TCP リスナー・プログラムを開始することができます。詳しくは、[Listeners](#) を参照してください。

### **z/OS** TCP リスナー・バックログ・オプションの使用

TCP/IP で受信するときに、未解決の接続要求の最大数を設定します。これらの未解決の要求は、リスナーが要求を受け入れるのを TCP/IP ポート上で待っている要求のバックログと考えることができます。

z/OS でのデフォルトのリスナー・バックログ値は 10000 です。バックログがこの値に到達すると、TCP/IP 接続は拒否され、チャンネルを開始できません。

MCA チャンネルの場合は、チャンネルが RETRY 状態になり、後で接続が再試行されます。

クライアント接続の場合は、クライアントは、MQCONN から理由コード MQRC\_Q\_MGR\_NOT\_AVAILABLE を受け取り、後で接続を再試行することができます。

### **z/OS** APPC/MVS を使用した z/OS 用の LU6.2 接続の定義

LU6.2 接続を定義するために、構成するいくつかの設定があります。

## APPC/MVS のセットアップ

チャンネル・イニシエーターの各インスタンスは、次の例に示すように、APPC/MVS に対して定義するために使用する LU の名前を SYS1.PARMLIB の APPCPMxx メンバー内に入れておく必要があります。

```
LUADD ACBNAME( luname ) NOSCHED TPDATA(CSQ.APPCTP)
```

*luname* は、使用する論理装置名です。NOSCHED は必須です。TPDATA は使用されません。ASCHPMxx メンバーまたは APPC/MVS TP プロファイル・データ・セットへの追加は必要ありません。

DQM によって使用される接続を定義するために、サイド情報データ・セットを拡張します。APPC ユーティリティー・プログラム ATBSDFMU によるサイド情報データ・セットの拡張方法については、システムから提供されるサンプル CSQ4SIDE を参照してください。使用する TPNAME 値の詳細については、以下の表の情報を参照してください。

リモート・プラットフォーム	TPNAME
z/OS または MVS	リモート・キュー・マネージャーの対応するサイド情報内の TPNAME と同じ。
IBM i	IBM i システムにおいて経路指定項目の比較値と同じ。
UNIX and Linux システム	リモート・キュー・マネージャーの対応するサイド情報内の TPNAME と同じ。
Windows	Windows Run Listener コマンドで指定されたとおり、または Windows において TpSetup を使用して定義された呼び出し可能なトランザクション・プログラム。

同一のマシン上に複数のキュー・マネージャーがある場合は、チャンネル定義内の TPname をそれぞれ固有にする必要があります。

VTAM 定義の情報が必要な場合も、「[Multiplatform APPC Configuration Guide](#)」を参照してください。

キュー・マネージャーが、APPC を使用して同一または別の z/OS システム上のキュー・マネージャーと通信している環境では、通信中の LU の VTAM 定義が SECACPT(ALREADYV) を指定しているか、LU 間の接続用の RACF の APPCLU プロファイルが CONVSEC(ALREADYV) を指定していることを確認してください。

基本およびリスナー LU の両方に対して z/OS コマンド VARY ACTIVE を発行してから、インバウンドまたはアウトバウンド通信のいずれかの開始を試みるようにします。



**重要:** APPC セットアップに加えて、以下のコマンドを発行する必要があります。

```
ALTER QMGR LUNAME(luname)
```

次に、チャンネル・イニシエーターを再始動します。

詳細については、[LUNAME](#) を参照してください。

## 関連概念

753 ページの『[LU 6.2 への接続](#)』

LU 6.2 に接続するために、構成するいくつかの設定があります。

753 ページの『[LU 6.2 での受信](#)』

LU 6.2 で受信するために、構成するいくつかの設定があります。

### LU 6.2 への接続

LU 6.2 に接続するために、構成するいくつかの設定があります。

チャンネル定義内の接続名 (CONNNAME) フィールドは、APPC/MVS のサイド情報データ・セット内に指定されているシンボリック宛先名に設定する必要があります。

使用する LU 名は (上述のように APPC/MVS に定義されます) チャンネル・イニシエーター・パラメーター内にも指定する必要があります。これは、リスナーによって受信される LU と同じものに設定します。

チャンネル・イニシエーターは、"SECURITY(SAME)"APPC/MVS オプションを使用するので、アウトバウンド伝送に使用されるチャンネル・イニシエーター・アドレス・スペースのユーザー ID であり、受信側に提示されます。

### LU 6.2 での受信

LU 6.2 で受信するために、構成するいくつかの設定があります。

送信側チャンネルからの始動要求に応じて、MCA の受信が開始されます。そのためには、リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャンネルを開始する必要があります。リスナー・プログラムは APPC/MVS サーバーの 1 つです。これは START LISTENER コマンドまたは操作および制御パネルを使用して開始します。サイド情報データ・セット内に定義されているシンボリック宛先名を使用して、LU 名を指定する必要があります。このようにして識別されるローカル LU は、チャンネル・イニシエーター・パラメーターに設定された、アウトバウンド伝送に使用されるものと同じでなければなりません。

## キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備

このセクションの説明を使用して、IBM MQ for z/OS で、キュー共有グループを使用した分散キューイングを構成します。

キュー共有グループを使用した構成の例については、[構成例 - キュー共有グループを使用する IBM MQ for z/OS](#) を参照してください。キュー共有グループを使用するメッセージ・チャンネルの計画例については、[キュー共有グループを使用する z/OS におけるメッセージ・チャンネルの計画例](#) を参照してください。

キュー共有グループを使用した分散キューイングを使用可能にするために、以下のコンポーネントを作成および構成する必要があります。

- [LU 6.2 および TCP/IP リスナー](#)

- [伝送キューおよびトリガー操作](#)
- [メッセージ・チャンネル・エージェント](#)
- [同期キュー](#)

通信をセットアップするために必要なコンポーネントを作成した後、758 ページの『[キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ](#)』を参照してください。

キュー共有グループを使用しているときに、チャンネルをモニターおよび制御する方法については、732 ページの『[z/OS におけるチャンネルのモニターと制御](#)』を参照してください。

キュー共有グループの概念および利点については、以下のセクションを参照してください。

## サービス・クラス

共有キューとは、さまざまなサービス・クラスを備えた一種のローカル・キューのことです。共有キュー上のメッセージはカップリング・ファシリティ (CF) に保管されます。そのため、キュー共有グループ内のどのキュー・マネージャーからでもアクセスすることができます。共有キュー上のメッセージの長さは、100 MB を超えないものでなければなりません。

## 汎用インターフェース

キュー共有グループには、ネットワークがグループを単一のエンティティと見なすのに必要な汎用インターフェースがあります。このように見なされるには、グループ内のどのキュー・マネージャーに接続するのも使用できる単一の汎用アドレスを設けます。

キュー共有グループのすべてキュー・マネージャーは、その汎用アドレスに論理的に関連したアドレスで、インバウンド・セッション要求を listen します。詳細は、755 ページの『[キュー共有グループの LU 6.2 および TCP/IP リスナー](#)』を参照してください

## ロード・バランス・チャンネル開始機能

共有伝送キューは、キュー共有グループ内の任意のチャンネル・イニシエーターで実行されているアウトバウンド・チャンネルを利用することができます。ロード・バランス・チャンネル開始機能によってチャンネル・コマンドの宛先が決められます。所定の通信サブシステムにアクセスできる適切なチャンネル・イニシエーターが選ばれます。例えば、TRPTYPE(LU6.2) と定義されたチャンネルを、TCP/IP サブシステムにしかアクセスできないチャンネル・イニシエーターで開始することはできません。

どのチャンネル・イニシエーターが選択されるかは、チャンネル・イニシエーターのチャンネル・ロードと余力によって決まります。チャンネル・ロードとは、チャンネル・イニシエーターのパラメーターに定義されている最大許容アクティブ・チャンネル数に対する実際のアクティブ・チャンネル数のパーセントのことです。余力とは、実際のアクティブ・チャンネル数と最大許容数の差のことです。

インバウンド共有チャンネルは、755 ページの『[キュー共有グループの LU 6.2 および TCP/IP リスナー](#)』で説明されているように、総称アドレスを使用して、キュー共有グループ間でロード・バランシングすることができます。

## 共有チャンネルの回復

以下の表に、共有チャンネルでの障害の種類とそれぞれの種類別の対処法を示します。

障害の種類	障害の内容とその対処
チャンネル・イニシエーターの通信サブシステムの障害	通信サブシステムに従属しているチャンネルは、チャンネル再試行に入ってから、ロード・バランス開始コマンドによって、該当するキュー共有グループ・チャンネル・イニシエーターで再始動されます。
チャンネル・イニシエーター障害	チャンネル・イニシエーターに障害が起きましたが、それに関連したキュー・マネージャーはアクティブのままです。キュー・マネージャーは障害をモニターし、回復処理を開始します。

キュー・マネージャー障害	キュー・マネージャーに障害が起き、それに関連したチャンネル・イニシエーターに障害が起きました。キュー共有グループ中の他のキュー・マネージャーが、イベントをモニターし、ピア回復を開始します。
共用状況障害	チャンネルの状態情報は Db2 に保管されるので、チャンネルの状態の変更のときに Db2 への接続が失われると障害が起きます。実行中のチャンネルは、このリソースにアクセスできなくても実行を継続できます。Db2 へのアクセスで障害が起きますと、チャンネルは再試行に入ります。

障害を起こしたシステムに代わって共用チャンネル回復処理を行う場合、共用チャンネル状況を取得するため、回復しようとするシステム上で Db2 への接続が使用可能になっている必要があります。

## クライアント・チャンネル

個々のキュー・マネージャーにではなく汎用インターフェースに接続されているキュー共有グループ内のメッセージの高い利用性をクライアント接続チャンネルで活用することができます。詳しくは、[クライアント接続チャンネル](#)を参照してください。

### 関連概念

[661 ページの『IBM MQ for z/OS のセットアップ』](#)

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

[728 ページの『他のキュー・マネージャーとの通信のセットアップ』](#)

このセクションでは、分散キューイングの使用を開始する前に行う必要のある IBM MQ for z/OS の準備について説明します。

[757 ページの『クラスターとキュー共有グループ』](#)

1つの定義で、クラスターが共有キューを使用可能にすることができます。そうするためには、共有キューを定義するときにクラスター名を指定します。

[757 ページの『チャンネルと直列化』](#)

共有キュー対等リカバリー中に、共有キュー上のメッセージを処理するメッセージ・チャンネル・エージェントは、それらのキューへの自身のアクセスを直列化します。

### 関連タスク

[140 ページの『分散キューイングの構成』](#)

このセクションでは、キュー定義、チャンネル定義、トリガー操作および同期点プロシージャを含む、IBM MQ インストール間の相互通信の詳細について説明します。

### 関連情報

[共用キューとキュー共有グループ](#)

[グループ内キューイング](#)

## キュー共有グループの LU 6.2 および TCP/IP リスナー

グループ LU 6.2 および TCP/IP のリスナーは、汎用アドレスに論理接続されているアドレスで listen します。

LU 6.2 リスナーの場合、指定の LUGROUP は、キュー共有グループに関連した VTAM 汎用リソースにマップされます。このテクノロジーをセットアップする例については、[752 ページの『APPC/MVS を使用した z/OS 用の LU6.2 接続の定義』](#)を参照してください。

TCP/IP リスナーの場合、指定のポートは、以下のいずれかの方法で汎用アドレスに接続することができます。

- IBM Network Dispatcher などのフロントエンド・ルーターの場合、インバウンド接続要求は、ルーターからキュー共有グループのメンバーに転送されます。
- TCP/IP のシスプレックス・ディストリビューターの場合、分散 DVIPA としてセットアップされた特定アドレス上で実行および listen 中の各リスナーに一定の比率で着信要求が割り振られます。このテクノロジーのセットアップの例は、[シスプレックス・ディストリビューターの使用に](#)記載されています。

## キュー共有グループの伝送キューおよびトリガー操作

キュー共有グループから宛先へのメッセージの移動までの間、そのメッセージを保管するのに共有伝送キューが使われます。

これが共有キューであり、これには、キュー共有グループのどのキュー・マネージャーからでもアクセスすることができます。

### トリガー発行

トリガー共有キューは、トリガー条件が満たされた場合に複数のトリガー・メッセージを生成することができます。トリガー共有キューに関連付けられたキュー共有グループ内のキュー・マネージャーで定義されている各ローカル開始キューごとにメッセージが1つずつ生成されます。

分散キューイングの場合、共有伝送キュー・トリガー条件が満たされると、各チャンネル・イニシエーターがトリガー・メッセージを受け取ります。ただし、トリガーの開始を実際に処理するチャンネル・イニシエーターは1つだけで、残りのプログラムは処理に携わりません。トリガーされたチャンネルはロード・バランス開始 (753 ページの『キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備』を参照) で開始され、トリガーされてチャンネル QSG.TO.QM2 を開始します。共有伝送キューを作成するには、IBM MQ コマンド (MQSC) を次の例のように使用します。

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') +
USAGE(XMITQ) QSGDISP(SHARED) +
CFSTRUCT(APPLICATION1) INITQ(SYSTEM.CHANNEL.INITQ) +
TRIGGER TRIGDATA(QSG.TO.QM2)
```

## キュー共有グループのメッセージ・チャンネル・エージェント

チャンネル・イニシエーターでチャンネルを開始できるのは、該当する名前の付いたチャンネルのチャンネル定義にアクセスできる場合だけです。

メッセージ・チャンネル・エージェントは、メッセージの送受信を制御する IBM MQ プログラムです。メッセージ・チャンネル・エージェントは、1つのキュー・マネージャーから別のキュー・マネージャーにメッセージを移動します。チャンネルの両端に1つずつ、メッセージ・チャンネル・エージェントが存在します。

チャンネル定義は、キュー・マネージャー専用として定義することができますが、共有リポジトリに保管してどこでも使える (グループ定義) ようにすることもできます。つまり、グループ定義のチャンネルは、キュー共有グループ内のすべてのチャンネル・イニシエーターで使えるということです。

**注:** グループ定義を専用コピーしたものは、変更または削除することができます。

グループ・チャンネル定義を作成するには、IBM MQ コマンド (MQSC) を次の例のように使用します。

```
DEFINE CHL(QSG.TO.QM2) CHLTYPE(SDR) +
TRPTYPE(TCP) CONNAME(QM2.MACH.IBM.COM) +
XMITQ(QM2) QSGDISP(GROUP)
```

```
DEFINE CHL(QM2.TO.QSG) CHLTYPE(RCVR) TRPTYPE(TCP) +
QSGDISP(GROUP)
```

キュー共有グループでの分散キューイングに使用されるメッセージ・チャンネル・エージェントには、次のように2とおりの見方があります。

### インバウンド

インバウンド・チャンネルは、グループ・リスナーを通してキュー・マネージャーに接続されていれば共有チャンネルになります。このチャンネルは、汎用インターフェースを通してキュー共有グループに接続されてグループ内のキュー・マネージャーにつながるか、あるいは特定のキュー・マネージャーのグループ・ポートまたはグループ・リスナーが使用する LU 名の対象になります。

## アウトバウンド

アウトバウンド・チャンネルは、共有伝送キューからメッセージを移動する場合は共有チャンネルになります。例のコマンドでは、送信側チャンネル QSG.TO.QM2 は、その伝送キュー QM2 が QSGDISP(SHARED) を使用して定義されているため、共有チャンネルになります。

### **z/OS** キュー共有グループの同期キュー

共有チャンネルには、SYSTEM.QSG.CHANNEL.SYNCQ という名前の独自の共有同期キューがあります。

この同期キューは、キュー共有グループ内のすべてメンバーにアクセスすることができます。(専用チャンネルは専用同期キューを引き続き使用します。731 ページの『IBM MQ オブジェクトの定義』を参照してください。つまり、通信サブシステム、チャンネル・イニシエーター、またはキュー・マネージャーに障害が起きた場合、キュー共有グループ内の別のキュー・マネージャーおよびチャンネル・イニシエーターのインスタンスからチャンネルを再始動できるということです。詳しくは、753 ページの『キュー共有グループを使用する DQM のための IBM MQ for z/OS の準備』を参照してください。

キュー共有グループを使用する DQM の場合、SYSTEM.QSG.CHANNEL.SYNCQ という名前の共有キューが使用可能でなければなりません。グループ・リスナーを正常に始動できるようにするには、このキューが使用可能になっていなければなりません。

このキューが使用可能でないためにグループ・リスナーに障害が起きても、チャンネル・イニシエーターのリサイクルなしに、このキューを定義してリスナーを再始動することができます。非共有チャンネルは影響を受けません。

このキューは、必ず INDXTYPE(MSGID) を使用して定義してください。この定義により、キューのメッセージに対するアクセス速度が向上します。

### **z/OS** クラスタとキュー共有グループ

1 つの定義で、クラスタが共有キューを使用可能にすることができます。そうするためには、共有キューを定義するときにクラスタ名を指定します。

キュー共有グループ内のすべてのキュー・マネージャーを共有キューはホストとして利用しているようにネットワーク内のユーザーには見えます。(共有キューがキュー共有グループをホストとして利用していることは公示されません。) クライアントは、キュー共有グループのどのメンバーとのセッションを開始しても、同じ共有キューにメッセージを書き込むことができます。

詳しくは、236 ページの『キュー・マネージャー・クラスタの構成』を参照してください。

### **z/OS** チャンネルと直列化

共有キュー対等リカバリー中に、共有キュー上のメッセージを処理するメッセージ・チャンネル・エージェントは、それらのキューへの自身のアクセスを直列化します。

メッセージ・チャンネル・エージェントが 1 つ以上の共有キュー上のコミットされていないメッセージを扱っているときにキュー共有グループ内のキュー・マネージャーに障害が起きた場合は、チャンネルおよび関連したチャンネル・イニシエーターは終了して、キュー・マネージャーの共有キュー対等リカバリーが行われます。

共有キュー対等リカバリーは非同期アクティビティであるため、対等チャンネル・リカバリーでは、共有キュー対等リカバリーが完了する前に、同時にキュー共有グループ内の別の部分のチャンネルに対して再始動の試行が行われる可能性があります。このイベントが起きた場合は、まだリカバリー中のメッセージより先に、コミット済みメッセージが処理される可能性があります。メッセージがこのように違う順序で処理されないようにするために、共有キュー上のメッセージを処理するメッセージ・チャンネル・エージェントは、それらのキューへの自身のアクセスを直列化します。

共有キュー対等リカバリーがまだ進行中のチャンネルを開始しようとすると、失敗になります。リカバリーが進行中であることを示すエラー・メッセージが出されて、チャンネルは再試行状態になります。キュー・マネージャー対等リカバリーが完了すると、チャンネルは次の再試行時に再始動されます。

チャンネルの解決 (RESOLVE)、PING、削除 (DELETE) を行おうとした場合も、同じ理由で失敗になります。

## **z/OS** キュー共有グループを使用する IBM MQ for z/OS の通信のセットアップ

分散キューイング管理チャンネルは、開始時に、チャンネル定義で指定された接続の使用を試みます。この試行を正常に行うためには、その接続が定義されていて、使用可能になっていなければなりません。

使用できる次の2つの形式の通信プロトコルから、いずれかを選択します。

- TCP
- LU 6.2 から APPC/MVS

構成例 - キュー共有グループを使用する IBM MQ for z/OS を参照すると役に立つ場合があります。

## **z/OS** キュー共有グループ用の TCP 接続の定義

キュー共有グループ用に TCP 接続を定義するには、送信側および受信側の特定の属性を構成する必要があります。

TCP のセットアップについては、750 ページの『[z/OS での TCP 接続の定義](#)』を参照してください。

## 送信側

キュー共有グループに接続するためのチャンネル定義内の接続名(CONNAME)フィールドは、キュー共有グループの汎用インターフェースに設定されている必要があります([キュー共有グループ](#)を参照)。詳細については、[シスプレックス・ディストリビューターの使用](#)を参照してください。

## キュー共有グループを使用した TCP での受信

受信側の共有チャンネル・プログラムは、送信側チャンネルからの始動要求に応じて開始されます。そのためには、リスナーを開始して着信ネットワーク要求を検出し、関連のチャンネルを開始する必要があります。このリスナー・プログラムの開始は START LISTENER コマンド、グループの到着属性指定、または操作および制御パネルを使用して行います。

キュー共有グループ内のすべてのグループ・リスナーが同一ポートで listen していなければなりません。単一の MVS イメージ上で複数のチャンネル・イニシエーターを実行する場合は、START LISTENER コマンドで IPADDR を指定すれば、仮想 IP アドレスを定義することができ、特定のアドレスまたはホスト名でのみ listen するように TCP リスナー・プログラムを開始することができます。(詳細については、[START LISTENER](#) を参照してください。)

## **z/OS** z/OS での LU 6.2 接続の定義

キュー共有グループ用に LU 6.2 接続を定義するには、送信側および受信側の特定の属性を構成する必要があります。

APPC/MVS のセットアップについては、[z/OS における通信のセットアップ](#)を参照してください。

## APPC/MVS (LU 6.2) への接続

キュー共有グループに接続するためのチャンネル定義内の接続名 (CONNAME) フィールドは、APPC/MVS のサイド情報データ・セットで指定されているシンボリック宛先名に設定する必要があります。このシンボリック宛先で定義されているパートナー LU は、汎用リソース名でなければなりません。詳細については、[汎用リソースの使用によるネットワークへの自身の登録](#)を参照してください。

## 汎用インターフェースを使用した LU 6.2 での受信

受信側の共有 MCA は、送信側チャンネルからの始動要求に応じて開始されます。そのためには、グループ・リスナー・プログラムを開始して着信ネットワーク要求を検出し、関連のチャンネルを開始する必要があります。リスナー・プログラムは APPC/MVS サーバーの1つです。リスナー・プログラムの開始は START LISTENER コマンド、到着属性指定グループ、または操作および制御パネルを使用して行います。サイド情報データ・セット内に定義されているシンボリック宛先名を使用して、LU 名を指定する必要があります。詳細については、[汎用リソースの使用によるネットワークへの自身の登録](#)を参照してください。

## z/OS IMS での IBM MQ の使用

IBM MQ-IMS アダプターおよび IBM MQ-IMS ブリッジは、IBM MQ が IMS と相互作用できるようにする 2 つのコンポーネントです。

IBM MQ と IMS が一緒に動作するよう構成するには、以下のタスクを完了する必要があります。

- [759 ページの『IMS アダプターのセットアップ』](#)
- [766 ページの『IMS ブリッジのセットアップ』](#)

### 関連概念

[767 ページの『CICS での IBM MQ の使用』](#)

CICS で IBM MQ を使用するには、IBM MQ CICS アダプターおよびオプションで IBM MQ CICS bridge コンポーネントを構成する必要があります。

[770 ページの『IMS における OTMA 出口の使用』](#)

このトピックは、IBM MQ for z/OS で IMS Open Transaction Manager Access 出口を使用する場合に使用します。

### 関連タスク

[657 ページの『z/OS でのキュー・マネージャーの構成』](#)

IBM MQ for z/OS でキュー・マネージャーを構成するには、この指示に従ってください。

### 関連資料

[768 ページの『言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用』](#)

必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

### 関連情報

[IBM MQ および IMS クライアント](#)

[IBM MQ for z/OS での IMS および IMS ブリッジ・アプリケーション](#)

## z/OS IMS アダプターのセットアップ

IMS 内で IBM MQ を使用するには、IBM MQ-IMS アダプター (通常、IMS アダプターと呼ばれます) が必要です。

このトピックでは、IMS アダプターを、どのようにして IMS サブシステムで使用可能にするかを説明します。IMS サブシステムの調整に慣れていない場合は、*IBM Documentation* の IMS 情報を参照してください。

IMS アダプターを IMS アプリケーションで使用できるようにするには、次のステップを実行してください。

1. IMS 外部サブシステム接続機能 (ESAF) を使用して、IBM MQ を IMS へ外部サブシステムとして定義します。

[761 ページの『IMS への IBM MQ の定義』](#) を参照してください。

2. JOBLIB 連結または STEPLIB 連結内の IBM MQ ロード・ライブラリー `thlqual.SCSQAUTH` を、使用している IMS 制御領域用および IBM MQ に接続した従属領域用の JCL に組み込みます (LPA またはリンク・リストに含まれていない場合)。また、使用している JOBLIB または STEPLIB が許可済みでない場合は、それを DFSESL 連結内の、IMS モジュールが入っているライブラリー (通常は IMS RESLIB) のあとに組み込みます。

また、`thlqual.SCSQANLx` ( $x$  は言語を示す文字) も組み込んでください。

DFSESL が存在する場合、`SCSQAUTH` および `SCSQANLx` を連結に含めるか、`LNKLIST` に追加する必要があります。JCL の STEPLIB または JOBLIB 連結に追加するだけでは十分ではありません。

3. IBM MQ アセンブラー・プログラム `CSQQDEFV` を、`thlqual.SCSQASMS` からユーザー・ライブラリーへコピーします。

4. 提供されるプログラム CSQQDEFV は、IMS 言語インターフェース・トークン (LIT) である MQM1 と共に、デフォルト値であるサブシステム名 CSQ1 を含んでいます。テストおよびインストール検査では、この名前をそのまま使用できます。

実動サブシステムでは、NAME=CSQ1 を独自のサブシステム名に変更するか、CSQ1 をそのまま使用します。必要に応じて、さらにサブシステム定義を追加することもできます。LITs の詳細については、764 ページの『[IMS アダプターへの IBM MQ キュー・マネージャーの定義](#)』を参照してください。
5. プログラムをアセンブルし、リンク・エディットして、CSQQDEFV ロード・モジュールを生成します。アセンブリーではライブラリー thlqual.SCSQMACS を SYSLIB 連結に組み込みます (リンク・エディット・パラメーター RENT を使用してください)。これは thlqual.SCSQPROC(CSQ4DEFV) のサンプル JCL に示されています。
6. JOBLIB 連結または STEPLIB 連結内で作成したモジュール CSQQDEFV が入っているユーザー・ライブラリーを、IBM MQ に接続した従属領域用の JCL に組み込みます。SCSQAUTH にデフォルトのロード・モジュールがあるので、このライブラリーを SCSQAUTH の前に入れてください。これを行わないと、IMS からユーザー 3041 異常終了コードを受け取ります。
7. IMS アダプターは、予期しない IBM MQ エラーを検出すると、DD 名 CSQSNAP に対して、z/OSSNAP ダンプを発行し、理由コード MQRC\_UNEXPECTED\_ERROR をアプリケーションに発行します。CSQSNAP DD ステートメントが IMS 従属領域 JCL に含まれていなかった場合、ダンプはとられません。この場合、JCL に CSQSNAP DD ステートメントを含めて、アプリケーションを再実行することができます。しかし、いくつかの問題が断続的に発生することがあるため、CSQSNAP DD ステートメントを含めて、問題の発生時に障害の理由を見極めることをお勧めします。
8. 動的 IBM MQ 呼び出し ([IBM MQ スタブの動的呼び出しでの説明](#)) を使用する場合は、761 ページの [図 116](#) に示されているように、動的スタブをビルドします。
9. IMS トリガー・モニターを使用する場合は、IMS トリガー・モニター・アプリケーション CSQQTRMN を定義し、PSBGEN と ACBGEN を実行します。765 ページの『[IMS トリガー・モニターのセットアップ](#)』を参照してください。
10. OPERCMDS クラスのリソースを保護するために RACF を使用している場合は、IBM MQ キュー・マネージャーのアドレス・スペースに関連したユーザー ID が、接続されている可能性のある IMS システムに対して、MODIFY コマンドを発行する権限を持っていることを確認してください。

```

//DYNSTUB EXEC PGM=IEWL,PARM='RENT,REUS,MAP,XREF'
//SYSPRINT DD SYSOUT=*
//ACSQMOD DD DISP=SHR,DSN=thlqual.SCSQLOAD
//IMSLIB DD DISP=SHR,DSN=ims.reslib
//SYSLMOD DD DISP=SHR,DSN=private.load1
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSLIN DD *
INCLUDE ACSQMOD(CSQSTUB)
INCLUDE IMSLIB(DFSLI000)
ALIAS MQCONN,MQCONN,MQDISC MQI entry points
ALIAS MQGET,MQPUT,MQPUT1 MQI entry points
ALIAS MQOPEN,MQCLOSE MQI entry points
ALIAS MQBACK,MQCMIT MQI entry points
ALIAS CSQBBAK,CSQBCMT MQI entry points
ALIAS MQINQ,MQSET MQI entry points
ALIAS DFSPLI,PLITDLI IMS entry points
ALIAS DFSCOBOL,CBLTDLI IMS entry points
ALIAS DFSFOR,FORTDLI IMS entry points
ALIAS DFSASM,ASMTDLI IMS entry points
ALIAS DFSPASCL,PASTDLI IMS entry points
ALIAS DFHEI01,DFHEI1 IMS entry points
ALIAS DFSAIBLI,AIBTDLI IMS entry points
ALIAS DFSESS,DSNWLI,DSNHLI IMS entry points
ALIAS MQCRTMH,MQDLTMH,MQDLTMP IMS entry points
ALIAS MQINQMP,MQSETMP,MQMHBUFF,MQBUFFMH IMS entry points
MODE AMODE(31),RMODE(24) Note RMODE setting
NAME CSQDYNS(R)
/*

```

<sup>1</sup>Specify the name of a library accessible to IMS applications that want to make dynamic calls to IBM MQ.

図 116. 動的呼び出しスタブをリンク・エディットする JCL のサンプル

## 関連概念

766 ページの『IMS ブリッジのセットアップ』

IBM MQ-IMS ブリッジは、IBM MQ 対応ではない既存のプログラムおよびトランザクションとの間で、IBM MQ による入出力を可能にするオプションのコンポーネントです。

## 関連情報

[IBM MQ および IMS クライアント](#)

[IBM MQ for z/OS での IMS および IMS ブリッジ・アプリケーション](#)

## z/OS IMS への IBM MQ の定義

IBM MQ は、IMS 制御領域に対して定義し、またその IBM MQ キュー・マネージャーにアクセスする各従属領域に対して定義する必要があります。このためには、IMS .PROCLIB ライブラリーにサブシステム・メンバー (SSM) を 1 つ作成し、その SSM を適用可能な IMS 領域に知らせなければなりません。

## IMS にサブシステム・メンバー項目を配置する。PROCLIB

IMS .PROCLIB 内の 1 つの SSM 項目は、IMS 領域から別のキュー・マネージャーへの 1 つの接続を定義します。

SSM の名前を付けるために、IMS IMSCTRL マクロの IMSID フィールドの値 (1 から 4 文字の英数字) とユーザーが定義した任意の名前 (1 から 4 文字の英数字) とを連結します。

1 つの SSM をすべての IMS 領域で共有するか、特定のメンバーを各領域用に定義できます。このメンバーには、外部サブシステムへの接続の数だけの項目が入っています。1 つの項目は、80 文字からなる 1 つのレコードです。

## 定位置パラメーター

この項目のフィールドは、次のとおりです。

SSN,LIT,ESMT,RTT,REO,CRC

ここで、

#### SSN

IBM MQ キュー・マネージャー名を指定します。このフィールドは必須であり、1～4文字が入っていないければなりません。

#### LIT

IMS に供給する言語インターフェース・トークン (LIT) を指定します。このフィールドは必須であり、その値は、CSQQDEFV モジュールのフィールドと一致する必要があります。

#### ESMT

外部サブシステム・モジュール表 (ESMT) を指定します。この表は、どの接続モジュールを IMS がロードしなければならないかを指定します。CSQQESMT がこのフィールドの必須の値です。

#### RTT

このオプションは、IBM MQ ではサポートされません。

#### REO

IMS アプリケーションが動作しない外部サブシステムを参照する場合、または資源がスレッド作成時に使用可能でない場合に使用される領域エラー・オプション (REO) を指定します。このフィールドの指定は任意であり、次の 1 文字が含まれます。

#### R

戻りコードをアプリケーションに渡し、IBM MQ サービスに対する要求が失敗したことを示します。

#### Q

アプリケーションを異常終了コード U3051 で終了させ、アクティビティを最後のコミット点までバックアウトし、トランザクションの PSTOP を行い、入力メッセージを再キューイングします。このオプションは、IMS アプリケーションが動作しない外部サブシステムを参照しようとするか、資源がスレッド作成時に使用可能でない場合にだけ適用されます。

IBM MQ が要求を処理している間、すなわちアダプターが要求を IBM MQ へ渡した後、IBM MQ の問題が起こった場合、IBM MQ の完了コードおよび理由コードがアプリケーションへ戻されます。

#### A

アプリケーションを異常終了コード U3047 で終了させ、入力メッセージを破棄します。このオプションは、IMS アプリケーションが動作しない外部サブシステムを参照するか、資源がスレッド作成時に使用可能でない場合にだけ適用されます。

IBM MQ が要求を処理している間、すなわちアダプターが要求を IBM MQ へ渡した後、IBM MQ の問題が起こった場合、IBM MQ の完了コードおよび理由コードがアプリケーションへ戻されます。

#### CRC

このオプションを指定できますが、IBM MQ では使用されません。

注：すべての定位置パラメーターについて詳しくは、[IMS への外部サブシステムの指定方法を参照](#)してください。

SSM 項目の例を次に示します。

CSQ1,MQM1,CSQQESMT,,R,

ここで、

<b>CSQ1</b>	IBM MQ で提供されているデフォルトのサブシステム名。この名前は、ご使用のシステムに合わせて変更できます。
<b>MQM1</b>	CSQQDEFV 内で提供されるデフォルトの LIT。
<b>CSQQESMT</b>	外部サブシステム・モジュール名。この値は必ず使用してください。
<b>R</b>	REO オプション。

### キーワード・パラメーター

IBM MQ パラメーターはキーワード形式で指定できます。SST パラメーターには、DB2 または MQ のいずれかの値を指定できます。IMS 14 で、MQ の値のサポートが追加されました。MQ を使用するとより明確になり、IMS サブシステム・コマンドに SST 値が含まれるようになりましたが、それ以外の大きな影響はありません。必要に応じて、DB2 の値を引き続き使用できます。その他のパラメーターについては[定位置パラメーター](#)に説明があります。また、以下の例も参照してください。

```
SST=MQ,SSN=SYS3,LIT=MQM3,ESMT=CSQQESMT
```

ここで、

<b>SYS3</b>	サブシステム名。
<b>MQM3</b>	CSQQDEFV に指定された LIT。
<b>CSQQESMT</b>	外部サブシステム・モジュール名。

### SSM EXEC パラメーターの指定

IMS 制御領域の始動プロシージャに SSM EXEC パラメーターを指定してください。このパラメーターには、1～4 文字のサブシステム・メンバー名 (SSM) を指定します。

IMS 制御領域について SSM を指定すると、その制御領域下で実行するどの従属領域も、SSM パラメーターで指定された IMS.PROCLIB メンバーで名前を指定した IBM MQ キュー・マネージャーに接続できます。IMS.PROCLIB メンバ名は IMSID(IMSID=xxxx)に SSM EXEC パラメータで指定された 1～4 文字を連結したものです。IMS ID は、IMSCTRL 生成マクロの IMSID パラメーターです。

IMS によって、いくつでも必要な数の外部サブシステム接続を定義できます。さまざまな IBM MQ キュー・マネージャーについて、複数の接続を定義できます。すべての IBM MQ 接続は、同じ z/OS システム内になければなりません。従属領域では、従属領域 SSM を指定するか、制御領域用に指定されたものを使用することができます。従属領域 SSM と制御領域 SSM で、異なる領域エラー・オプション (REO) を指定できます。[763 ページの表 51](#)は、使用できる各種の SSM 指定を示しています。

制御領域用 SSM	従属領域用 SSM	アクション	コメント
No	No	なし	外部サブシステムを接続することはできません。
No	Yes	なし	外部サブシステムを接続することはできません。
Yes	No	制御領域 SSM を使用します。	この領域内でスケジュールされたアプリケーションは、制御領域 SSM 内で識別される外部サブシステムにアクセスできます。各接続に対する出口および制御ブロックは、制御領域と従属領域のアドレス・スペースにロードされます。

表 51. SSM 指定オプション (続き)

制御領域用 SSM	従属領域用 SSM	アクション	コメント
Yes	あり (空)	従属領域については、SSM は使用されません。	この領域内でスケジュールされたアプリケーションは、DL/I データベースにだけアクセスできます。各接続に対する出口および制御ブロックは、制御領域のアドレス・スペースにロードされます。
Yes	あり (空ではない)	従属領域 SSM を制御領域 SSM と照合します。	この領域内でスケジュールされたアプリケーションは、両方の SSM で識別される外部サブシステムにだけアクセスできます。各接続に対する出口および制御ブロックは、制御領域と従属領域のアドレス・スペースにロードされます。

指定可能な SSM の最大数を制御する特定のパラメーターはありません。

## IMS アダプターのプリロード

IMS アダプターは、IMS によってプリロードされると、パフォーマンスが向上する場合があります。プリロードは、IMS .PROCLIB の DFSMPLxx メンバーによって制御されます。詳しくは、「IMS 管理の手引き: システム」を参照してください。指定する IBM MQ モジュール名は次のとおりです。

CSQACLST	CSQAMLST	CSQAPRH	CSQAVICM	CSQFSALM	CSQQDEFV
CSQQCONN	CSQQDISC	CSQQTERM	CSQQINIT	CSQQBACK	CSQQCMMT
CSQQESMT	CSQQPREP	CSQQTTHD	CSQQWAIT	CSQQNORM	CSQQSSOF
CSQQSSON	CSQFSTAB	CSQQRESV	CSQQSNOP	CSQQCMND	CSQQCVER
CSQQTMID	CSQQTRGI	CSQQCON2	CSQBPAPI	CSQBCRMH	CSQBAPPL

IBM MQ classes for JMS の使用について詳しくは、[IMS での IBM MQ classes for JMS の使用](#)を参照してください。

IMS の現行リリースでは、PDS-E 形式ライブラリーの IBM MQ モジュールをプリロードすることがサポートされるのは、MPP、BMP、IFP、JMP および JBP 領域のみです。その他のタイプの IMS 領域では、PDS-E ライブラリーからのプリロードがサポートされていません。その他のタイプの領域でプリロードが必要な場合は、提供されている IBM MQ モジュールを PDS 形式ライブラリーにコピーする必要があります。

## IMS アダプターへの IBM MQ キュー・マネージャーの定義

IBM MQ キュー・マネージャーの名前とそれに対応する言語インターフェース・トークン (LIT) をキュー・マネージャーの定義表に定義する必要があります。

提供された CSQQDEFX マクロを使用して、CSQQDEFV ロード・モジュールを作成する。[764 ページの図 117](#)は、アセンブラー・マクロの構文を示します。

```
CSQQDEFX TYPE=ENTRY|DEFAULT, NAME=qmgr-name, LIT=token
or
CSQQDEFX TYPE=END
```

図 117. CSQQDEFX マクロの構文

### パラメーター

#### TYPE=ENTRY|DEFAULT

TYPE=ENTRY または TYPE=DEFAULT を次のように指定します。

## TYPE=ENTRY

IMS アプリケーションで使用可能な IBM MQ キュー・マネージャーを記述する表項目を生成することを指定します。これが最初の項目である場合、CSQQDEFV CSECT ステートメントを含む表ヘッダーも生成されます。

## TYPE=DEFAULT

TYPE=ENTRY の場合と同様です。指定されるキュー・マネージャーは、MQCONN または MQCONNX でブランクのみからなる名前を指定したときに使用されるデフォルトのキュー・マネージャーになります。このような項目は、表内に 1 つのみ指定できます。

## NAME=qmgr-name

MQCONN または MQCONNX で指定したキュー・マネージャーの名前を指定します。

## LIT= token

IMS がキュー・マネージャーを識別するために使用する言語インターフェース・トークン (LIT) の名前を指定します。

MQCONN または MQCONNX 呼び出しは、*name* 入力パラメーターと *hconn* 出力パラメーターを名前ラベルに関連付けます。これにより、CSQQDEFV 項目内の LIT が関連付けられます。*hconn* パラメーターを渡す、それ以降の IBM MQ 呼び出しは、MQCONN または MQCONNX 呼び出し内で識別された CSQQDEFV 項目からの LIT を使用して、同じ LIT を持つ IMS SSM PROCLIB メンバーの中に定義される IBM MQ キュー・マネージャーに呼び出しを方向付けます。

要約すると、MQCONN または MQCONNX 呼び出しの **name** パラメーターは、CSQQDEFV 内の LIT を識別し、SSM メンバー内の同じ LIT が IBM MQ キュー・マネージャーを識別します。(MQCONN 呼び出しについては、[MQCONN-キュー・マネージャーの接続](#)を参照してください。MQCONNX 呼び出しについては詳しくは、[MQCONNX - キュー・マネージャーの接続 \(拡張\)](#)を参照してください。)

## TYPE=END

表が完了したことを指定します。このパラメーターを省略した場合は、TYPE=ENTRY が指定されたものと見なされます。

## CSQQDEFX マクロの使用

765 ページの図 118 には、キュー・マネージャー定義テーブルの一般的なレイアウトが示されます。

```
CSQQDEFX NAME=subsystem1,LIT=token1
CSQQDEFX NAME=subsystem2,LIT=token2,TYPE=DEFAULT
CSQQDEFX NAME=subsystem3,LIT=token3
...
CSQQDEFX NAME=subsystemN,LIT=tokenN
CSQQDEFX TYPE=END
END
```

図 118. キュー・マネージャー定義表のレイアウト

## IMS トリガー・モニターのセットアップ

IMS バッチ型プログラムをセットアップして、IBM MQ 開始キューをモニターすることができます。

thlqual.SCSQPROC ライブラリー内のモデル CSQQTAPL ([CSQQTRMN 用のトランザクション定義の例を参照](#))を使用して、アプリケーションを IMS へ定義してください。

thlqual.SCSQPROC ライブラリー内のモデル CSQQTPSB ([CSQQTRMN 用の PSB 定義の例を参照](#))を使用して、PSB および ACB を生成します。

```
* This is the application definition *
* for the IMS Trigger Monitor BMP *
```

```
APPLCTN PSB=CSQQTRMN,
PGMTYPE=BATCH,
SCHDTYP=PARALLEL
```

図 119. CSQQTRMN 用のトランザクション定義の例

```
PCB TYPE=TP,          ALTPCB for transaction messages
MODIFY=YES,           To "triggered" IMS transaction
PCBNAME=CSQQTRMN
PCB TYPE=TP,          ALTPCB for diagnostic messages
MODIFY=YES,           To LTERM specified or "MASTER"
PCBNAME=CSQQTRMG,
EXPRESS=YES
PSBGEN LANG=ASSEM,
PSBNAME=CSQQTRMN,    Runs program CSQQTRMN
CMPAT=YES
```

図 120. CSQQTRMN 用の PSB 定義の例

IMS トリガー・モニターの開始および停止について詳しくは、[IMS トリガー・モニターの制御](#)を参照してください。

## IMSブリッジのセットアップ

IBM MQ-IMS ブリッジは、IBM MQ 対応ではない既存のプログラムおよびトランザクションとの間で、IBM MQ による入出力を可能にするオプションのコンポーネントです。

このトピックでは、IBM MQ-IMS ブリッジのカスタマイズに必要な作業について説明します。

### IBM MQ 用に XCF と OTMA パラメーターを定義します。

このステップは、IBM MQ システム用に XCF グループとメンバー名を定義し、さらに他の OTMA パラメーターを定義します。IBM MQ と IMS は、同じ XCF グループに属している必要があります。CSQ6SYSP マクロの OTMACON キーワードを使用して、システム・パラメーター・ロード・モジュール内でこれらのパラメーターを調整します。

詳細については、[CSQ6SYSP の使用](#)を参照してください。

### XCF と OTMA パラメーターを IMS に定義します。

このステップでは、IMS システム用の XCF グループとメンバー名を定義します。IMS と IBM MQ は、同じ XCF グループに属している必要があります。

JCL または IMS PROCLIB 内のメンバー DFSPBxxx の IMS パラメーター・リストに、次のパラメーターを追加してください。

#### OTMA=Y

これにより、IMS の開始時に OTMA が自動的に開始されます。(これはオプションです。OTMA=N を指定した場合は、IMS コマンド /START OTMA を発行することによって OTMA を開始することもできます。)

#### GRNAME=

このパラメーターには XCF グループ名を指定します。

これはストレージ・クラス定義の中で指定されたグループ名と同じです(次のステップを参照)。また、CSQ6SYSP マクロの OTMACON キーワードの **Group** パラメーターで指定されたグループ名と同じです。

#### OTMANM=

このパラメーターには IMS システムの XCF メンバー名を指定します。

これはストレージ・クラス定義の中で指定されたメンバー名と同じです (次のステップを参照)。

### IBM MQ に IMS システムの XCF グループとメンバー名を指示します。

これはキューのストレージ・クラスによって指定されます。IBM MQ-IMS ブリッジを介してメッセージを送る場合、キュー用にストレージ・クラスを定義するときに、これを指定する必要があります。ストレージ・クラスには、対象の IMS システムの XCF グループとメンバー名を定義する必要があります。これを行うには、IBM MQ の操作および制御パネルを使用するか、[プログラマブル・コマンド・フォーマットの概要](#)に説明されているように IBM MQ コマンドを使用します。

### 必要なセキュリティを設定します。

/SECURE OTMA IMS コマンドは、OTMA を介して IMS に接続するすべての IBM MQ キュー・マネージャーに適用されるセキュリティのレベルを決定します。詳しくは、[IBM MQ と IMS を併用する場合のセキュリティに関する考慮事項](#)を参照してください。

## 追加の IMS 接続の同じキュー・マネージャーへの追加

IMS 接続を同じキュー・マネージャーに追加するには、以下のようにする必要があります。

- 2 番目のストレージ・クラス [STGCLASS](#) を定義して、新しい IMS を指すようにします。詳しくは、[DEFINE STGCLASS](#) を参照してください。
- 2 番目のストレージ・クラスを指す新しいローカル・キューを追加します。

### 重要：

- 1 つのローカル・キューが 2 つのストレージ・クラスを指すことはできません。
- 1 つのストレージ・クラスが 2 つの IMS ブリッジを指すことはできません。
- IBM MQ と IMS は、同じ XCF グループに属している必要があります。CSQ6SYSP マクロの OTMACON キーワードを使用して、システム・パラメーター・ロード・モジュール内でこれらのパラメーターを調整します。

詳細については、[CSQ6SYSP の使用](#)を参照してください。

### 関連概念

759 ページの『[IMS アダプターのセットアップ](#)』

IMS 内で IBM MQ を使用するには、IBM MQ-IMS アダプター (通常、IMS アダプターと呼ばれます) が必要です。

### 関連情報

[IBM MQ および IMS クライアント](#)

[IBM MQ for z/OS での IMS および IMS ブリッジ・アプリケーション](#)

## CICS での IBM MQ の使用

CICS で IBM MQ を使用するには、IBM MQ CICS アダプターおよびオプションで IBM MQ CICS bridge コンポーネントを構成する必要があります。

IBM MQ CICS アダプターおよび IBM MQ CICS bridge コンポーネントの構成について詳しくは、CICS 資料の「[MQ への接続の構成](#)」セクションを参照してください。

### 関連概念

759 ページの『[IMS での IBM MQ の使用](#)』

IBM MQ-IMS アダプターおよび IBM MQ-IMS ブリッジは、IBM MQ が IMS と相互作用できるようにする 2 つのコンポーネントです。

### 関連資料

768 ページの『[言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用](#)』

必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

### 関連情報

[IBM MQandCICS](#)

## 言語環境プログラムまたは z/OS 呼び出し可能サービスのアップグレードおよび保守の適用

必要なアクションは、CALLLIBS または LINK のどちらを使用するか、および使用する SMP/E のバージョンによって異なります。

次の表には、以下の製品のレベルをアップグレードする場合、または保守を適用する場合に、IBM MQ for z/OS に行う必要がある事項を示しています。

- Language Environment
- z/OS 呼び出し可能サービス (例えば、APPC および RRS)

製品	CALLLIBS および SMP/E V3R2 以降を使用する場合の処置	LINK を使用する場合の処置
	<p>注 : Language Environmentp、および呼び出し可能サービスのために異なるジョブを実行する必要はありません。1つのジョブを実行するだけで済みます。</p>	
Language Environment	<ol style="list-style-type: none"> <li>1. ターゲット・ゾーンに対する境界を SMP/E 上に設定する。</li> <li>2. SMPCNTL カードに LINK LMODS CALLLIBS を指定する。また、CHECK、RETRY(YES)、RC など、他のパラメーターも指定できる。詳しくは、「SMP/E for z/OS: Commands」を参照。</li> <li>3. SMP/E ジョブを実行する。</li> </ol>	<p>処置は不要。ただし、SMP/E ゾーンが自動再リンクにセットアップされ、CSQ8SLDQ ジョブが実行されている。</p>
呼び出し可能サービス	<ol style="list-style-type: none"> <li>1. ターゲット・ゾーンに対する境界を SMP/E 上に設定する。</li> <li>2. SMPCNTL カードに LINK LMODS CALLLIBS を指定する。また、CHECK、RETRY(YES)、RC など、他のパラメーターも指定できる。詳しくは、「SMP/E for z/OS: Commands」を参照。</li> <li>3. SMP/E ジョブを実行する。</li> </ol>	<p>処置は不要。ただし、SMP/E ゾーンが自動再リンクにセットアップされ、CSQ8SLDQ ジョブが実行されている。</p>

表 53. 新しい SMP/E 環境およびライブラリーで、製品のいずれかが新しいリリースに更新された場合		
製品	CALLIBS および SMP/E V3R2 以降を使用する場合の処置	LINK を使用する場合の処置
	<p>注：Language Environment、および呼び出し可能サービスのために異なる 3 つのジョブを実行する必要はありません。両製品に対して 1 つのジョブを実行するだけで済みます。</p>	
Language Environment	<ol style="list-style-type: none"> <li>1. SCEELKED および SCEESPC のための DDDEF を新しいライブラリーを指すように変更する。</li> <li>2. ターゲット・ゾーンに対する境界を SMP/E 上に設定する。</li> <li>3. SMPCNTL カードに LINK LMODS CALLLIBS を指定する。また、CHECK、RETRY(YES)、RC など、他のパラメーターも指定できる。詳しくは、「SMP/E for z/OS: Commands」を参照。</li> <li>4. SMP/E ジョブを実行する。</li> </ol>	<ol style="list-style-type: none"> <li>1. IBM MQ for z/OS 宛先ゾーンの中で次の LMOD 項目の XZMOD 副項目を削除する。 CMQXDCST, CMQXRCTL, CMQXSUPR, CSQCBE00, CSQCBE30, CSQCBP00, CSQCBP10, CSQCBR00, CSQUCVX, CSQUDLQH, CSQVXPCB, CSQVXSPT, CSQXDCST, CSQXRCTL, CSQXSUPR, CSQXTDMI, CSQXTCP, CSQXTNSV, CSQ7DRPS, IMQB23IC, IMQB23IM, IMQB23IR, IMQS23IC, IMQS23IM, IMQS23IR</li> <li>2. IBM MQ ゾーンと言語環境プログラム・ゾーンの間に適切な ZONEINDEX をセットアップする。</li> <li>3. LINK コマンドの FROMZONE パラメーター上で新しいゾーンを参照するよう CSQ8SLDQ を調整する。CSQ8SLDQ は SCSQINST ライブラリーで見つけることができる。</li> <li>4. CSQ8SLDQ を実行する。</li> </ol>
呼び出し可能サービス	<ol style="list-style-type: none"> <li>1. CSSLIB のための DDDEF を、新しいライブラリーを指すように変更する。</li> <li>2. ターゲット・ゾーンに対する境界を SMP/E 上に設定する。</li> <li>3. SMPCNTL カードに LINK LMODS CALLLIBS を指定する。また、CHECK、RETRY(YES)、RC など、他のパラメーターも指定できる。詳しくは、「SMP/E for z/OS: Commands」を参照。</li> <li>4. SMP/E ジョブを実行する。</li> </ol>	<ol style="list-style-type: none"> <li>1. IBM MQ for z/OS 宛先ゾーンの中で次の LMOD 項目の XZMOD 副項目を削除する。 CMQXRCTL, CMQXSUPR, CSQBSRV, CSQILPLM, CSQXJST, CSQXRCTL, CSQXSUPR, CSQ3AMGP, CSQ3EPX, CSQ3REPL</li> <li>2. IBM MQ ゾーンと呼び出し可能サービス・ゾーンとの間に適切な ZONEINDEX をセットアップする。</li> <li>3. LINK コマンドの FROMZONE パラメーター上で新しいゾーンを参照するよう CSQ8SLDQ を調整する。CSQ8SLDQ は SCSQINST ライブラリーで見つけることができる。</li> <li>4. CSQ8SLDQ を実行する。</li> </ol>

CALLLIB を使用するときモジュールを再リンクするジョブの例については、769 ページの『LINK CALLLIBS ジョブの実行』を参照してください。

## ▶ z/OS LINK CALLIBS ジョブの実行

CALLLIB を使用するときモジュールを再リンクするジョブの例。

次の例は、SMP/E V3R2 システムで CALLLIB を使用するときモジュールを再リンクするジョブの例です。JOB CARD を用意し、IBM MQ for z/OS が入っている SMP/E CSI のデータ・セット名を指定する必要があります。

```
//*****  
//* RUN LINK CALLLIBS.  
//*****  
//CALLLIBS EXEC PGM=GIMSMP,REGION=4096K  
//SMPCSI DD DSN=your.csi  
// DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SMPCNTL DD *  
SET BDY(TZONE).  
LINK LMODS CALLLIBS .  
/*
```

図 121. SMP/E LINK CALLLIBS ジョブの例

## z/OS IMS における OTMA 出口の使用

このトピックは、IBM MQ for z/OS で IMS Open Transaction Manager Access 出口を使用する場合に使用します。

IMS トランザクションからの出力を IBM MQ へ送る場合、そのトランザクションが IBM MQ の中で生じたものでなければ、1つ以上の IMS OTMA 出口をコーディングする必要があります。

同様に、OTMA 以外の宛先へ出力を送る場合、トランザクションが IBM MQ の中で生じたものであれば、1つ以上の IMS OTMA 出口をコーディングする必要があります。

IMS と IBM MQ 間の処理をカスタマイズするために、IMS では、以下の出口が使用可能です。

- OTMA 事前経路指定出口
- 宛先解決ユーザー (DRU) 出口

### OTMA 出口名

事前経路指定出口の名前は DFSYPRX0 にする必要があります。DRU 出口の名前は、IMS の既存モジュール名と競合しなければ、どのような名前でも構いません。

#### 宛先解決ユーザー出口名の指定

CSQ6SYSP マクロの OTMACON キーワードにある *Druexit* パラメーターを使用して、IMS によって実行される OTMA DRU 出口の名前を指定することができます。

オブジェクトの識別を単純化するために、DRU0xxxx の命名規則の使用を検討してください。ここで、xxxx は使用中の IBM MQ キュー・マネージャーの名前です。

DRU 出口の名前を OTMACON パラメーターで指定しなければ、デフォルトは DFSYDRU0 になります。このモジュールのサンプルが、IMS によって提供されています。これについては、「IMS/ESA® カスタマイズの手引き」を参照してください。

#### IMS 宛先の命名規則

IMS プログラムから出力を送る場合、その宛先について命名規則が必要です。この宛先は、IMS アプリケーションの CHNG 呼び出しに設定されるものであるか、IMS PSB に事前設定されているものです。

### OTMA 出口のサンプル・シナリオ

IMS 用の事前経路指定出口および宛先経路指定出口の例については、以下のトピックを使用してください。

- [771 ページの『事前経路指定出口 DFSYPRX0』](#)
- [772 ページの『宛先解決ユーザー出口』](#)

識別を単純化するために、OTMA 宛先名を IBM MQ キュー・マネージャー名と同じにすることを勧めます (例えば、繰り返される IBM MQ キュー・マネージャー名)。この場合、IBM MQ キュー・マネージャー名が「**VCPE**」であれば、CHNG 呼び出しによって設定される宛先は「**VCPEVCPE**」になります。

## 関連概念

759 ページの『[IMS での IBM MQ の使用](#)』

IBM MQ-IMS アダプターおよび IBM MQ-IMS ブリッジは、IBM MQ が IMS と相互作用できるようにする 2 つのコンポーネントです。

## 関連情報

[IBM MQ および IMS クライアント](#)

[IBM MQ for z/OS での IMS および IMS ブリッジ・アプリケーション](#)

## 事前経路指定出口 DFSYPRX0

このトピックには、IMS における OTMA の事前経路指定出口のサンプルが含まれます。

最初に、事前経路指定出口 DFSYPRX0 をコード化する必要があります。IMS によってこのルーチンに渡されるパラメーターについては、「[IMS/ESA カスタマイズの手引き](#)」に説明があります。

この出口は、メッセージの宛先が既知の OTMA 宛先 (例では VCPEVCPE) に向けられたものかどうかをテストします。もしそうであれば、メッセージを送っているトランザクションが OTMA で生じたものかどうかを出口で検査する必要があります。メッセージが OTMA で生じたものであれば、それは OTMA ヘッダーを持っているので、レジスター 15 を 0 に設定して、DFSYPRX0 を終了します。

- メッセージを送っているトランザクションが OTMA の中で生じたものでなければ、クライアント名を有効な OTMA クライアントに設定する必要があります。これは、メッセージを送る IBM MQ キュー・マネージャーの XCF メンバー名です。この値の設定場所については、「[IMS/ESA カスタマイズの手引き](#)」に説明されています。使用するクライアント名 (CSQ6SYSP マクロの OTMACON パラメーターの中) は、キュー・マネージャー名に設定することをお勧めします。これはデフォルトです。次に、レジスター 15 を 4 に設定して、DFSYPRX0 を終了します。
- メッセージを送っているトランザクションが OTMA で生じたものであり、OTMA 以外の宛先であれば、レジスター 15 を 8 に設定して終了します。
- その他の場合、レジスター 15 は 0 に設定します。

OTMA クライアントの名前を、IMS が認知できない名前に設定すると、アプリケーションの CHNG 呼び出しまたは ISRT 呼び出しは A1 状況コードを戻します。

IMS システムが複数の IBM MQ キュー・マネージャーと通信している場合、ロジックを各 IBM MQ キュー・マネージャーごとに繰り返してください。

772 ページの [図 122](#) にサンプル・アセンブラー・コードが示されています:

```

TITLE 'DFSYPRX0: OTMA PRE-ROUTING USER EXIT'
DFSYPRX0 CSECT
DFSYPRX0 AMODE 31
DFSYPRX0 RMODE ANY
*
SAVE (14,12),,DFSYPRX0&SYSDATE&SYSTIME
SPACE 2
LR R12,R15          MODULE ADDRESSABILITY
USING DFSYPRX0,R12
*
L R2,12(,R1)        R2 -> OTMA PREROUTE PARMS
*
LA R3,48(,R2)        R3 AT ORIGINAL OTMA CLIENT (IF ANY)
CLC 0(16,R3),=XL16'00' OTMA ORIG?
BNE OTMAIN          YES, GO TO THAT CODE
*
NOOTMAIN DS 0H          NOT OTMA INPUT
LA R5,8(,R2)          R5 IS AT THE DESTINATION NAME
CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE EXIT0            NO, NORMAL PROCESSING
*
L R4,80(,R2)          R4 AT ADDR OF OTMA CLIENT
MVC 0(16,R4),=CL16'VCPE' CLIENT OVERRIDE
B EXIT4              AND EXIT
*
OTMAIN DS 0H           OTMA INPUT
LA R5,8(,R2)          R5 IS AT THE DESTINATION NAME
CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE EXIT8            NO, NORMAL PROCESSING

*
EXIT0 DS 0H
LA R15,0              RC = 0
B BYEBYE
*
EXIT4 DS 0H
LA R15,4              RC = 4
B BYEBYE
*
EXIT8 DS 0H
LA R15,8              RC = 8
B BYEBYE
*
BYEBYE DS 0H
RETURN (14,12),,RC=(15) RETURN WITH RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
END

```

図 122. アセンブラーで書いた OTMA 事前経路指定出口のサンプル

## 2/05 宛先解決ユーザー出口

このトピックには、IMS の宛先解決ユーザー出口のサンプルが含まれます。

DFSYPRX0 において、レジスター 15 を 4 に設定した場合、トランザクションのソースが OTMA であり、かつレジスター 15 を 0 に設定すると、DRU 出口が呼び出されます。この例では、DRU 出口の名前は DRU0VCPE です。

DRU 出口は、宛先が VCPEVCPE であるかどうかを検査します。その場合には、OTMA ユーザー・データ (OTMA 接頭部の中にある) を次のように設定します。

### Offset

OTMA ユーザー・データ

(decimal)

0

OTMA ユーザー・データの長さ (この例では、334)

## 2

### MQMD

#### 326

##### 様式化する応答

これらのオフセットは、IBM MQ - IMS ブリッジが該当する情報の存在を予想する場所です。

DRU 出口はできるだけ簡単にすることをお勧めします。したがって、このサンプルでは、特定の IBM MQ キュー・マネージャーに対して、IMS で発生するすべてのメッセージは、同じキュー IBM MQ キューに入れられます。

持続メッセージにする必要がある場合は、IMS で同期化トランザクション・パイプを使用しなければなりません。このためには、DRU 出口で OUTPUT フラグを設定する必要があります。詳細については、「IMS/ESA カスタマイズの手引き」を参照してください。

このキューを処理するために IBM MQ アプリケーションを作成し、MQMD 構造、MQIIH 構造(もしあれば)、またはユーザー・データからの情報を使用して、各メッセージをその宛先へ送ります。

サンプルのアセンブラー DRU 出口が [773 ページの図 123](#) に示されています。

```
TITLE 'DRU@VCPE: OTMA DESTINATION RESOLUTION USER EXIT'
DRU@VCPE CSECT
DRU@VCPE AMODE 31
DRU@VCPE RMODE ANY
*
SAVE (14,12),,DRU@VCPE&SYSDATE&SYSTIME
SPACE 2
LR R12,R15          MODULE ADDRESSABILITY
USING DRU@VCPE,R12
*
L R2,12(,R1)        R2 -> OTMA DRU PARMS
*
L R5,88(,R2)        R5 ADDR OF OTMA USERDATA
LA R6,2(,R5)        R6 ADDR OF MQMD
USING MQMD,R6      AS A BASE
*
LA R4,MQMD_LENGTH+10 SET THE OTMA USERDATA LEN
STH R4,0(,R5)      = LL + MQMD + 8
*
MVI 0(R6),X'00'    ...NULL FIRST BYTE
MVC 1(255,R6),0(R6) ...AND PROPAGATE IT
MVC 256(MQMD_LENGTH-256+8,R6),255(R6) ...AND PROPAGATE IT
*
VCPE DS 0H
CLC 44(16,R2),=CL16'VCPE' IS DESTINATION VCPE?
BNE EXIT4          NO, THEN DEST IS NON-OTMA
MVC MQMD_REPLYTOQ,=CL48'IMS.BRIDGE.UNSOLICITED.QUEUE'
MVC MQMD_REPLYTOQMGR,=CL48'VCPE' SET QNAME AND QMGRNAME
MVC MQMD_FORMAT,MQFMT_IMS SET MQMD FORMAT NAME
MVC MQMD_LENGTH(8,R6),MQFMT_IMS_VAR_STRING
*
B EXIT0            SET REPLYTO FORMAT NAME
*
EXIT0 DS 0H
LA R15,0           SET RC TO OTMA PROCESS
B BYEBYE          AND EXIT
*
EXIT4 DS 0H
LA R15,4           SET RC TO NON-OTMA
B BYEBYE          AND EXIT
*
BYEBYE DS 0H
RETURN (14,12),,RC=(15) RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
CMQA EQUONLY=NO
CMQMDA DSECT=YES
SPACE 2
END
```

図 123. アセンブラーで書いた DRU 出口のサンプル

IBM z/OS Management Facility(z/OSMF)は、統合されたユーザー・アシスタンスを使用して、タスク指向の Web ブラウザー・ベースのユーザー・インターフェースにシステム管理機能を提供するため、メインフレーム z/OS システムの日常の運用および管理をより簡単に管理できます。

従来のタスクの一部を簡素化し、その他を自動化することにより、z/OSMF は z/OS システム管理のいくつかの領域の単純化に貢献します。

ユーザー提供のポータルから、ボタンをクリックするだけで、リソースをプロビジョンまたはプロビジョン解除することができます。z/OSMF には、このタスクに役立つ REST API が用意されています。

z/OSMF に付属するサンプルのマーケットプレイス・ポータルを使用して、リソースのプロビジョンおよびプロビジョン解除を行うこともできます。あるいは、経験豊富なユーザーであれば z/OSMF の Web ユーザー・インターフェース (WUI) を使用することもできます。

このセクションでは、z/OSMF を理解していることを前提としていますが、z/OSMF を理解していない場合は、[z/OSMF の概要](#)を読んでください。また、このセクションには z/OSMF WUI オンライン・ヘルプからもアクセスできます。

以下のような z/OS クラウド構成を理解する必要があります。

- クラウド・プロビジョニング - リソース管理およびソフトウェア・サービス
- 構成 - 構成アシスタント、およびパフォーマンス・ワークロード管理、および
- パフォーマンス - ワークロード管理

これらについての詳細が、『[Getting Started Tutorial - Cloud](#)』とともに『[What's New in this Release](#)』セクションに記載されています。

z/OSMF 2.2 で、役割ベースのアクティビティとタスクが導入されたため、以下のような概念を理解することが重要です。

ドメイン  
管理者  
承認者  
テナント  
テンプレート  
インスタンス  
ワークフロー

などです。「[z/OSMF プログラミング・ガイド](#)」または z/OSMF WUI ヘルプの『[クラウド・プロビジョニング](#)』を参照してください。

サンプル IBM MQ z/OSMF ワークフローおよび関連ファイルが提供されています。これらを IBM MQ for z/OS UNIX System Services Components フィーチャーの一部としてインストールできます。このフィーチャーのインストール手順と、ディレクトリーおよびファイル構造については、「[IBM MQ for z/OS Program Directory](#)」(IBM Publications Center からダウンロード可能)に説明があります。

サンプル・ワークフローは XML で記述されており、IBM MQ キュー・マネージャー、チャネル・イニシエーター、ローカル・キューのプロビジョン(作成)またはプロビジョン解除(破壊)を自動化する方法、およびプロビジョンされた IBM MQ リソースに対してアクションを実行する方法を示します。これらのワークフロー内のステップでは、ジョブ(JCL)の実行依頼、REXX 実行ファイルの実行、シェル・スクリプトの処理、または REST API 呼び出しの実行を行います。

サンプルは、z/OSMF を使用して実現できる機能のタイプを示すように設計されています。リソースのプロビジョンには通常、z/OSMF ワークフローが使用されること、PUT メッセージや GET メッセージのようなアクションは本質的に IBM MQ アプリケーションを使用して行われることが予期されています。

ワークフロー変数プロパティが設定されている場合(後続のセクションで説明します)は、提供されたままのサンプル・ワークフローを実行できます。あるいは、必要に応じてカスタマイズすることもできます。追加の機能を実行するために独自のワークフローを作成することもできます。サンプル・ワークフローを実行する前に、以下を参照してください。

- [775 ページの『前提条件』](#)

• [776 ページの『セキュリティー設定』](#)

• [779 ページの『制限』](#)

サンプル・ワークフロー・アプリケーションは以下の目的で提供されています。

• [780 ページの『IBM MQ キュー・マネージャーのプロビジョンまたはプロビジョン解除の自動化、およびプロビジョンされたキュー・マネージャーに対するアクションの実行』](#)

• [781 ページの『IBM MQ ローカル・キューのプロビジョンまたはプロビジョン解除の自動化、およびプロビジョンされたキューに対するアクションの実行』](#).

## 関連概念

[661 ページの『IBM MQ for z/OS のセットアップ』](#)

このトピックを、IBM MQ for z/OS システムのカスタマイズのステップバイステップ・ガイドとしてご利用ください。

V 9.0.1

z/OS

## 前提条件

IBM MQ で IBM z/OS Management Facility (z/OSMF) を実行するために必要な前提条件

IBM MQ 9.0.1 で提供される IBM MQ ワークフローは、z/OS 2.1 と 2.2 の両方で APAR を介して提供される z/OSMF の新機能を活用します。詳細を、以下のテキストで説明します。

1. IBM z/OS Management Facility 2.2 が正しくインストールされ、構成されている。セキュリティーを有効にして実行している場合は、z/OSMF に説明があるように、すべてのセキュリティー設定が構成されている必要があります。
2. 以下に対する APAR がインストールされている。

### z/OS 2.1

- PI71068
- PI71079
- PI71082
- PI71084
- OA50130

### z/OS 2.2

- PI70526
- PI70521
- PI70527
- PI67839
- PI70767
- PI46315
- OA49081
- OA49802
- OA50130

3. z/OSMF のエンジェル・プロセス (必要な場合) およびサーバー・プロセスが構成されている。
4. z/OS クラウド環境が構成されている (上記で少し触れていますが z/OSMF に説明があります)
5. IBM MQ for z/OS 9.0.1 がインストールされていて、製品ロード・ライブラリーが使用可能である。
6. 以下の IBM MQ キュー・マネージャーのカスタマイズ・タスクが実行済みである。

タスク	説明
1	z/OS システム・パラメーターを識別する
2	IBM MQ ロード・ライブラリーの APF 許可を行う

タスク	説明
3	z/OS リンク・リストおよび LPA を更新する
4	z/OS プログラム特性表を更新する

7. サンプル・ワークフローおよび関連ファイルが適切な UNIX System Services for z/OS (USS) ディレクトリーにインストールされている。
8. provision.xml ワークフローが一時ファイルを **'/tmp'** USS ディレクトリーに作成できるように、このディレクトリーが使用可能である。ファイルが作成された場合、通常は使用された後にワークフローによってそのファイルは削除されます。
9. deprovision.xml ファイル内に、REXX 実行ファイル CSQ4ZWS1.rexx および CSQ4ZWS2.rexx を呼び出すステップが含まれている。これらの実行ファイルは、キュー・マネージャーおよびチャネル・イニシエーター・サブシステムが停止するまで待機します。つまり、実行ファイルが USS 「SLEEP」コマンドをシステム呼び出しとして起動します。

ご使用の USS 構成によっては、コードされたとおりに「SLEEP」コマンドが機能しないことがあります。処理中に「SLEEP」コマンドが見つからないことを示すエラーが発生した場合は、実行ファイル CSQ4ZWS1.rexx および CSQ4ZWS2.rexx 内の以下の行を置き換えてみてください。

```
CALL SYSCALLS('ON')           /* Enable USS calls */
ADDRESS SYSCALL
"SLEEP" 10                     /* Sleep for 10 seconds */
CALL SYSCALLS 'OFF'          /* Disable USS calls */
```

を次のコマンドに置き換えてください:

```
'sleep' 10
```

次に、Open MVS (OMVS) **env** コマンドを実行して PATH 環境変数の設定値を確認します。sleep コマンドが含まれているディレクトリーが PATH に定義されていることを確認します。通常、sleep コマンドは /bin ディレクトリーにあることに留意してください。

10. z/OSMF が始動していることを確認する。

z/OSMF のエンジェル・プロセスとサーバー・プロセスの両方が開始されていて、z/OSMF Web ユーザー・インターフェース (WUI) が稼働している必要があります。詳細については、『[Liberty profile: Process types on z/OS](#)』を参照してください。

REST API を使用してワークフローを起動する場合でも、z/OSMF WUI を開始する必要があります。z/OSMF WUI は、ワークフローの作成と実行をモニターするのに役立ちます。

## 関連概念

774 ページの『[IBM z/OSMF を使用した IBM MQ の自動化](#)』

IBM z/OS Management Facility(z/OSMF)は、統合されたユーザー・アシスタンスを使用して、タスク指向の Web ブラウザー・ベースのユーザー・インターフェースにシステム管理機能を提供するため、メインフレーム z/OS システムの日常の運用および管理をより簡単に管理できます。

## V 9.0.1 z/OS セキュリティー設定

z/OSMF の実行に必要なセキュリティー設定です。

以下のユーザー ID 変数プロパティーは、properties ファイルに定義されています。詳細については、784 ページの『[ワークフローの実行](#)』を参照してください。

ユーザー ID プロパティー	説明
CSQ_USERID	ワークフロー・ステップの実行に使用するユーザー ID。ただし、選択されたステップ (一般に、昇格された権限レベルが必要なもの) は、下記にリストしている一連の <b>CSQ_ADMIN_*</b> のユーザー ID の設定を基に別のユーザー ID で実行されることに注意してください。使用するユーザー ID は、ワークフローの各ステップの <b>runAsUser</b> プロパティーで指定します。

ユーザー ID プロパティ	説明
CSQ_ADMIN_APF_USERID	キュー・マネージャーのシステム・パラメーター・モジュールを含むロード・ライブラリーを APF が許可するとき使用するユーザー ID。
CSQ_APF_APPROVAL_ID	データ・セット APF 許可ステップをユーザー CSQ_ADMIN_APF_USERID として実行することをユーザーに許可するために使用する承認 ID。
CSQ_ADMIN_CONSOLE_USERID	z/OS コンソール・コマンドを発行するステップの実行に使用するユーザー ID。  <b>重要:</b> このユーザー ID には、開始タスク・プロファイル (MVS.START.STC. *) への UPDATE アクセスを許可する必要があります。"OPERCMD5" クラスにあります。詳しくは、IBM Documentation の IBM z/OS 情報の「z/OS オペレーター・コンソール・オペレーション」セクションを参照してください。
CSQ_CONSOLE_APPROVAL_ID	z/OS コンソール・コマンドを発行するステップをユーザー CSQ_ADMIN_CONSOLE_USERID として実行することをユーザーに許可するために使用する承認 ID。
CSQ_ADMIN_SAF_USERID	SAF コマンドを実行するとき使用するユーザー ID。
CSQ_SAF_APPROVAL_ID	SAF コマンド・ステップをユーザー CSQ_ADMIN_SAF_USERID として実行することをユーザーに許可するために使用する承認 ID。
CSQ_ADMIN_SSI_USERID	SETSSI コマンドを実行して z/OS にプロビジョンされるサブシステムを識別するとき使用するユーザー ID。
CSQ_SSI_APPROVAL_ID	SETSSI コマンド・ステップをユーザー CSQ_ADMIN_SSI_USERID として実行することをユーザーに許可するために使用する承認 ID。

注：ワークフローのプロビジョンおよびプロビジョン解除の実行に使用するユーザー ID には、以下にリストする十分な権限がなければなりません。

1. キュー・マネージャーのプロビジョンおよびプロビジョン解除のワークフローは、SETPROG コマンドを使用してデータ・セットの APF 許可を行います。ユーザー ID をプロパティ CSQ\_ADMIN\_APF\_USERID に設定するか、ワークフローの実行に使用するユーザー ID にこのコマンドを実行する権限を付与する必要があります。これを行うには、以下のコマンドを実行します。

```
PERMIT MVS.SETPROG CLASS(OPERCMD5) ID(value of CSQ_ADMIN_APF_USERID) ACCESS(UPDATE)
```

注：SETPROG コマンドは z/OS システムの IPL 後は持続されない可能性があるため、IPL 後は次の SETPROG コマンドを手動で実行する必要があります。

```
SETPROG APF,ADD,DSN=value of CSQ_AUTH_LIB_HLQ.value of CSQ_SSID.APF.LOAD,SMS
```

SETPROG コマンドの詳細については、[RACF を使用した APF リストの制御](#)を参照してください。

さらに、どのライブラリーに APF 許可を付与するかを制御するために FACILITY クラスを有効にした可能性があるため、次のコマンドを実行する必要があります。

```
PERMIT CSVAPF.libname CLASS(FACILITY) ID(value of CSQ_ADMIN_APF_USERID) ACCESS(UPDATE)
```

2. キュー・マネージャーのプロビジョン・ワークフロー内のステップにより、z/OS に対して IBM MQ サブシステムを識別するための SETSSI コマンドが発行されます。プロパティ

CSQ\_ADMIN\_SSI\_USERID に設定されているユーザー ID が、このコマンドの使用を許可されている必要があります。これを行うには、以下のコマンドを実行します。

```
PERMIT MVS.SETSSI.ADD CLASS(OPERCMD) ID(value of CSQ_ADMIN_SSI_USERID)
ACCESS(CONTROL)
```

注: SETSSI コマンドを使用して z/OS に対して識別されたサブシステムは、z/OS システムの IPL 後は持続しません。そのため、IPL 後は次の SETSSI コマンドを手動で実行しなければならないことがあります。

```
SETSSI ADD,S='value of CSQ_SSID',I=CSQ3INI,
P='CSQ3EPX,value of CSQ_CMD_PFX,S'
```

SETSSI コマンドの詳細については、[SETSSI コマンド](#)を参照してください。

3. ワークフローはキュー・マネージャー・コマンドを実行するので、セキュリティを有効にする予定の場合は、プロパティ CSQ\_ADMIN\_RACF\_USERID に設定されているユーザー ID (またはワークフローの実行に使用されているユーザー ID) に対し、(使用するクラスに応じて) MQADMIN クラスまたは MXADMIN クラスの CLAUTH (クライアント認証) 権限を (使用するクラスに応じて) 付与する必要があります。これにより、このユーザー ID がこれらのクラスに対してセキュリティ・プロファイルを定義できるようになります。これを行うには、以下のコマンドを実行します。

```
ALTUSR value of CSQ_ADMIN_RACF_USERID CLAUTH(MQADMIN)
```

**CLAUTH** について詳しくは、[CLAUTH \(クラス権限\) 属性](#)を参照してください。

4. deprovision.xml ワークフローは z/OS コマンド (例えばジョブの DISPLAY ACTIVE、サブシステムの CANCEL および FORCE) を実行するので、プロパティ CSQ\_ADMIN\_CONSOLE\_USERID に設定するユーザー ID (あるいはワークフローの実行に使用するユーザー ID) は、そのようなコマンドを実行するための適切な権限を持っている必要があります。
5. ソフトウェア・サービス・タスクのテンプレート・テーブルを使用してキュー・マネージャー・インスタンスを要求しているユーザーは、z/OSMF および構成アシスタントにアクセスするための権限 (z/OSMF で定義されている) を持っている必要があります。
6. キュー・マネージャーをプロビジョンするコンシューマーのユーザー ID には、変数 CSQ\_PROC\_LIB で定義された PROCLIB データ・セットのメンバーを追加および削除する権限が必要です。
7. キューをプロビジョンする前に、キュー・マネージャーをプロビジョンする必要があります。
8. queueLoad.xml ワークフローおよび queueOffload.xml ワークフローを使用するには、使用するデータ・セットをあらかじめ定義しておく必要があります。また、これらのワークフローの実行に使用されるユーザー ID は、データ・セットに対する UPDATE 権限が付与されている必要があります。
9. キュー・マネージャーの provision.xml ワークフロー内のステップは、現在はサブシステム・セキュリティを無効にします。IBM MQ リソースを保護するための適切なセキュリティ・コマンドを追加することにより、ジョブ csq4znse.jcl を変更して、サブシステム・セキュリティを使用可能にする。ただし、コマンドを追加する場合は、deprovision.xml ワークフローが実行依頼する csq4dse.jcl でセキュリティ権限を削除するコマンドも追加する必要があることに注意してください。

注: このステップは、RACF セキュリティ・コマンドを発行します。別のセキュリティ製品を使用する場合は、このステップを変更して、そのセキュリティ製品の適切なコマンドを発行してください。

## ネットワーク要件

キュー・マネージャーのテンプレートおよびそのテンプレート用のリソースを追加する場合は、「**ネットワーク・リソース・プールの作成 (Create network resource pool)**」をクリックします。これによって、そのテンプレートのネットワーク・リソースを持つリソース・プールが作成されます。

ネットワーク管理者は、構成アシスタントでそのテンプレートに割り振られるポート数の限度を定義して、このネットワーク・リソース・プール定義を完了する必要があります。

各テンプレート・インスタンスについて、provision.xml ワークフローは範囲内のポートを1つ割り振り、そのポートでlistenするリスナーを開始します。

## IBM ワークロード・マネージャーによる分類

WLMを使用してキュー・マネージャーおよびチャンネル・イニシエーターのアドレス・スペースを分類する場合は、キュー・マネージャーのプロビジョニング用テンプレートを追加する際にこの分類を指定する必要があります。

分類するかどうかは、ファイルworkflow\_variables.propertiesに設定されているフラグ **CSQ\_DEFINE\_MSTR\_WLM\_RULE** および **CSQ\_DEFINE\_CHIN\_WLM\_RULE** によって制御されます。

WLMを使用した分類について詳しくは、「z/OSMF 構成ガイド」を参照してください。

### 関連概念

775 ページの『前提条件』

IBM MQ で IBM z/OS Management Facility (z/OSMF) を実行するために必要な前提条件

### V 9.0.1 z/OS 制限

IBM MQ で z/OSMF を使用する場合の制限です。

1. 現在、provision.xml ワークフローは、以下の強調表示されたキュー・マネージャーのカスタマイズ・タスクを自動化します。

タスク	説明
1	z/OS システム・パラメーターを識別する
2	IBM MQ ロード・ライブラリーを APF 許可する ( <b>provision.xml は一部のライブラリーの APF 許可を行います</b> )
3	z/OS リンク・リストおよび LPA を更新する
4	z/OS プログラム特性表を更新する
5	<b>IBM MQ サブシステムを z/OS に定義する</b>
6	<b>IBM MQ キュー・マネージャー用のプロシージャーを作成する</b>
7	チャンネル・イニシエーター用のプロシージャーを作成する
8	<b>IBM MQ サブシステムを z/OS WLM サービス・クラスに定義する</b>
9	カップリング・ファシリティ・オフロード・ストレージ環境の選択と設定
10	カップリング・ファシリティのセットアップ
11	ESM セキュリティ管理を実施する
12	SYS1.PARMLIB メンバーを更新する
13	初期設定入力データ・セットをカスタマイズする
14	ブートストラップ・データ・セットおよびログ・データ・セットを作成する
15	ページ・セットを定義する
16	IBM MQ 項目を Db2 データ共有グループに追加する
17	システム・パラメーター・モジュールを調整する (一部)
18	チャンネル・イニシエーター・パラメーターを調整する (一部)
19	バッチ、TSO、および RRS アダプターをセットアップする
20	操作パネルおよび制御パネルをセットアップする

タスク	説明
21	IBM MQ ダンプ・フォーマット・メンバーを組み込む
22	通知メッセージを抑止する
23	Advanced Message Security 用にシステム DIAG メンバーを更新する
24	Advanced Message Security 用のプロシージャーを作成する
25	開始タスク・ユーザー Advanced Message Security をセットアップする
26	Advanced Message Security のセキュリティー管理者に RACDCERT 権限を付与する
27	Advanced Message Security のためのリソース・アクセス権をユーザーに付与する

2. 太字のテキストで強調表示されていないカスタマイズ・タスクは、必要な場合に手動で実行する必要があります。
3. サンプルの INP1 メンバー および INP2 メンバーは現在もそのまま使用されます。必要な場合は、追加のプロパティを定義して、これらのメンバーによって定義されたリソースを制御できます。
4. properties ファイルにリストされている特定のプロパティに関連するコメントは、それらのプロパティの使用に関する制約を示しています。詳細については、[784 ページの『ワークフローの実行』](#)を参照してください。

#### 関連概念

[776 ページの『セキュリティー設定』](#)

z/OSMF の実行に必要なセキュリティー設定です。

V 9.0.1

z/OS

### IBM MQ オブジェクトのプロビジョニングの自動化

キュー・マネージャーおよびローカル・キューのプロビジョニングを自動化するためのサンプルが用意されています。

### IBM MQ キュー・マネージャーのプロビジョンまたはプロビジョン解除の自動化、およびプロビジョンされたキュー・マネージャーに対するアクションの実行

以下のキュー・マネージャー固有のサンプル z/OSMF ワークフローが用意されています。

ワークフロー名	説明
provision.xml	<p>IBM MQ for z/OS キュー・マネージャーをプロビジョンする</p> <p>このサンプル・ワークフローは以下のことを行います。</p> <ul style="list-style-type: none"> <li>• キュー・マネージャーに必要なシステム・リソースをプロビジョンします。</li> <li>• チャンネル・イニシエーターに必要なシステム・リソースをプロビジョンします。</li> <li>• キュー・マネージャーを開始します (これにより、チャンネル・イニシエーターおよび TCP/IP リスナーも開始します)</li> <li>• サンプル・キュー・マネージャーのインストール検証プログラムを実行します。</li> </ul> <p>環境プロパティを設定して、異なる特性を持つキュー・マネージャーのプロビジョニングを制御できます。詳しくは、<a href="#">784 ページの『ワークフローの実行』</a>を参照してください。</p>

ワークフロー名	説明
	<p><b>注:</b> このワークフロー用のテンプレートの追加に役立つマニフェスト・ファイル (provision.mf) が用意されています。このファイルには追加の情報を含む <b>qaas_readme.pdf</b> ファイルへの参照が組み込まれています。テンプレートが追加されると、リンクを介してこのファイルにアクセスできるようになります。</p>
deprovision.xml	<p>IBM MQ for z/OS キュー・マネージャーをプロビジョン解除する</p> <p>このサンプル・ワークフローは以下のことを行います。</p> <ul style="list-style-type: none"> <li>• チャンネル・イニシエーターを停止します (これにより、TCP/IP リスナーも停止します) また、キュー・マネージャーも停止します。</li> <li>• サブシステムが停止するのを待機します</li> <li>• すべてのチャンネル・イニシエーターおよびキュー・マネージャーのシステム・リソースをプロビジョン解除します。</li> </ul>
startQMgr.xml	<p>IBM MQ for z/OS キュー・マネージャーを始動する</p> <p>このサンプル・ワークフローはキュー・マネージャーを開始します (これにより、チャンネル・イニシエーターおよび TCP/IP リスナーも開始します)。</p>
stopQMgr.xml	<p>IBM MQ for z/OS キュー・マネージャーを停止する</p> <p>このサンプル・ワークフローは、チャンネル・イニシエーター (TCP/IP リスナーも停止します) とキュー・マネージャーを停止します。</p>

各ワークフローは、1 つ以上のステップを実行します。ワークフロー内のコメントは、各ステップで実行される機能について説明しています。ステップには、データ入力を要求するだけのものもあれば、定義された機能を遂行するために JCL を実行依頼したり、REXX 実行ファイルやシェル・スクリプトを呼び出したリ、REST API 呼び出しを実行したりするものもあります。

JCL または REXX 実行ファイルの正確な名前については、各ステップを参照してください。ワークフローおよび関連する JCL または REXX 実行ファイルは、1 つ以上の変数 XML ファイルで宣言されている変数を参照します。詳細については、783 ページの『ワークフロー変数宣言ファイル』を参照してください。

**deprovision**、**startQMgr**、および **stopQMgr** は、プロビジョンされた IBM MQ for z/OS キュー・マネージャーに対するアクションとして実行できます。

## IBM MQ ローカル・キューのプロビジョンまたはプロビジョン解除の自動化、およびプロビジョンされたキューに対するアクションの実行

以下のキュー固有のサンプル z/OSMF ワークフローが用意されています。

ワークフロー名	説明
defineQueue.xml	<p>ローカル・キューを定義する</p> <p>このサンプル・ワークフローでは、z/OSMF ワークフローをどのように使用して、プロパティ設定に基づいた小規模、中規模、大規模のキューを定義できるかが示されています。</p> <p><b>注:</b> このワークフロー用のテンプレートの追加に役立つマニフェスト・ファイル (provision.mf) が用意されています。このファイルには追加の情報を含む <b>qaas_readme.pdf</b> ファイルへの参照が組み込まれています。テンプレートが追加されると、リンクを介してこのファイルにアクセスできるようになります。</p>
displayQueue.xml	ローカル・キューの選択した属性を表示する

ワークフロー名	説明
	<p>このサンプル・ワークフローは、ローカル・キューの選択した属性を表示します。属性は、z/OSMF 変数 (変数名についてはワークフロー内のステップを参照してください) で返され、その後に表示されます。必要な場合は、REST API を使用して変数の内容にアクセスすることができます。</p> <p>詳しくは、「z/OSMF Programming Guide」に記載されている『REST APIs for Cloud Provisioning』、および『z/OSMF workflow services』を参照してください。</p>
deleteQueue.xml	<p>ローカル・キューを削除する</p> <p>このサンプル・ワークフローは、指定されたキュー・マネージャーのローカル・キューを削除します。</p>
putQueue.xml	<p>1つ以上のメッセージをローカル・キューに書き込む。</p> <p>このサンプル・ワークフローは1つ以上のメッセージをローカル・キューに書き込みます。メッセージ・テキストは指定できますが、複数のメッセージをローカル・キューに同時に書き込むと、同じメッセージ・テキストが使用されます。</p>
getQueue.xml	<p>1つ以上のメッセージをローカル・キューから読み取る。</p> <p>このサンプル・ワークフローは1つ以上のメッセージをローカル・キューから読み取ります。メッセージは、z/OSMF 変数 (変数名についてはワークフロー内のステップを参照してください) で返され、その後に表示されます。必要な場合は、REST API を使用して変数の内容にアクセスすることができます。</p> <p>詳細については、<a href="#">Cloud provisioning REST APIs</a> を参照してください。また、<a href="#">z/OSMF ワークフロー・サービス</a> も参照してください。</p>
loadQueue.xml	<p>データ・セットからローカル・キューにメッセージをロードする。</p> <p>このサンプル・ワークフローは、メッセージをデータ・セットからローカル・キューにロードします。データ・セットのデフォルト名は、プロパティを設定することによって指定します。詳細については、<a href="#">784 ページの『ワークフローの実行』</a>を参照してください。</p>
offloadQueue.xml	<p>メッセージをローカル・キューからデータ・セットにオフロードする。</p> <p>このサンプル・ワークフローは、メッセージをローカル・キューからデータ・セットにオフロードします。データ・セットのデフォルト名は、プロパティを設定することによって指定します。詳細については、<a href="#">784 ページの『ワークフローの実行』</a>を参照してください。</p>
clearQueue.xml	<p>ローカル・キューのメッセージを消去する。</p> <p>このサンプル・ワークフローは、ローカル・キューのすべてのメッセージを消去 (削除) します。</p>

注:

1. キューへの書き込み (**PutQueue**) アクションを使用すると、いくつかのメッセージ・データを入力し、1つ以上のメッセージをキューに書き込むことができます。1つの要求で複数のメッセージがキューに書き込まれた場合は、同じメッセージ・データが使用されます。
2. loadQueue.xml および offloadQueue.xml ワークフローは、IBM MQ for z/OS QLOAD ユーティリティを呼び出します。これは、基本的に IBM MQ for Multiplatforms で使用可能な **dmpmqmsg** ユーティリティです。したがって、データ・セットからキューに、またはキューからデータ・セットにロードされるメッセージは、**dmpmqmsg** 形式であると想定されます。

loadQueue および offloadQueue アクションを試す最も簡単な方法は、以下のとおりです。

- a. **putQueue** を数回実行して、いくつかのメッセージをキューに書き込みます。
- b. **offloadQueue** を使用して、メッセージをキューからデータ・セットにオフロードします。
- c. 必要であれば、**clearQueue** を実行してキューからすべてのメッセージを除去します。
- d. **loadQueue** を使用して、メッセージをデータ・セットから同じまたは別のキューにロードします。

**dmpmqmsg** 形式を確認するには、オフロード要求を発行した後に、データ・セットのコンテンツをブラウズしてください。

3. **displayQueue**、**deleteQueue**、**putQueue**、**getQueue**、**loadQueue**、**offloadQueue**、および **clearQueue** を、プロビジョンされた IBM MQ for z/OS ローカル・キューに対するアクションとして実行できます。アクションおよびアクション・ファイルについては、「[z/OSMF プログラミング・ガイド](#)」を参照してください。
4. デフォルトでは、ワークフローに関連するすべてのアクションが削除されます。これは、ユーザーがワークフローをクリーンアップする必要性を最小限にするためです。

ただし、ここで問題なのは、アクションが何らかの出力を生成する場合です。例えば、**displayQueue** アクションと **getQueue** アクションの両方が出力を生成します。

アクションの実行が完了するとすぐに、関連するワークフローが削除されるため、出力は表示されません。したがって、z/OS WUI からワークフロー・アクションを駆動する場合は、出力を表示するアクションごとに、**< workflow>** タグの **cleanAfterComplete** フラグを **false** に設定する必要があります。

例えば、**displayQueue** の出力を表示するには、フラグを以下のように設定します。

```
<action name="displayQueue">
  <workflow cleanAfterComplete="false">
    ...
  </workflow>
</action>
```

ただし、これは、アクションに関連するワークフローを手動でクリーンアップしなければならないことを意味します。

各サンプル z/OSMF ワークフローは、1 つ以上のステップを実行します。ワークフロー内のコメントは、各ステップで実行される機能について説明しています。ステップには、データ入力を要求するだけのもの、JCL を実行依頼するもの、REXX 実行ファイル呼び出しで記述されている機能を実行するものなどがあります。

JCL または REXX 実行ファイルの正確な名前については、各ステップを参照してください。ワークフローおよび関連する JCL または REXX 実行ファイルは、1 つ以上の [783 ページの『ワークフロー変数宣言ファイル』](#) で宣言されている変数を参照します。

## 関連概念

[779 ページの『制限』](#)

IBM MQ で z/OSMF を使用する場合の制限です。

## V 9.0.1 z/OS ワークフローの実行

サンプル z/OSMF ワークフローが参照するファイルと、ワークフローの実行方法についての説明です。

## ワークフロー変数宣言ファイル

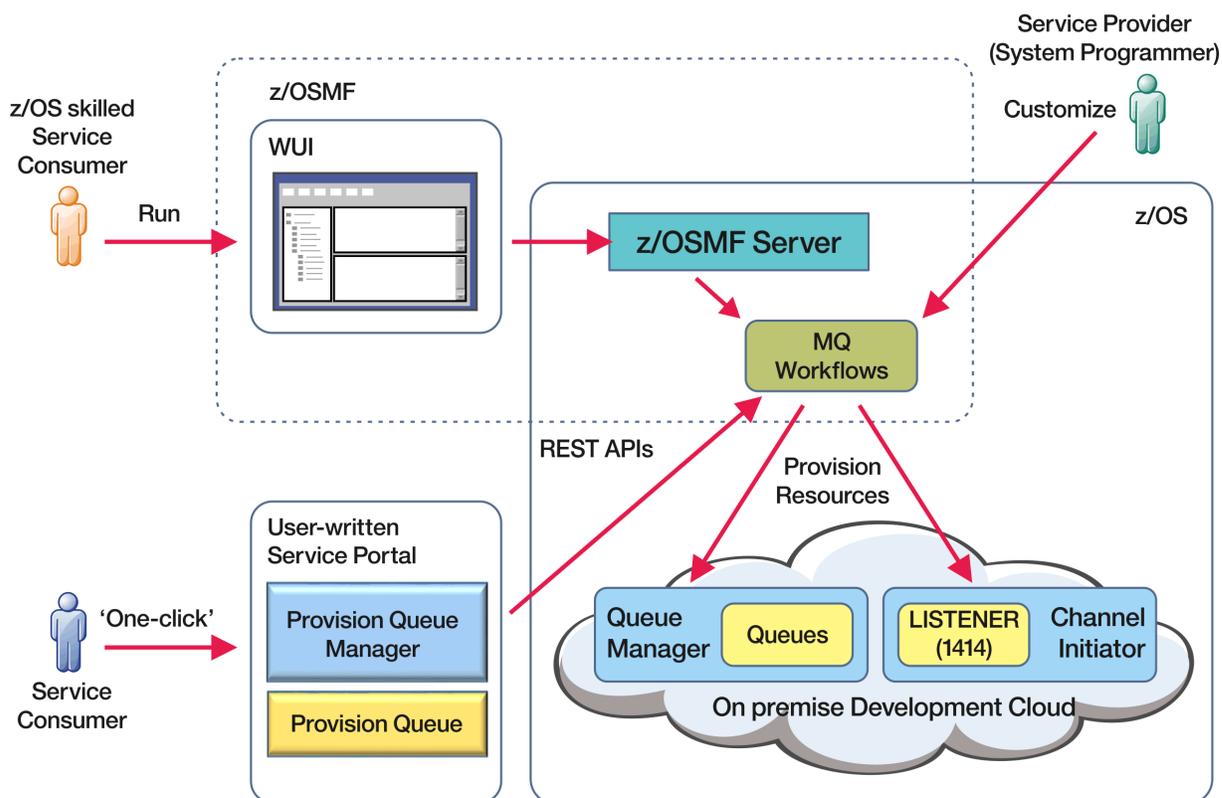
以下のファイルは、サンプル z/OSMF ワークフローおよび関連する JCL ファイルまたは REXX 実行ファイルが参照する変数を宣言します。

ワークフロー変数宣言ファイル名	説明
common_variables.xml	キュー・マネージャー (およびチャネル・イニシエーター) とキュー・ワークフローの両方に共通の変数。
qmgr_variables.xml	キュー・マネージャー (およびチャネル・イニシエーター) ワークフロー固有の変数。
queue_variables.xml	キュー・ワークフロー固有の変数。
tcPIP_variables.xml	キュー・マネージャー (およびチャネル・イニシエーター) ワークフロー固有で、TCP/IP リソースの識別に使用される変数。

注: 変数のデフォルトの可視性は *private* です。z/OSMF REST API を使用して変数を照会できるように、選択された変数に *public* というマークが付けられています。しかし、必要に応じて変数の可視性は変更することができます。

## ワークフローの実行

図 124. IBM MQ for z/OS リソースの「ワンクリック」プロビジョニング



ワークフローを実行する前に、一部のプロパティを以下のファイルに設定する必要があります。

ワークフロー変数プロパティ・ファイル名	説明
workflow_variables.properties	ワークフロー変数の初期プロパティ。ファイル内のコメントに各プロパティの目的が示されています。 <ul style="list-style-type: none"> <li>メタ括弧 (&lt;&gt;) 内のプロパティは、ユーザー固有の値に設定する必要があります。</li> </ul>

ワークフロー変数プロパティ・ファイル名	説明
	<ul style="list-style-type: none"> <li>環境プロパティを設定して、開発 (DEV) 環境、テスト (TEST) 環境、品質保証 (QA) 環境、実動 (PROD) 環境用にキュー・マネージャーをプロビジョンできます。</li> <li>追加のプロパティ設定で、各環境用にプロビジョンされるキュー・マネージャーの特性を制御します。例えば、環境タイプごとにアクティブ・ログ数またはページ・セット数を変更することができます。</li> <li>その他のプロパティは IBM MQ デフォルト値に設定されますが、必要に応じてローカルの規則に合わせて変更できます。</li> </ul>

通常、プロパティを設定したら、ワークフローはそのまま実行できます。ただし、必要であれば、ワークフローをカスタマイズして既存のステップを変更または除去したり、新規ステップを追加したりできます。

ワークフローは、以下のように実行できます。

- z/OSMF WUI から。

WUI の「クラウド・プロビジョニング」->「ソフトウェア・サービス」から、自動モードまたは手動モードでワークフローを実行できます。テストの際は手動モードが便利です。また、両方のモードで、ワークフロー内の各ステップの進行状況をモニターできます。

詳細については、z/OSMF WUI ヘルプの「クラウド・プロビジョニング」を参照してください。また、[ワークフローの作成](#)も参照してください。

- z/OSMF REST Workflow Services の使用。

REST Workflow Services を使用して REST API を介してワークフローを実行できます。このモードは、ユーザー作成ポータルからのワンクリック操作を作成する際に役立ちます。

詳細については、[Cloud provisioning REST APIs](#) を参照してください。また、[z/OSMF ワークフロー・サービス](#)も参照してください。

- z/OSMF に用意されているサンプルのマーケットプレイス・ポータルを使用する。

## 関連概念

780 ページの『[IBM MQ オブジェクトのプロビジョニングの自動化](#)』

キュー・マネージャーおよびローカル・キューのプロビジョニングを自動化するためのサンプルが用意されています。

z/OS

V 9.0.3

MQ Adv. VUE

## IBM MQ Advanced for z/OS VUE の構成

この情報を使用して、IBM MQ Advanced for z/OS VUE ライセンスで使用可能な機能を構成します。

### このタスクについて

IBM MQ 9.0.3 以降では、IBM MQ Advanced for z/OS VUE Connector Pack で提供される機能を使用して、z/OS 上の MFT トポロジーを簡素化し、IBM MQ Advanced for z/OS, Value Unit Edition キュー・マネージャーから IBM Cloud (formerly Bluemix) の IBM Blockchain サービスへの接続を利用することができます。

V 9.0.4

IBM MQ 9.0.4 以降では、クライアント接続を使用して、**ADVCAP(ENABLED)** 属性を持つ z/OS 上のキュー・マネージャーに IBM MQ classes for JMS または IBM MQ classes for Java アプリケーションを接続できます。

### 手順

1. IBM MQ Advanced for z/OS, Value Unit Edition を使用した Managed File Transfer エージェント・リモート接続の有効化。

2. IBM Cloud の IBM Blockchain サービスで使用するために IBM MQ Advanced for z/OS VUE を構成します。

## IBM Cloud (formerly Bluemix) の IBM Cloud Product Insights サービスで使用するための IBM MQ Advanced for z/OS VUE の構成

IBM Cloud Product Insights サービスは使用可能ではなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## IBM Cloud (formerly Bluemix) での IBM Cloud Product Insights サービス・インスタンスの作成

IBM Cloud Product Insights サービスは使用可能ではなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## IBM Cloud (formerly Bluemix) の IBM Cloud Product Insights サービス・インスタンスで使用するための z/OS キュー・マネージャーの構成

IBM Cloud Product Insights サービスは使用可能ではなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## HTTP プロキシを介した IBM Cloud の IBM Cloud Product Insights への接続

IBM Cloud Product Insights サービスは使用可能ではなくなりました。詳しくは、ブログ投稿 [Service Deprecation: IBM Cloud Product Insights](#) を参照してください。

## MFT エージェントからリモート z/OS キュー・マネージャーへの接続

z/OS 上の Managed File Transfer エージェントは、IBM MQ Advanced for z/OS VUE の製品 ID(PID)の下で実行されており、クライアント接続を使用して、z/OS 上のリモート・キュー・マネージャーに接続することができます。

詳しくは、[IBM MQ Advanced for z/OS, Value Unit Edition のみを対象にしたエージェント・リモート接続の有効化](#)を参照してください。

## ブロックチェーンで使用するための IBM MQ Advanced for z/OS VUE の構成

z/OS キューマネージャーと IBM Blockchain を安全に接続する IBM MQ ための IBM MQ Bridge to blockchain セットアップと実行。IBM MQ Advanced キュー・マネージャーに接続されているメッセージ・アプリケーションを使用すると、ブリッジを使用して、ブロックチェーンのリソースの状態に非同期で接続し、この状態を検索および更新できます。

### 始める前に

- IBM MQ Bridge to blockchain は、IBM MQ Advanced for z/OS Value Unit Edition 9.0.4 の Connector Pack の一部として提供されています。同じコマンド・レベルで実行されている z/OS キュー・マネージャーに接続できます。
- IBM MQ Bridge to blockchain は、Hyperledger Fabric 1.0 architecture に基づくブロックチェーン・ネットワークでの使用がサポートされています。

- 以下の製品がインストールされている x86 Linux 環境で、IBM MQ Bridge to blockchain をインストールし、構成し、実行する必要があります。
  - IBM MQ 9.0.3 再配布可能 Java クライアント。
  - IBM Java runtime environment バージョン 8。

IBM MQ 9.0.4 再配布可能 Java クライアントおよび IBM Java runtime environment バージョン 8 が既にインストールされている場合は、ステップ [788 ページの『4』](#) および [789 ページの『5』](#) を実行する必要はありません。

## このタスクについて

ブロックチェーンとは、分散型の共有デジタル台帳であり、ネットワーク上のピアの間で合意されたトランザクションを表すブロックがチェーン状に連なったもので構成されます。チェーン内のそれぞれのブロックは前のブロックにリンクしているため、最終的には最初のトランザクションにリンクしています。

IBM Blockchain は、Hyperledger Fabric 上に構築され、Docker でローカルに開発するか、IBM Cloud (formerly Bluemix) のコンテナ・クラスターで開発できます。また、実稼働環境で IBM Blockchain ネットワークをアクティブ化して使用し、セキュリティー、プライバシー、およびパフォーマンスのレベルの高いビジネス・ネットワークを構築し、管理することもできます。詳細については、[IBM Blockchain Platform](#) を参照してください。

Hyperledger Fabric は、Hyperledger Project のメンバー (初期コード・コントリビューターとして IBM を含む) によって共同で開発された、オープン・ソースのエンタープライズ・ブロックチェーン・フレームワークです。Hyperledger Project、または Hyperledger は、業界横断のブロックチェーン・テクノロジーを推進するための Linux Foundation オープン・ソースのグローバルなコラボレーション・イニシアチブです。詳細については、[IBM Blockchain](#)、[Hyperledger Projects](#)、および [Hyperledger Fabric](#) を参照してください。

IBM MQ Advanced for z/OS VUE と IBM Blockchain を既に使用している場合は、IBM MQ Bridge to blockchain を使用して、単純な照会や更新情報を送信してブロックチェーン・ネットワークから応答を受け取ることができます。そのようにして、オンプレミスの IBM ソフトウェアをクラウドのブロックチェーン・サービスに統合できます。

ブリッジ操作プロセスの概要は、[Figure 1](#) に記載されています。ユーザー・アプリケーションは、JSON 形式のメッセージを z/OS キュー・マネージャー上の入力/要求キューに入れます。ブリッジはキュー・マネージャーに接続し、入力/要求キューからメッセージを取得し、JSON が正しくフォーマット設定されていることを確認してから、照会または更新をブロックチェーンに発行します。ブロックチェーンによって返されるデータは、ブリッジによって解析され、元の IBM MQ 要求メッセージに定義されているように応答キューに入れられます。ユーザー・アプリケーションは、キュー・マネージャーに接続して、応答キューから応答メッセージを取得し、その情報を使用できます。

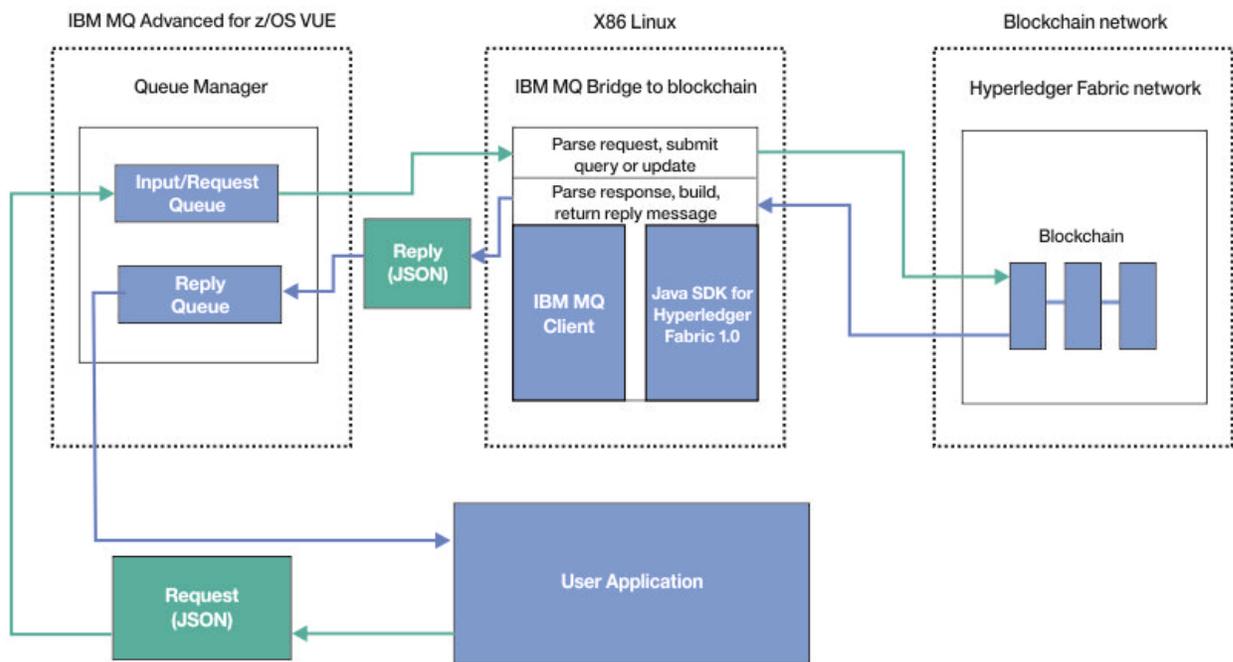


図 125. IBM MQ Bridge to blockchain

参加者またはピアとしてブロックチェーン・ネットワークに接続するように IBM MQ Bridge to blockchain を構成できます。ブリッジの実行中に、メッセージング・アプリケーションからブリッジに、リソースの状態を照会または更新して結果を応答としてメッセージング・アプリケーションに返すチェーンコード・ルーチンを実行するように要求します。

## 手順

1. csq4bcbq.jcl で定義されたブリッジのオブジェクトを作成します。  
ユーザー資格情報およびブリッジへのメッセージ入力に使用されるデフォルトの名前付きキュー (SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE と SYSTEM.BLOCKCHAIN.INPUT.QUEUE) 用にブリッジ・キュー定義のサンプルが提供されています。
  - a) csq4bcbq.jcl を z/OS データ・セットにコピーします。
  - b) csq4bcbq.jcl を編集して、z/OS キュー・マネージャーをカスタマイズします。キュー・マネージャー名と IBM MQ 製品ライブラリーの高位修飾子を指定する必要があります。 **APPL1** ブリッジ・キューの例を変更するか、追加のアプリケーション用に INPUT および REPLY キューを追加するかを選択できます。
  - c) csq4bcbq.jcl を実行依頼して、定義したオブジェクトを作成します。
2. 使い慣れた方法を使用して、x86download.tar.gz を x86download ディレクトリーから x86 Linux 環境に転送します。  
ファイルがバイナリー・モードで転送されるようにしてください。
3. x86 Linux で、x86download.tar.gz を解凍します。

```
tar -xvzf x86download.tar.gz
```

解凍される 4 つのディレクトリーは、bin、lib、prereqs、および samp です。

4. IBM Java runtime environment バージョン 8 を x86 Linux 環境にダウンロードします。
  - a) [IBM Java SDK Developer Centre Java 8 Downloads page](#) で、「**Linux on AMD64/EMT64T**」セクションからファイル名が `ibm-java-x86_64-jre-8.0-4.6.bin` の **Installable package (InstallAnywhere as root)** リンクをクリックします。  
「IBM SDK, Java Technology Edition, Version 8」ライセンス・ページが表示されます。

- b) ライセンスに同意して続行します。  
ダウンロード・ウィンドウで、「ファイルの保存」を選択してダウンロードを開始します。
- c) `ibm-java-x86_64-jre-8.0-4.6.bin` ファイルを実行して、x86 Linux 環境にインストールします。  
デフォルトのインストール場所は、`/opt/ibm/` ディレクトリーです。

```
./ibm-java-x86_64-jre-8.0-4.6.bin
```

- d) IBM 8 JRE のパスを設定します。

```
export PATH=/opt/ibm/java-x86_64-80/jre/bin:$PATH
```

5. IBM MQ 9.0.4 再配布可能 Java クライアントを [Fix Central](#) からダウンロードします。

- a) `9.0.4.0-IBM-MQC-Redist LinuxX64` リンクをクリックします。
- b) 「**ブラウザー (HTTPS) を使用したダウンロード**」を選択します。  
「次へ進む」をクリックします。
- c) ライセンス条件に同意します。
- d) `9.0.4.0-IBM-MQC-Redist-LinuxX64.tar.gz` リンクをクリックし、「ファイルの保存」を選択してダウンロードします。
- e) `9.0.4.0-IBM-MQC-Redist-LinuxX64.tar.gz` を x86 Linux 環境のディレクトリーに解凍します。
- f) Redistributable Java クライアントの解凍先ディレクトリーのパスを設定します。

```
export MQ_JAVA_INSTALL_PATH=/unpack_location/java
```

## タスクの結果

z/OS から x86 Linux 環境に IBM MQ Bridge to blockchain を転送し、IBM JRE 8 と IBM MQ 9.0.4 再配布可能 Java クライアントをインストールしました。

## 次のタスク

z/OS キュー・マネージャーの情報とブロックチェーン・ネットワークからの資格情報を使用して、IBM MQ Bridge to blockchain の構成ファイルを作成します。

## IBM MQ Bridge to blockchain の構成ファイルの作成

キュー・マネージャーおよびブロックチェーン・ネットワークのパラメーターを入力して、IBM MQ Bridge to blockchain が IBM MQ および IBM Blockchain ネットワークに接続するための構成ファイルを作成します。

## 始める前に

- ブロックチェーン・ネットワークを作成し、構成しておきます。
- ブロックチェーン・ネットワークからの資格情報ファイルを入手します。
- x86Linux 環境に、IBM MQ Bridge to blockchain をインストールしました。
- x86 Linux 上に IBM MQ Bridge to blockchain、IBM MQ 9.0.4 再配布可能 Java クライアント、および IBM Java runtime environment バージョン 8 があります。

## このタスクについて

このタスクでは、IBM MQ Bridge to blockchain 構成ファイルを作成し、正常に IBM Blockchain および IBM MQ ネットワークに接続するために必要な最小限のセットアップについて説明します。

ブリッジを使用して、Hyperledger Fabric 1.0 architecture を基盤にしたブロックチェーン・ネットワークに接続できます。ブリッジを使用するには、ブロックチェーン・ネットワークからの構成情報が必要です。このタスクの各ステップで、2つの異なる構成済みブロックチェーン・ネットワークに基づく構成の詳細の例を示します。

- Docker で実行される Hyperledger Fabric ネットワーク。詳細については、[Getting started with Hyperledger Fabric](#)、[Writing your first application](#)、および [649 ページの『Hyperledger Fabric ネットワーク資格情報ファイルの例』](#)を参照してください。
- IBM Cloud (formerly Bluemix)の Kubernetes クラスター内で稼働する Hyperledger Fabric ネットワーク。詳細については、[Develop in a cloud sandbox on IBM Blockchain Platform](#) および [651 ページの『Kubernetes コンテナ・クラスター・ネットワーク構成ファイルの例』](#)を参照してください。

すべての IBM MQ Bridge to blockchain パラメーターの意味とオプションの詳細については、[runmqbcb コマンド](#)を参照してください。独自のセキュリティ要件を考慮し、デプロイメントに適したパラメーターをカスタマイズする必要があります。

## 手順

1. ブリッジを実行して構成ファイルを作成します。

ブロックチェーン・ネットワークの資格情報ファイルおよび z/OS キュー・マネージャーのパラメーターが必要です。タスク [786 ページの『ブロックチェーンで使用するための IBM MQ Advanced for z/OS VUE の構成』](#)で z/OS 環境からブリッジを移動したときに、そのブリッジをアンパックした場所の bin ディレクトリーからブリッジ・スクリプトを実行します。

```
./runmqbcb -o config_file_name.cfg
```

以下の例に示すように、既存の値は大括弧に囲まれて表示されます。既存の値を受け入れる場合は、Enter キーを押します。値をクリアする場合は、Space キーを押してから Enter キーを押します。新しい値を追加する場合は、大括弧で囲んで入力してから Enter キーを押します。値(ピアなど)のリストをコンマで区切ることも、新しい行にそれぞれの値を入力して区切ることもできます。ブランク行はリストの終わりを示します。

**注:** 既存の値は編集できません。それらを保持、置換、またはクリアすることはできます。

2. z/OS キュー・マネージャーへの接続の値を入力します。

接続に必要な最小値は、キュー・マネージャー名、ブリッジ入力の名前、およびユーザーが定義したアイデンティティー・キューの名前です。リモート・キュー・マネージャーに接続する場合は、**MQ Channel**と **MQ Conname** (キュー・マネージャーが実行されているホスト・アドレスとポート) も必要です。ステップ [791 ページの『6』](#)で IBM MQ に接続するために TLS を使用するには、JNDI または CCDT を使用し、それに応じて **MQ CCDT URL** または **JNDI implementation class** および **JNDI provider URL** を指定する必要があります。

```
Connection to Queue Manager
-----
Queue Manager                : [z/OS_qmgr_name]
Bridge Input Queue           : [APPL1.BLOCKCHAIN.INPUT.QUEUE]
Bridge User Identity Queue   : [SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE]
MQ Channel                   : [SYSTEM.DEF.SVRCONN]
MQ Conname                   : [host1.example.com(3714)]
MQ CCDT URL                  : []
JNDI implementation class    : []
JNDI provider URL           : []
MQ Userid                    : []
MQ Password                  : []
```

3. ブロックチェーン・ネットワークの認証局のログインの詳細を入力します。

ローカル Hyperledger Fabric および Kubernetes クラスターの例のデフォルト値は、**Userid** が **admin**、**Enrollment Secret** が **adminpw** です。ご使用のブロックチェーン・ネットワーク用にこれらの値を変更する場合は、適切な値を使用してブリッジを構成していることを確認します。

```
Blockchain - User Identification
-----
```

```
Blockchain Userid          : []admin
Enrollment Secret        : []*****
```

4. ブロックチェーン・ネットワークのメンバーシップおよびアイデンティティ・ルールを管理するメンバーシップ・サービス・プロバイダー ID (MSPID) を入力します。

資格情報ファイルから、**Organisation Name** および **Organisation MSPID** の **msp\_id** パラメータを指定します。649 ページの『Hyperledger Fabric ネットワーク資格情報ファイルの例』から、ファイルの **peer** セクションの **CORE\_PEER\_LOCALMSPID** 値を使用します。651 ページの『Kubernetes コンテナ・クラスター・ネットワーク構成ファイルの例』から、**mSPID** 値を使用します。

```
Blockchain - Organisation Identification
-----
Organisation Name          : []Org1MSP
Organisation MSPID        : []Org1MSP
```

5. ブロックチェーン・ネットワーク・サーバーのロケーション値を入力します。  
649 ページの『Hyperledger Fabric ネットワーク資格情報ファイルの例』から、認証局、ピア、および順序付けプログラムの各エレメントの名前とサーバー: ポートの場所を指定します。

```
Blockchain server locations
-----
Certificate Authority servers : [ca.example.com Docker_container_host:7054] (for
example ca.example.com localhost:7054)
Peer servers                  : [peer0 localhost:7051]
Orderer servers               : [orderer0 localhost:7050]
Peer Event servers            : [peer0 localhost:7053]
Location of PEM file for Blockchain certificate : []
```

651 ページの『Kubernetes コンテナ・クラスター・ネットワーク構成ファイルの例』から、認証局、ピア、および順序付けプログラムの各エレメントの名前とサーバー: ポートの場所を指定します。

```
Blockchain server locations
-----
Certificate Authority servers : [CA1
your_blockchain_network_public_ip_address:30000] (for example CA1 123.456.789.10:30000)
Peer servers                  : [blockchain-org1peer1
your_blockchain_network_public_ip_address:30110]
Orderer servers               : [blockchain-orderer
your_blockchain_network_public_ip_address:31010]
Peer Event servers            : [blockchain-org1peer1
your_blockchain_network_public_ip address:30111]
Location of PEM file for Blockchain certificate : []
```

6. TLS 接続の証明書ストア値を入力します。

ブリッジは、キュー・マネージャーに接続している IBM MQ Java クライアントとして機能します。つまり、TLS セキュリティーを使用して、その他すべての IBM MQ Java クライアントと同様に安全に接続するようにブリッジを構成できます。TLS 接続の構成の詳細は、ステップ [790 ページの『2』](#) で JNDI または CCDT 情報を指定した後にのみ公開されます。

```
Certificate stores for TLS connections
-----
Personal keystore           : []
Keystore password           : []
Trusted store for signer certs : []
Trusted store password      : []
Use TLS for MQ connection    : [N]
Timeout for Blockchain operations : [12]
```

7. IBM MQ Bridge to blockchain のログ・ファイルの場所を入力します。

構成ファイルまたはコマンド行でログ・ファイルの名前と場所を指定する必要があります。

```
Behavior of bridge program
-----
Runtime logfile for copy of stdout/stderr : [/var/mqm/errors/runmqbcb.log]
Done.
```

## タスクの結果

IBM MQ Bridge to blockchain が IBM Blockchain ネットワークおよび IBM MQ z/OS キュー・マネージャーに接続するために使用する構成ファイルを作成しました。

## 次のタスク

793 ページの『IBM MQ Bridge to blockchain の実行』のステップに従います。

### 関連情報

[runmqbc \(IBM MQ Bridge to blockchain の実行\)](#)

## **z/OS V 9.0.3 IBM MQ Bridge to blockchain で使用されるキューのセキュリティ**

IBM MQ Bridge to blockchain で使用されるように定義されている z/OS キューのセキュリティのセットアップに関する考慮事項。

以下の例は、IBM MQ Bridge to blockchain のキューを保護する 1 つの方法を示す RACF プロファイルを示しています。

### RESLEVEL

IBM MQ Bridge to blockchain は、**SVRCONN** チャネルを介して **CHINIT** に接続します。ブリッジ・ユーザーが使用する有効な z/OS ユーザー ID に対して、特定のセキュリティ・チェックが必要であることを想定しています。つまり、ユーザー ID が **CHINIT** タスクで検査されることを確認する必要があります。

**RESLEVEL** プロファイルに対する権限によって、1 つのユーザー ID (チャンネル・ユーザー ID) のみが検査されるか、2 つのユーザー ID (**channel** ユーザー ID と **CHINIT** ユーザー ID の両方) が検査されるかが決定されます。以下に例を示します。

- 以下のコードは、**RESLEVEL** プロファイルの **CHINIT** に対する **READ** 権限を付与します。したがって、**channel** ユーザー ID のみが検査されます。

```
PERMIT RESLEVEL CLASS(MQADMIN) ID(CHINIT) ACCESS(READ)
```

- 以下のコードは **RESLEVEL** プロファイルの **CHINIT** に権限を付与しません。したがって、2 つのユーザー ID が検査され、追加の権限を **CHINIT** ユーザー ID に付与する必要があります。

```
PERMIT RESLEVEL CLASS(MQADMIN) ID(CHINIT) ACCESS(NONE)
```

次のセクションでは、追加の権限を付与するコード行が強調表示されています。

詳細については、[クライアント MQI 要求](#)を参照してください。

## キュー・リソース権限

ID キューをロックダウンし、ブリッジ ID による入出力での使用を許可する

```
RDEFINE MQQUEUE SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE UACC(NONE)
PERMIT SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE CLASS(MQQUEUE) ID(MQBBCART) ACCESS(UPDATE)
PERMIT SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE CLASS(MQQUEUE) ID(CHINIT) ACCESS(UPDATE)
```

ブリッジ ID が入力用にキューを開くことができる

```
DEF QL(CARTAX.BLOCKCHAIN.INPUT.QUEUE) LIKE(SYSTEM.BLOCKCHAIN.INPUT.QUEUE)
RDEFINE MQQUEUE CARTAX.BLOCKCHAIN.INPUT.QUEUE UACC(NONE)
PERMIT CARTAX.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(MQBBCART) ACCESS(UPDATE)
PERMIT APPL1.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(CHINIT) ACCESS(UPDATE)
```

APPCART グループのアプリケーション ID が出力用に要求キューを開くことができる

```
PERMIT CARTAX.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(APPCART) ACCESS(UPDATE)
```

アプリケーション応答キューを対象とするプロファイル

```
RDEFINE MQQUEUE CARTAX.APP.REPLY.** UACC(NONE)
```

APPCART グループのアプリケーション ID が入力用に応答キューを開くことができる

```
RDEFINE MQADMIN CONTEXT.CARTAX.APP.REPLY.** UACC(NONE)  
PERMIT CARTAX.APP.REPLY.** CLASS(MQQUEUE) ID(APPCART) ACCESS(UPDATE)
```

ブリッジ ID が出力用に応答キューを開き、**set\_identity\_context** を使用して書き込むことができる

```
PERMIT CARTAX.APP.REPLY.** CLASS(MQQUEUE) ID(MQBBCART) ACCESS(UPDATE)  
PERMIT CONTEXT.CARTAX.APP.REPLY.** CLASS(MQADMIN) ID(MQBBCART) ACCESS(UPDATE)  
PERMIT CARTAX.APP.REPLY.** CLASS(MQQUEUE) ID(CHINIT) ACCESS(UPDATE)  
PERMIT CONTEXT.CARTAX.APP.REPLY.** CLASS(MQADMIN) ID(CHINIT) ACCESS(UPDATE)
```

## 関連タスク

797 ページの『[IBM MQ Bridge to blockchain クライアント・サンプルの実行](#)』

IBM MQ Bridge to blockchain に用意されている JMS クライアント・サンプルを使用して、Blockchain Bridge が検査している入力キューにメッセージを書き込み、受け取った応答を表示できます。

## 関連情報

[キュー・セキュリティーのためのプロファイル](#)

[API リソース・セキュリティーのアクセス権に関する早見表](#)

## IBM MQ Bridge to blockchain の実行

IBM MQ Bridge to blockchain を実行して、IBM Blockchain と IBM MQ に接続します。接続すると、ブリッジは照会メッセージを処理し、それをブロックチェーン・ネットワークに送信して、応答を受信したり処理したりできるようになります。

## このタスクについて

前のステップで作成した構成ファイルを使用して、IBM MQ Bridge to blockchain を実行します。

## 手順

1. ブリッジで使用する z/OS キュー・マネージャーを始動します。
2. IBM MQ Bridge to blockchain を開始して、ブロックチェーン・ネットワークと z/OS キュー・マネージャーに接続します。

ブリッジ・スクリプトは、786 ページの『[ブロックチェーンで使用するための IBM MQ Advanced for z/OS VUE の構成](#)』タスクで z/OS 環境から移動したときにブリッジを解凍した場所の bin ディレクトリから実行します。

```
./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
```

ブリッジが接続されると、以下のような出力が返されます。

```
Fri Oct 06 06:32:11 PDT 2017 IBM MQ Bridge to Blockchain  
5724-H72 (C) Copyright IBM Corp. 2017, 2023.
```

```
Fri Oct 06 06:32:17 PDT 2017 Ready to process input messages.
```

3. オプション: ブリッジの実行後に、接続が成功しなかったことを示すメッセージが返された場合は、z/OS キュー・マネージャーとブロックチェーン・ネットワークへの接続のトラブルシューティングを行います。

a) デバッグ・オプション 1 を指定して、デバッグ・モードでコマンドを発行します。

```
./runmqbcb -f /config_file_location/config_file_name.cfg -r /log_file_location/  
logFile.log -d 1
```

ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが簡潔モードで表示されます。

- b) デバッグ・オプション 2 を指定して、デバッグ・モードでコマンドを発行します。

```
./runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/  
logFile.log -d 2
```

ブリッジによって接続のセットアップがステップスルーされ、処理メッセージが冗長モードで表示されます。全出力がログ・ファイルに書き込まれます。

## タスクの結果

IBM MQ Bridge to blockchain が開始され、キュー・マネージャーとブロックチェーン・ネットワークに接続されます。

## 次のタスク

- 797 ページの『[IBM MQ Bridge to blockchain クライアント・サンプルの実行](#)』のステップに従って、照会または更新メッセージをフォーマット設定し、ブロックチェーン・ネットワークに送信します。
- `MQBCB_EXTRA_JAVA_OPTIONS` 変数を使用して、例えば、IBM MQ トレースを有効化する JVM プロパティを渡します。詳しくは、[IBM MQ Bridge to blockchain のトレース](#)を参照してください。

## z/OS V 9.0.3 MQ Adv. VUE IBM MQ Bridge to blockchain のメッセージの形式

IBM MQ Bridge to blockchain によって送受信されるメッセージのフォーマット設定に関する情報。

アプリケーションは、IBM MQ Bridge to blockchain が照会を実行するか、またはブロックチェーンで保持されている情報を更新することを要求します。アプリケーションでは、ブリッジ要求キューに要求メッセージを配置することによって、これを行います。照会または更新の結果は、ブリッジによって応答メッセージの形式に設定されます。ブリッジでは、応答メッセージの宛先として、要求メッセージの `MQMD` の `ReplyToQ` および `ReplyToQMgr` フィールドに格納されている情報を使用します。

ブリッジによって消費または生成されるメッセージは、JSON 形式のテキスト (MQSTR) メッセージです。入力メッセージは単純な JSON であり、プログラムでは文字列の連結を使用して、入力メッセージを生成できます。args 以外のすべてのフィールドが必須です。このフィールドの引数リストには、保管されているチェーンコードの関数の知識が必要です。

## 要求メッセージの形式

入力メッセージの形式:

```
{ "function": functionName,  
  "channel" : chainName,  
  "chaincodeName" : codeName,  
  "args" : [ argument list]  
}
```

機能する [Fabcar](#) サンプルを含むローカルの Hyperledger ネットワーク例の場合。

- ブロックチェーンに保持されている車両の詳細を表す JSON オブジェクトのリストを返す `fabcar` チェーンコードの `queryAllCars` 関数を呼び出す照会メッセージを使用するには、メッセージを以下のようにフォーマット設定します。

```
{ "function": "queryAllCars",  
  "channel": "mychannel",  
  "chaincodeName": "fabcar",  
  "args": []  
}
```

応答例:

```
{  
  "statusCode": 200,  
}
```

```

    "statusCode": 200,
    "statusType": "SUCCESS",
    "message": "OK",
    "data": [
      {"Record": {"owner": "Tomoko", "colour": "blue", "model": "Prius", "make": "Toyota"}, "Key": "CAR0"},
      {"Record": {"owner": "Brad", "colour": "red", "model": "Mustang", "make": "Ford"}, "Key": "CAR1"},
      {"Record": {"owner": "Jin Soo", "colour": "green", "model": "Tucson", "make": "Hyundai"}, "Key": "CAR2"},
      {"Record": {"owner": "Max", "colour": "yellow", "model": "Passat", "make": "Volkswagen"}, "Key": "CAR3"},
      {"Record": {"owner": "Adriana", "colour": "black", "model": "S", "make": "Tesla"}, "Key": "CAR4"},
      {"Record": {"owner": "Michel", "colour": "purple", "model": "205", "make": "Peugeot"}, "Key": "CAR5"},
      {"Record": {"owner": "Aarav", "colour": "white", "model": "S22L", "make": "Chery"}, "Key": "CAR6"},
      {"Record": {"owner": "Pari", "colour": "violet", "model": "Punto", "make": "Fiat"}, "Key": "CAR7"},
      {"Record": {"owner": "Valeria", "colour": "indigo", "model": "Nano", "make": "Tata"}, "Key": "CAR8"},
      {"Record": {"owner": "Shotaro", "colour": "brown", "model": "Barina", "make": "Holden"}, "Key": "CAR9"}
    ]
  }
}

```

応答メッセージには、現在ブロックチェーンに保持されているすべての車両レコードが含まれます。

- ブロックチェーン台帳で新しい車両項目を作成する `fabcar` チェーンコード例の `createCar` 関数を呼び出す更新メッセージを使用するには、メッセージを以下のようにフォーマット設定します。

```

{
  "function": "createCar",
  "channel": "mychannel",
  "chaincodeName": "fabcar",
  "args": ["CAR10", "Ford", "Mustang GT", "Blue", "Bob"]
}

```

応答例:

```

{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}

```

ブロックチェーンに新しい車両項目が追加されたことを確認するには、すべての車両を返す初期メッセージを再度使用します。

機能する `example02` デモを含む Kubernetes クラスタ・ネットワーク例の場合。

- ブロックチェーン台帳内のエンティティ `"a"` の値を返す `example02` チェーンコードの `query` 関数を呼び出す照会メッセージを使用するには、メッセージを以下のようにフォーマット設定します。

```

{
  "function": "query",
  "channel": "channel1",
  "chaincodeName": "example02",
  "args": ["a"]
}

```

応答例:

```

{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": "100"
}

```

- 最初の引数に指定されるエンティティを減少させ、2番目の引数に指定されているエンティティを、3番目の引数に指定されている値だけ増分させる `invoke` 関数の `example02` チェーンコードを呼び出すメッセージを使用するには、メッセージを以下のようにフォーマット設定します。

```

{
  "function": "invoke",
  "channel": "channel1",
  "chaincodeName": "example02",
  "args": ["a", "b", "10"]
}

```

値は以下のとおりです。

– 前: a=100、b=200

– 後: a=90、b=210

応答例:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
```

新しい値を検査するには、**"a"** および **"b"** の値を検索する新規メッセージ照会メッセージを実行依頼します。

## 応答メッセージの形式

応答メッセージでは、その関連 ID がインバウンド・メッセージのメッセージ ID に設定されています。ユーザー定義のプロパティが入力メッセージから出力メッセージにコピーされます。応答のユーザー ID は、set-identity コンテキストによって発信元のユーザー ID に設定されます。

正常な処理の例:

```
{ "data": "500", "message": "OK", "statusCode": 200, "statusType": "SUCCESS" }
```

このメッセージの応答データは、チェーンコード応答によって生成されたものです (UTF-8 文字列に変換されるバイト数)。

すべてのエラー応答には、それがブリッジ自体によって生成されたか、またはブロックチェーンの呼び出しやチェーンコード呼び出しから生成されたかにかかわらず、同じフィールドがあります。以下に例を示します。

- 誤ったチャンネル名

```
{
  "message": "Bad newest block expected status 200 got 404, Chain myUnknownChannel",
  "statusCode": 404,
  "statusType": "FAILURE"
}
```

- 誤った JSON 入力メッセージ

```
{
  "message": "Error: Cannot parse message contents.",
  "statusCode": 2110,
  "statusType": "FAILURE"
}
```

- チェーンコードに対する誤ったパラメーター

```
{
  "message": "Sending proposal to fabric-peer-1a failed because of gRPC
failure=Status{code=UNKNOWN, description={\"Error\": \"Nil amount for c\"}, cause=null}",
  "statusCode": 500,
  "statusType": "FAILURE"
}
```

アプリケーションでは、**statusType** 文字列またはデータ・フィールドの存在を確認することによって、要求が成功したか失敗したかがわかります。入力メッセージの処理でエラーが発生し、ブリッジがブロックチェーンに送信しない場合、ブリッジから返される値は MQRC 値で、通常、**MQRC\_FORMAT\_ERROR** です。

## の実行

IBM MQ Bridge to blockchain に用意されている JMS クライアント・サンプルを使用して、Blockchain Bridge が検査している入力キューにメッセージを書き込み、受け取った応答を表示できます。

## 始める前に

IBM MQ Bridge to blockchain が実行中で、IBM MQ Advanced キュー・マネージャーおよびブロックチェーン・ネットワークに接続されており、入力メッセージを処理する準備ができています。

## このタスクについて

IBM MQ Bridge to blockchain の samp ディレクトリーで JMS サンプル・アプリケーションを見つけます。

## 手順

1. クライアント・サンプルの Java ソース・ファイルを編集します。

サンプル内の指示に従って、ご使用の IBM MQ 環境およびブロックチェーン・ネットワークに合わせて構成します。サンプルの以下のコードは、ブリッジに送信する JSON 要求メッセージを定義しています。

```
// Create the JSON request message.
// Modify "query", "exampleBlockchainChannelName", and "exampleChaincodeName" to
// match your deployed blockchain chaincode.
// The "operation" field is optional, but recommended. It should be set to QUERY
// or UPDATE to match what the chaincode is going to do.

JSONObject inputMsg = new JSONObject();
inputMsg.put("operation", "QUERY");

inputMsg.put("function", "query");
inputMsg.put("channel", "exampleBlockchainChannelName");
inputMsg.put("chaincodeName", "exampleChaincodeName");

// Create the JSON arguments for the request message.
// Modify "a" to match your deployed blockchain chaincode
// requirements, and add further arguments as necessary

JSONArray myArgs = new JSONArray();
myArgs.add("a");
inputMsg.put("args", myArgs);

TextMessage message = session.createTextMessage(inputMsg.serialize());
message.setJMSReplyTo(replyToQueue);
```

2. サンプルをコンパイルします。

ブリッジ・ディレクトリーに出荷される IBM MQ クライアント・クラスおよび JSON4j.jar ファイルを指します。

```
javac -cp $MQ_JAVA_INSTALL_PATH/lib/*:../prereqs/JSON4J.jar SimpleBCBCClient.java
```

3. コンパイルされたクラスを実行します。

```
java -cp $MQ_JAVA_INSTALL_PATH/lib/*:../prereqs/JSON4J.jar:. SimpleBCBCClient
```

```
Starting Simple MQ Blockchain Bridge Client
Created the message. Starting the connection
Sent message:
```

```
JMSMessage class: jms_text
JMSType:          null
JMSDeliveryMode: 2
JMSDeliveryDelay: 0
JMSDeliveryTime: 1508427559117
```

```

JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:414d5120424342514d2020202020209063e859ea36aa24
JMSTimestamp: 1508427559117
JMSCorrelationID: null
JMSDestination: queue:///APPL1.BLOCKCHAIN.INPUT.QUEUE
JMSReplyTo: queue:///APPL1.BLOCKCHAIN.REPLY.QUEUE
JMSRedelivered: false
  JMSXAppID: java
  JMSXDeliveryCount: 0
  JMSXUserID: USER1
  JMS_IBM_PutApplType: 6
  JMS_IBM_PutDate: 20171019
  JMS_IBM_PutTime: 15391912
{"args":
["a"],"function":"query","channel":"exampleBlockchainChannelName","operation":"QUERY","chaincodeName":"exampleChaincodeName"}

```

応答メッセージ:

```

JMSMessage class: jms_text
JMSType: null
JMSDeliveryMode: 1
JMSDeliveryDelay: 0
JMSDeliveryTime: 0
JMSExpiration: 0
JMSPriority: 4
JMSMessageID: ID:c3e2d840e2e2f0f840404040404040d2afa27229838af2
JMSTimestamp: 1497439784000
JMSCorrelationID: ID:414d5120424342514d2020202020209063e859ea36aa24 *(JMSMessageID of
the input message)
JMSDestination: null
JMSReplyTo: null
JMSRedelivered: false
  JMSXAppID: java
  JMSXDeliveryCount: 1
  JMSXUserID: USER1
  JMS_IBM_Character_Set: UTF-8
  JMS_IBM_Encoding: 273
  JMS_IBM_Format: MQSTR
  JMS_IBM_MsgType: 8
  JMS_IBM_PutApplType: 2
  JMS_IBM_PutDate: 20171019
  JMS_IBM_PutTime: 15392014
{
  "data": "20",
  "message": "OK",
  "statusCode": 200,
  "statusType": "SUCCESS"
}
Response text:
{
  "data": "20",
  "message": "OK",
  "statusCode": 200,
  "statusType": "SUCCESS"
}
SUCCESS

```

応答の待機中にクライアントがタイムアウト・エラーを受け取った場合は、ブリッジが実行されていることを確認します。

## 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おりません。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 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: